Giờ thứ 01: Làm quen với AS, Your First Look at AS

AS là một ngôn ngữ lập trình được xây dựng trong Flash. AS giống như các ngôn ngữ khác như Javascript, C++ v.v. nhưng bạn không cần biết các ngôn ngữ khác để học AS (nếu có thì càng dễ hơn).

Bắt đâu học AS, thì chúng ta coi thử AS là gì, làm được những gì và có quan hệ gì với các chức năng khác của Flash. Trong bài này, bạn sẽ tìm hiểu coi AS ra đời như thế nào, xem cách viết AS làm sao, tìm hiểu AS làm được những gì và sau sau cùng là xác định được nơi lưu trữ AS ở trong Flash.

Để đi sát nghĩa và tiện cho sau này, mình sẽ không dịch một số từ ngữ ra tiếng việt như script, movieclip v.v.

Script là gì?

AS là ngôn ngữ lập trình, với các lệnh sai khiến Flash movie làm việc theo đúng những gì mình viết (chỉ có mình viết sai chứ computer không làm sai). Phần nhiều thì AS chỉ làm việc trong môi trường của Flash, tuy nhiên AS cũng có thể gửi lệnh cho browser, hệ điều hành v.v.

Script có thể ngắn gọn vài chữ hay cũng có thể dài cả trăm trang. Script có thể được việt gộp lại một chổ hay cũng có thể viết rãi rác khắp nơi trong movie.

Xuất xứ của AS

AS rât giống ngôn ngữ C++, Java, javascript .v.v và được dựa trên tiêu chuẩn do ECMA (European Computer Manufactuers Association) lập ra gọi là ECMAScript. Nhiều người hiểu lầm rằng AS dựa trên Javascript, nhưng thực chất cả 2 đều dựa trên ECMAScript.

Lúc đầu viết script trong Flash rất đơn giản và cho tới Flash 4 mới được phát triển nhưng cũng vẫn còn "thô sơ" với những vòng lặp và các điều kiện "if...else". Cho tới Flash 5 thì dân Flash mới có thể lập trình và gắn liền script với các yếu tố trong movie. . Sang tới Flash MX thì AS đã trở thành ngôn ngữ lập trình toàn diện với hơn 300 câu lệnh, hàm .v.v

Nhận biêt AS

AS đơn giản là những câu lệnh được viết bằng tiếng Anh (vì lẽ này mà mình sẽ không dịch các tư tiếng Anh liên quan đến AS, và một phần thì mình không giỏi thuật ngữ computer hay tiếng việt cho lắm) và các phép tính và dấu câu. Ví dụ sau:

```
ActionScript
on (press) {
   gotoAndPlay ("my frame");
}
```

Bạn có thể giải nghĩa đoạn mã trên bằng cách tìm hiểu các từ chính trong đó. Chữ "press" gợi ý răng người dùng đang kích chuột vào một cái gì đó, (và trong trường hợp này là cái nút) Chữ kế tiếp "gotoAndPlay" do 4 chữ "go to and play" gộp lại, gợi ý rằng AS ra lệnh cho Flash tớimột điểm nào đó trong movie và bắt đầu chơi từ điểm đó.

AS có thể làm những gì?

Flash movie gồm có các scence (cảnh), và mỗi cảnh sẽ có 1 timeline (thời giản biểu???) và timeline sẽ có các frame (khung) bắt đầu từ số 1. Thông thường thì Flash sẽ chơi từ frame 1 cho tới frame cuối của scence với tốc độ cố định và dừng lại hay lặp lại từ đầu tuy theo người làm Flash.

Mục đích chính của AS là thay đổi thứ tự trong cách chơi của Flash. AS có thể dùng ở bất frame nào, hay chạy ngược trở lại frame trước hay nhảy vài frame rồi chơi tiếp. Nhưng đó không chỉ là những gì AS có thể làm được. AS có thể biến film hoạt hình của Flash thành một chương trình ứng dụng có sự tương tác của người dùng. Dưới đây là những cơ bản mà AS có thể làm:

- Hoạt hình: Bạn không cần AS để làm hoạt hình, nhưng với AS thì bạn có thể tạo những hoạt hình phức tạp hơn. Ví dụ, trái banh có thể tưng xung quanh mành hình mà không bao giờ ngừng, và tuân theo các định luật vật lý như lực hút, lực ma sát, lực phản v.v Nếu không có AS thì bản cần phải dùng cả hàng ngàn frame để làm, còn với AS thì chỉ 1 frame cũng đủ
- Navigation (hông biết dịchlàm sao cho hay): thay vì movie chỉ chơi từng frame 1 theo thứ tự thì bạn có thể dừng movie ỏ bất cứ frame nào, và cho phép người dùng có thể chơi ở bất cứ frame nào .v.v
- Thu nhập thông tin từ người dùng (user input): bạn có thể dùng AS để hỏi người dùng 1 cầu hỏi, rồi dùng thông tìn đó trong movie hay có thể gửi cho server hay làm những gì bạn muốn.
- Thu nhập thông tìn từ các nguồn khác: AS có thể tương tác với server và lấy các thông tin từ server hay text file
- Tính toán: AS có thể làm bất cứ phép tính nào mà toán học cho phép
- Thay đổi hình ảnh trong movie: AS có thể thay đổi kích thước, màu sắc, vị trí của bất cứ movie clip (MC) nào trong movie flash của bạn. Bạn có thể tạo thêm phiên bản hay xoá bới phiên bản của MC với AS
- Phân tích môi trường của máy tính: Với AS bạn có thể lấy giờ từ hệ điều hành hay địa chỉ đang chơi movie Flash đó.
- Điều khiển âm thanh trong flash movie: AS là cách tốt nhất để điều khiển âm thanh trong Flash,AS có chơi chậm, chơi nhanh, ngừng, quay vòng .v.v bất kỳ âm thanh nào trong Flash.

Phát triển các khả năng của AS

Điều quan trọng nhất mà AS có thể làm cho bạn là những gì chưa ai nghĩ tới Với AS và trí tưởng tượng và óc sáng tạo của bạn thì không có gì là không thể xảy ra với AS được.

Mục đích của các sách, và của VNFX là hướng dẫn bạn hiểu biết về Flash để từ đó

bạn có thể tự dùng nó để làm ra những sản phẩm tuyệt vời mà không ai có.

Viết AS ở đâu

Câu hỏi đầu tiên những ai tìm hiểu AS thường hỏi là "Viết AS vào đâu?". Flash có một môi trường làm multimedia khá phức tạp. Nếu bạn đã dùng Flash rồi hay đã đọc qua các hướng dẫn đi kèm với Flash, thì bạn đã biết các yếu tớ cơ bản như timeline, key frame v.v. nêu bạn chưa biết thì mình khuyên bạn nên tìm hiểu trước rồi tiếp tục ở đây. Ơ trong mục Flash Tổng Quá, VNFX có post link để download 3 cái CD hướng dẫn cơ bản cho người chưa bao giờ dùng Flash.

Viết AS ở time line

AS có thể viết vào key frame ở bất cứ timeline nào. Để làm như vậy, bạn chọn key frame ở trong timeline trước rồi bạn mở Action Panel ra (có thể nhấn F9) để viết hay xem AS đã được viết ở đó. Mình khuyên bạn nên dùng Action Panel dưới dạng Expert hơn là Normal, vì như vậy bạn sẽ học được nhiều và lẹ hơn.

Khi viết AS vào key frame như vậy thì AS ở key frame đó sẽ hoạt động khi mà Flash chơi tới key frame đó. Ví dụ như lệnh AS **stop()** được đặt ở key frame 5 thì Flash chơi tới key frame 5 thì sẽ dừng lại cho tới khi có lệnh khác. cách viết này gọi là viết vào time line

Một trong những lý do viết script vào time line là khi bạn muốn dùng function (hàm), vì function cho phép chúng ta tái sử dụng đoạn mã đó từ nhiều nơi, nhiều level khác.

Viêt AS ở nút

Các phần tử của Flash movie được gọi là symbol (biểu tượng). thường thì symbol là các hình ảnh. Và có 3 loại symbol: button (nút), graphich (hình ảnh), movie clips(đoạn phim). 3 loại trên thì AS không thể viết liền với graphic, còn button và movie clip thì OK.

Nút sẽ không có tác dụng gì cả nếu như không kèm AS lên trên nó. Muốn kèm AS vào nút thì chọn nút trên stage (sân khấu: diện tích của flash movie), rồi sau đó mở Action panel và viết vào đó như ta viết vào key frame.

AS viết trên nút chỉ hoạt động khi chuột hoặc bàn phím tác động lên nút mà thôi

Viết AS ở MC

MC khác với graphic ở chổ MC được đặt tên khi mang vào stage, MC thường là hình ành động, và có thể có AS kèm theo MC. Để viết AS vào MC thì cũng tương tự như ta làm với nút.

AS kèm theo MC có thể điều khiển MC đó hay là các MC trong cùng một timeline hay các timeline ở ngoài movie

Bài tập

1. Tạo một movie mới
2. tạo 3 key frames
3. Dung Flash vẽ mỗi hình khác nhau cho mỗi frame
4. Chạy thử movie (nhấn Control + enter)
5. Sau đó trở lại timeline, viết vào keyframe 2 đoạn mã : "stop();"
6. Cho chạy thử movie lần nữa
7. Và kỳ này bạn chỉ thấy Flash chơi tới frame thứ 2 thì ngừng, và bạn đã thành công trong việc viết AS :

Giờ thứ 02: Sử dụng Action panel, Using the Script Editing Window

Nếu dân đồ hoạ coi Timeline là nhà thì dân lập trình với AS coi Action Panel là nơi cư ngụ của mình, tại đó dân AS có thể thay đổi, viết tất cả các lệnh. Vì vậy rất quan trọng là bạn có thể sự dụng quen thuộc Action Panel

Flash MX sử dụng các panel để giúp bạn có thể lấy các thông tin về movie bạn đang làm một cách dễ dàng. Khi bạn cài đặt và chay Flash lần đầu tiên, Flash sẽ hỏi bạn muốn sử dụng Flash như thế nào và Flash sẽ xếp đặt các panel cho hợp lý. Bạn cũng có thể thay đổi cách xếp đặt theo ý của bạn.

Nói chung bạn nên xếp đặt các panel của bạn làm sao cho thuận tiện cho bạn nhất, và cái này cũng cần có kinh nghiệm đó ***Mình dùng dual monitor, mình close hết các panel chỉ giữ lại 2 cái chính trên màn hình, mình kéo cái action panel và property panel qua một bên monitor, còn cái bên kia để cho cái stage, nếu cần mở panel nào thì mình dùng phím tắt để mở.***

Sử dụng Action panel có 2 cách: bình thường (normal) và chuyên dụng (expert). Khi mới chạy máy lần đầu thì Action panel được set dưới dạng bình thường. Ở dạng bình thường thì bạn không có thể đánh trực tiếp các câu lệnh, mà phải chọn từ các menu bên tay trái của panel. Với setup như vậy thì bạn không bao giờ phạm phải lỗi khi viết AS cả.

Mình sẽ bỏ phần hướng dẫn dùng normal vì mình thấy dùng normal mode sẽ không bao giờ tiến được, dùng expert mode lúc đầu hoi mệt nhưng bảo đảm là các bạn sẽ học được mau và nhớ lâu hơn

Sử dụng Action panel với expert mode thì bạn có nhiều tự do hơn, nhưng tự do đồng nghĩa với trách nhiệm nên bạn phải cẩn thật khi viết code. Từ normal mode chuyển sang expert mode thì nhấn phím tắt CONTROL + SHIFT + E, chuyển ngược lại thì dùng CONTROL + SHIFT + N (nhớ kích chuột vào action panel trước khi dùng phím tắt). Bạn có thể đổi từ Normal mode sang expert mode bất cứ lúc nào, nhưng đổi ngược lại thì chỉ được khi mà code của bạn không có lỗi.

Action panel có popup menu (mũi tên chỉ xuống dưới, nằm ở góc phải của action panel) và trong đó có nhiều lệnh rất tốt cho bạn khi bạn dùng Action panel, nhất là đối với Expert mode. Bạn có thể tìm và thay thế bất cứ câu nào, chữ nào trong script của bạn. Phím tắt cho tìm kiếm là Control + F và cho thay thế là Control + H, nhấn F3 thì Flash sẽ tiếp tục tìm chữ, câu bạn muốn tìm.

Một lệnh nữa là "Goto Line" giúp bạn tìm được câu bạn muốn tìm. Lệnh "Check Syntax" sẽ rà soát script của bạn và tìm những lỗi cú pháp. Lúc mới tập viết AS, thì dùng nên thường xuyên dùng Chech Syntax để check lỗi trong sctipt. Lệnh tiếp theo là "Auto Format" và "Auto Format Options" giúp cho bạn trình bày script của bạn cho dễ đọc hơn. Bạn có thể thay đổi màu sắc, font chữ v.v. cho script của bạn để cho bạn dễ đọc code hơn bằng cách vào Edit > Preferences > ActionScript Editor. Các phần còn lại của pop-up window cho phép bạn được xuất hay nhập AS dưới dạng text file và in script ra giấy.

Tuy gọi là expert mode nhưng thực ra dùng còn dễ hơn là normal mode. Bởi vì dưới expert mode thi Action panel chẳng khác gì chương trình "note pad" hay "text pad". Bạn chỉ việc viết những gì bạn muốn vào đó. Khi dùng dưới expert mode, bạn vẫn có thể tham khảo các câu lệnh, các cú pháp của AS ở cái khung bên trái của Action Panel. Và nếu bạn tôn thủ một số quy tắt thì bạn sẽ sử được chức năng "Code Hint" của Action Panel, chức năng này tạo nên nhưng khung nhỏ nhỏ trong đó có những câu lệnh của AS đúng hợp quy cho bạn lựa chọn.

Để trở thành một "cao thủ" về AS thì bạn không thể nào không nắm vững về Action panel được.

Sau đây là bài thực hành nho nhỏ, giúp bạn làm quen với Action panel dưới expert mode

1. Tạo một movie mới
2. Chọn key frame đầu tiên (frame số 1) của layer 1 (mình không dịch chữ layer = lớp vì dễ trùng với class= lớp)
3. Mở Action panel (có thể dùng phím tắt hay vào trong phần Window > Action panel
4. Chuyển sang expert mode (dùng phím tắt hày vào trong pop-up menu của Action panel để đổi)
5. Vào chổ để viết AS
6. Viết đoạn code sau:

ActionScript
```
trace ("I am expert");
```

Khi bạn thử movie thì dòng chữ "I am expert" sẽ được hiện ra trong Output panel, và mình sẽ đề cập đến lệnh trace() vào những giờ tới 

Giờ thứ 03: Học cách lập trình, Learning to program

Lập trình có thể coi là khoa học và nghệ thuật. Vì vậy lập trình viên giỏi cần có 2 điều kiện kiến thức về ngôn ngữ mà họ dùng và tính sáng tạo, ít khi nào có trường hợp chỉ có 1 cách lập trình cho một đề án lắm.

Trong quá trình viết AS, bạn sẽ tạo ra cho riêng mình một phong thái viết code khá đặc trưng. Có thể mới đầu bạn sẽ dùng phong cách viết code trong những bài viết này, rồi sau đó bạn sẽ từ từ chuyển sang phong cách của riêng bạn

Thường thì khái niệm về các ngôn ngữ lập trình đều giống nhau, và trong bài này sẽ bàn về những khái niệm lập trình và làm sao áp dụng nó trong Flash

Cách suy nghĩ của computer

Thật ra computer dốt lắm chẳng biết suy nghĩ đâu, chỉ biết làm việc theo mệnh lệnh một cách không suy nghĩ. Vì vậy computer không bao giờ lỗi mà lỗi ở người viết lệnh sai khiến nó.

Còn AS chỉ là một chuỗi lệnh, chỉ thị cho computer, hay đúng ra là Flash phải làm những gì trong môi trường runtime (runtime environment).

Khi viết AS thì bạn nên tự coi mình "ngu" như computer, và đọc lại những dòng lệnh AS và nghĩ tới kết quả của việc thi hành lệnh đó. Khi tự đặt mình vào vị trí của computer, bạn sẽ đoán được script của bạn sẽ làm những gì trong môi trường runtime, và bạn có thể thấy ra những lỗi mà bạn vô ý mắc phải. Khi mà bạn trở thành diễn viên tuyệt vời trong vai computer thì bạn cũng trở thành một lập trình viên tài ba.

Lệnh, hàm, và phép tính

Lệnh (command) là yếu tố căn bản của AS để chỉ thị Flash làm một hành động cụ thể. Lệnh được thi hành tuyệt đối trong các trường hợp khả thi. Ví dụ nếu như ta viết **gotoAndPlay(5)** để cho Flash tới frame 5 và chơi, nhưng nếu trong movie không có frame 5 thì lệnh không thể làm được.

Hàm (function, mình sẽ dùng từ function nhiều hơn là hàm) là phần tính toán các phép tính và cho ra (return) một kết quả. Ví dụ hàm random() sẽ cho ra một con số ngẫu nhiên nào đó, hay hàm sqrt() sẽ cho ra giá trị bình phương của 1 số mà hàm nhận được truyền cho.

Lệnh và hàm đều có thể được truyền cho 1 hay nhiều thông số (parameter, mình sẽ dùng từ parameter thay vì thông số), và những thông số đó giúp cho hàm và lệnh được cụ thể hơn. Ví dụ **gotoAndPlay(5)** thì 5 là thông số, nếu như không có thông số này thì lệnh "gotoAndPlay" sẽ không biết rõ phải nói Flash đi đâu. cũng như hàm sqrt() cũng cần có thông số để cho ra giá trị bình phương của số ấy.

Sự khác biệt giữa hàm và lệnh chính là các phép tính. Phép tính thường là những dấu +, -, *, / hay là phép so sánh <, ==. >. Hàm có phép tính, lệnh thì không

Biến số

Biến số (parameter) cần thiết trong mọi ngôn ngữ lập trình vì nó lưu trử thông tin về một cái gì đó trong chương trình. Biến số gồm có 2 phần: tên và giá trị. Đôi khi bạn chỉ cần lưu trử thông tin trong biến số một thời gian ngắn, ví dụ bạn muốn Flash thi hành 1 số lệnh 10 lần thì bạn cần đếm số lần mà Flash đã thi hành lệnh để có thể ra lệnh cho Flash ngưng ngay sau khi thi hành xong lệnh lần thứ 10. Nhưng cũng có lúc bạn cần phải lưu lại thông tin trong một thời gian dài hay trong suốt quá trình thời gian Flash chơi.

Tên của biến số thường là một nhóm từ, hay cũng có thể chỉ đơn giản là 1 từ, hay 1 chữ cái. Thông thường tên của biến số nên rõ ràng dễ hiểu và nói lên được tính chất của thông tin mà biến số đang lưu trử. Ví dụ nếu bạn muốn lưu trử tên của người dùng trong biến số thì bạn nên đặt tên biến số đó là userName (mình nghĩ đặt tên tiếng Anh dễ hiểu hơn, vì tiếng việt không dấu cũng dễ bị hiểu lầm), bạn có thể đặt tên biến số đó là "n" nhưng quá ngắn, người đọc script của bạn sẽ khó hiểu, nếu đặt là name thì dễ trùng với các biến số hay từ khoá khác (_name là một đặc tính của đối tượng MovieClip)

Khi viết AS, bạn cần lưu ý viết làm sao cho dể hiểu và dễ đọc. Và các lập trình viên quy ước tên biến số theo quy tắc sau: tên biến số luôn viết thường chữ đầu tiên, và viết hoa chữ cái đầu tiên cho các chữ kế tiếp, ví dụ userName, userLastName, userMotherMaidenName v.v.

Lưu ý rằng chỉ các chữ cái và số mới được dùng để đặt tên cho biến số, và luôn bắt đâu tên bằng chữ cái.

Có nhiều loại biến số để lưu trử nhiều loại thông tin, và ngay mỗi loại thông tin cũng có thể có nhiều loại khác nhau nữa. Như số (number) là một loại giá trị của biến số, và dưới nó còn có các loại khác như số nguyên (integer), số ... không nguyên  (floating, double). Bạn cũng có thể dùng biến số để chứa các chuỗi (chuỗi gọi tắt cho chuỗi chữ cái), chuỗi có thể chỉ có 1 chữ cái, nhiều chữ cái hay là không có gì hết, tất cả các chuỗi sau đây đều hợp lệ: "ablsdfjksl", "a", " ", "". Khi viết một chuỗi thì cần dùng dấu ngoặc kép ("") để bắt đầu và kết thúc chuỗi.

Trong các ngôn ngữ lập trình khác như Java, C++, hay ngay cả ActionScript 2.0 trong Flash MX 2004 thì bạn phải xac định trước loại thông tin nào bạn sẽ lưu trong biến số. Nhưng với ActionScript 1 (trong FlashMX) thì bạn không cần làm việc này. Ví dụ biến số userName lúc đầu chứa 1 chuỗi, nhưng sau đó lại chứa 1 số nguyên thì cũng vẫn hợp lệ.

Ngoài chuỗi và số, còn có nhiều loại khác nữa nhưng chúng ta sẽ đề cập tới vào những bài sau.

Điều kiện

Trong những trường hợp chúng ta không biết nên ra lệnh cho Flash phải làm gì cho thích hợp với từng tình huống thì ta có thể dùng "điều kiện" để ra lệnh cho Flash.

Ví dụ nếu như bạn ra lệnh cho Flash không cho người dùng coi một đoạn phim trong Flash nếu như user dưới 18 tuổi, nếu user trên 18 tuổi thì hãy chơi đoạn phim ấy. Trước tiên, Flash sẽ so sánh số tuổi của user với số 18, nếu như số tuổi của user thoả mãn điều kiện chúng ta đặt ra thì Flash sẽ có 1 giá trị **true** từ phép so sánh trên, và ngược lại sẽ là **false** Điều kiện sẽ luôn luôn là đúng (true) hay sai (false). Một giá trị chỉ có đúng hay sai thì được gọi là **boolean**. Sau khi thực hiện phép so sánh và có được kết quả từ phép so sánh trên, Flash sẽ chọn một trong 2 giải pháp do chúng ta đưa ra cho từng trường hợp.

Đôi khi chúng ta cần có nhiều điều kiện hơn chỉ là đơn giản "true" hay "false", ví dụ như bạn muốn Flash chơi đoạn phim A cho ngươi trên 18 tuổi, dưới 18 nhưng trên 13 thì chơi đoạn phim B, và những ai dưới 13 thì chơi đoạn phim C.

Vòng lặp

Con người làm ra computer vì lười  không muốn làm nhiều, mà bán cái cho computer. Nhất là phải làm đi làm lại 1 việc nào đó thì càng nhàm chán. Vì vậy vòng lặp (loop) là một yếu tố quan trọng trong các ngôn ngữ lập trình. AS cũng vậy, bạn có thể dùng vòng lặp trong script.

Trong vòng lặp, "điều kiện" rất quan trọng. Mọi vòng lặp cần có điểm bắt đầu và điểm dừng và một điều kiện để báo hiệu điểm dừng của vòng lặp. Ví dụ như bạn muốn cho vòng lặp chạy 10 lần thì sẽ có 1 biến số dùng để đếm vòng lặp, bắt đầu từ 0, Mỗi vòng lặp chạy thì biến số này sẽ tăng thêm 1. Khi tới 9 thì vòng lặp sẽ dừng lại. Sau đây minh hoạ của vòng lặp này:

1. Một số lệnh trứoc vòng lặp
2. Bắt đầu vòng lặp, set biến số counter = 0
3. Làm một số lệnh trong vòng lặp
4. Tăng biến số counter +=1
5. Nếu biến số counter nhỏ hơn 9, trở lại bước thứ 3
6. Ra khỏi vòng lặp, và tiếp tục chương trình.

Ở đây chúng ta bắt dầu biến số counter = 0 vì quy ước thông thường các ngôn ngữ lập trình đều bắt đầu vòng lặp ở 0.

Một điểm đáng lưu ý của vòng lặp là điều kiện đước xét trước khi thực hiện các lệnh trong vòng lặp. Trong vài trường hợp, điều kiện sẽ được xét sau khi thực hiện các lệnh trong vòng lặp. Chúng ta cũng có thể ngưng vòng lặp trước điểm dừng của nó, và điều này sẽ được đề cập tới vào bài kế tiếp

Làm những điều không tưởng

Lệnh, hàm, phép tính, biến số, thông số, điều kiện, vòng lặp là những phần căn bản trong ngôn ngữ lập trình, và cái này thì ai cũng biết nhưng làm sao phối hợp lại thành một chương trình hoàn hảo mới là cái khó.

Chương trình đơn giản chỉ là một tập hợp lệnh cho computer để giải quyết 1 vấn đề nào đó. Vì vậy trước khi viết 1 chương trình chúng ta cần phải xác định "vấn đề" cần phải giải quyết. Ví dụ thực tế ở ngoài đời, mẹ bạn nhờ bạn ra chợ mua gà. Nhưng chỉ

đơn giản nói ra chợ mua gà thì chưa đủ vì biết mua gà sống hay gà làm rồi. nếu mua gà sống thì mua loại nào v.v. Đó là chưa nói tới mua gà ở chở nào, giá cả ra sao v.v. Nếu như bạn mẹ nói rõ là ra chợ bến thành, mua 1 con gà mái dầu, nặng khoảng 2 kg, với giá khoảng 10000 thì thật là dễ dàng cho bạn phải không?

Biết được vấn đề mua gà rồi thì tìm cách mua gà. Bạn phải "lên kế hoạch", nên nhờ bạn chở hay đi taxi ra chợ bến thành. Nếu đi taxi đi hãng nào, tìm số phone để gọi, v.v. còn nhờ bạn chở đi thì nhờ tên nào. Rồi nên mặc quần áo nào đi chợ. Ra đến chợ thì phải mặc cả làm sao, trả tiền mặt hay ghi sổ nợ. Bạn thấy không, từ một chuyện mua gà đơn giản vậy mà có thể tốn cả ngày trời để lên chương trình

Viết AS, bạn cần phải lưu ý tới tất cả mọi việc dù nhỏ cách mấy để cho Flash có thể làm đúng theo như ý bạn trong mọi tình huống. Các chương trình có bug không phải vì người viết dỡ mà vì chưa nghĩ tới hết mọi tình huống thôi.

Tóm lại, điều quan trọng trong lập trình là khả nắng phân tích một vấn đề chính thành nhiều vấn đề nhỏ cho tới khi không còn nhỏ hơn nữa, và sau đó xây dựng chương trình từ giải quyếy các vấn đế nhỏ lên dần cho tới vấn đề chính.

Viết mã hoàn chỉnh

Bọ (bug) đơn giản là lỗi của chương trình mà bạn tạo ra. Bug có thể chỉ là những lỗi cú pháp đơn giản, hay là những lỗi phức tạp do cách bạn giải quyết vấn đề trong môi trường runtime.

Để tránh có bug trong script của bạn thì bạn nên tốn nhiều thì giờ vào giai đoạn phân tích, thiết kế chương trình cho script của bạn. Kiểm tra các đoạn code nhiều lần, và đóng vai "computer" cho thật giống, đừng suy nghĩ, mà thi hành các lệnh trong script của bạn. Viết từng đoạn code nhỏ rồi ráp lại với nhau (vì vậy mà lập trình theo hướng đối tượng được sử dụng nhiều nhất)

Nếu script của bạn có bug thì đừng thế mà nản lòng, vì không có chương trình nào do con người làm ra mà không có bug cả. Vì vậy bạn đừng cố gắng viết code sao cho tuyệt hảo không có lỗi. Nếu có bug thì bạn diệt bug thôi Bạn có thể chuẩn bị "chiến đấu" với bug bằng cách viết code sao cho dễ hiểu và dễ đọc, viết nhiều chú thích cho các đoạn mã. Đôi khi diệt bug (debug) rất đơn giản nhưng có khi cũng rất kho mà biết bug ở đâu để diệt. Flash có kèm theo vài công cụ giúp bạn debug dễ dàng hơn, chúng ta sẽ nói tới các công cụ này trong bài tới.

Giờ thứ 04: Viêt code trong Flash, Writing code in Flash

Khi bạn viết script, bạn sẽ dùng tất cả những từ khóa, và ký hiệu v.v. Vậy trước tiên chúng ta sẽ phân tích một đoạn script sau đây. Đoạn script này gắn vào 1 button.

ActionScript
```
on (press) {
  var myVariable = 7;
  var myOtherVariable = "Macromedia";
```

```
  for (var i=0; i<10; i++) {
     trace(i);
     if (myVariable + 3 == 5) {
        trace(myOtherVariable);
     }
  }
}
```

Dòng đầu tiên xác định những đoạn code sau đó được kích hoạt khi mà user nhấn vào nút. hàm **on(press)** chỉ có thể sự dụng trong vào trong nút mà thôi. Ngoài ra bạn có thể dùng **on(release)** nếu như bạn muốn kích hoạt đoạn mã trên sau khi user nhấn nút, và buông tay.

Dấu ngoặc móc **{}** ở đầu và cuối đoạn code, gói đoạn code đó vào làm một, và đều được kích hoạt khi nhấn nút. Các bạn lưu ý thấy đoạn code được viết vào trong để cho dễ nhận thấy cả đoạn code này phụ thuộc vào **on(press)**. Các bạn nên viết lùi vào trong cho những đoạn code phụ thuộc vào 1 lệnh hay hàm nào đó.

Dòng thứ hai lập ra biến số (mình sẽ viết tắt là **var** nhe) tên **myVariable**, và đặt giá trị bằng 7 cho nó. tương tự hàng kế tiếp lập ra var tên là **myOtherVariable** và cho giá trị của nó là "Macrmedia". Cả 2 câu này được kết thúc bằng dấu chấm phẩy **;**

Dòng thứ 4 chính là vòng lặp (mình sẽ gọi là loop cho tiện nhe) **for** (sẽ nhắc tới các loại loop ở phía dưới) và các điều kiện của loop này. Nó sẽ lặp 10 lần với điều kiện bắt đầu là **i=0** và nó sẽ tăng thêm 1 cho mỗi lần lặp cho tới khi nó lên tới 9. Tương tự như **on(press)**, for loop cũng có đoạn code cho riêng nó, và đực bọc quanh bằng **{ }** và đoạn code trong đó chỉ kích hoạt khi mà các điều kiện của for loop được thỏa mãn.

Dòng thứ 5 là lệnh **trace()**, lệnh này chỉ viết thông tin ra ở output window trong lúc bạn làm việc với Flash, user sẽ không thấy được. Kế tiếp là điều kiện, **if** là từ khoá trong Flash, và nó sẽ kiểm tra kết quả của phép so sánh **myVariable + 3 ==5**. Nếu mà kết quả là đúng thì nó sẽ kích hoạt lệnh **trace** ở trong, và sẽ viết ra output window giá trị của **myOtherVariable**

Vậy là bạn đã thấy một đoạn script hoàn chỉnh bằng AS rồi. Tiếp theo chúng ta sẽ nói đến vài phần khác của AS.

Output window

Ở trên chúng ta có nhắc tới output window, mà không nói rõ là gì. Output window là một window trong software Flash, và chỉ xuất hiện khi mà bạn chảy thử movie. Nếu Flash compile movie cho bạn và phát hiện ra lỗi thì Flash sẽ viết ra các output window này. Trong khi chạy thử movie, thì lệnh **trace()** mới có thể viết các thông tin ra output window. Output window rất quan trọng trong việt "diệt bọ", dùng nó chung với **trace** bạn có thể quan sát các thông số, biến số, các đối tượng, thuộc tính đối tượng .v.v

Trong lúc học AS, bạn có thể viết một đoạn script ngắn không làm gì cả những chỉ để viết thông tin ra output window, như đoạn code trên.

Thực hành: viết thông tin ra output windown

Cách tốt nhất để hiểu rõ chức năng của output window là sử dụng nó. Bây giờ mình sẽ viết một đoạn code ngắn gửi thôi tin ra output window nhe.

1. Tạo một file mới

2. Chọn frame đầu tiên của movie, mở Action panel. Chỉnh kích thước cho action panel đủ lớn để viết, và nhớ dùng expert mode

3. Kích chuột vào phần viết script và viết câu sau: **trace("hello world");**

4. Chạy thử movie (Control + Enter)

5. Bạn thấy gì ở output window? (Nếu output window của bạn chưa mở thì hãy nhấn F2)

Giống như action panel, output window có 1 cái pop-up menu nho nhỏ ở trên góc phải. Bạn có thể dùng nó để copy, xoá hay save nội dung của output window hay có thể tìm kiếm chữ, v.v. Ngoài ra, pop-up menu này có phần cho bạn chỉnh chế độ "diệt bọ", bạn có thể chọn không cần output window in ra các lỗi (none), hay chỉ in lỗi (error), hay chỉ in cảnh báo (error) và cuối cùng là in ra chi tiết các lỗi hay cảnh báo (verbose)

Biến số cục bộ và toàn bộ

Ở trong giờ thứ 3, chúng ta có nhắc đến variable (biến số) dùng để lưu trữ thông tin. Và sử dụng variable trong AS rất dễ dàng. Bạn chỉ cần ấn định giá trị cho variable. Ví dụ: **myVariable = 7**. Chúng ta tạo variable có tên là myVariable và ấn định 7 là giá trị cho nó. (bạn có thể đặt tên cho variable là bất cứ gì bạn muôn)

Bây giờ bạn có thể thử viết đoạn code sau:

```
ActionScript
var1= 7;
var2= "hello world";
trace ("var1: " + var1+ " /var2: " + var2);
```

Khi bạn chạy thử movie thì output window sẽ có hàng chữ sau: **var1: 7 /var2: hello world** Vì số 7 và "hello world" được chứa trong var1 và var2 sẽ được in ra.

Variable có 2 loại, local và global. **Global variable** (biến số toàn bộ) thì bạn có thể truy cập giá trị của nó ở bất cứ nơi nào trong movie. Tạo global variable không đòi hỏi một bạn phải làm một cái gì đặc biệt cả, bạn có thể dùng nó như cách trên, và Flash tự động biến nó thành global variable. Flash movie dùng hệ thống **level**, và timeline của movie chính là **root leve** (gốc), còn các movie clip cũng chính là một Flash movie nhỏ ở trong Flash movie lớn. Các hình ảnh, script ở trong một movie clip là 1 level thấp hơn root level.

Local variable (biến số cục bộ), khác với global variable, local variable chỉ có trể truy cập trong cùng một đoạn code, hay trong cùng một timeline. Khi dùng local variable thì khi ra khỏi timeline hay đoạn code đó thì Flash sẽ xoá local variable ra khỏi bộ nhớ. Muốn tạo local variable thì dùng từ khoá **var** trước tên của local variable, ví dụ: **var myLocal = "This is local";** Bạn chỉ cần dùng từ khoá **var** 1 lần thôi, những lần dùng sau đó thì chỉ cần dùng tên của local variable thôi. Ví dụ đoạn code:

ActionScript
```
var myLocal = 9;
myLocal = 11;
trace(myLocal);
```

Phép so sánh và các phép tính

So sánh 2 giá trị trong AS rất đơn giản, dùng các ký hiệu toán học như **<, >, =**

Khi ấn định giá trị thì dùng dấu =, và để cho khác biệt thì khi so sánh dùng ==. Đoạn code sau xét coi giá trị của a có bằng 7 không, và in kết quả ra output window. Và khi test đoạn code sau, bạn sẽ thấy **true** ở output window

ActionScript
```
var a = 7;
trace(a == 7);
```

Nếu bạn dùng lộn = với == thì sẽ bị lỗi ở runtime chứ Flash sẽ không có thể tìm được lỗi này cho bạn.

Bạn có thể dùng == để so sánh 2 chuỗi mẫu tự:

ActionScript
```
var myString = "Hello World.";
trace(myString == "Hello World.");
trace(myString == "hello world.");
```

Khi bạn test đoạn code trên thì bạn sẽ có được "true" và "false" ở output window, vì lần so sánh thứ nhất thì bằng nhau, nhưng lần thứ hai thì không vì chữ H và h khác nhau.

Nếu bạn muốn thử coi 2 giá trị có khác nhau không thì dùng ky hiệu **!=**

ActionScript
```
var a = 7;
trace(a != 9);
trace(a != 7);
```

Hàm trace đầu tiên sẽ cho ra "true" vì 7 khác 9, và cái thứ hai thì sẽ cho ra "false"

Nếu bạn muốn thử giá trị coi lớn hơn hay nhỏ hơn thì dùng > và <

ActionScript
```
var a = 7;
trace(a < 8);
trace(a > 6);
trace(a < 1);
```

Đoạn code trên sẽ cho ra "true", "true" và "false" trong output window vì "a" quả thật nhỏ hơn 8 và lớn 6, nhưng không nhỏ hơn 1.

Nếu bạn muốn thử giá trị coi lớn hơn hay bằng nhau hoặc là nhỏ hơn hay bằng nhau thì dùng ký hiệu >= và <=

ActionScript
```
var a = 7;
trace(a <= 9);
trace(a >= 5);
trace(a >= 7);
```

Đoạn code trên sẽ cho ra 3 kết quả "true" cả.

Bạn có thể thay đổi giá trị của variable với các phép tính đơn giản như cộng (+), trừ (-), nhân (*), chia (/). Ví dụ muốn thêm 4 vào giá trị của a thì viết **a = a + 4**. AS cũng có cách viết tắt như Java, C++ cho vì dụ này, **a += 4** Nếu bạn muốn thêm 1 vào a thì bạn có thể viết như 2 cách trên **a = a +1** và **a += 1** và còn cách thứ 3 **a++** . Dấu ++ chỉ làm tăng thêm 1 cho giá trị đó thôi. Có 2 cách dùng ++, 1 là để sau variable như cách trên, và 1 cách thì để trước variable. Bây giờ thư đoạn code sau:

ActionScript
```
var a = 7;
trace(a++);
trace(a);
```

Ở output window bạn sẽ thấy 7 rồi mới tới 8. Ở dòng thứ 2, hàm trace sẽ cho ra giá trị của a trước rồi mới tăng giá trị của a lên thêm 1. Bây giờ xem ví dụ khác:

ActionScript
```
var a = 7;
trace(++a);
trace(a);
```

Ở output window bạn sẽ thấy 8 và 8. Ở dòng thứ 2 của ví dụ này, hàm trace sẽ tăng giá trị của a trước rồi mới cho ra giá trị của a. Tương tự như ++, AS cũng có cách viết tắt cho giãm giá trị của variable là dấu **-, -=, =-** và **--**. Phép tính nhân và chia thì chỉ có ***, *=, /, /=** thôi.

Điệu kiện

Trong AS và các ngôn ngữ khác, **if ... else** chính là các từ khoá của điếu kiện. Tiếng

việt có nghĩa là "nếu ... nếu không". Từ khoá **if** dùng kết quả của sự so sánh nào đó để đi tới quyết định kích hoạt một đoạn code. Đoạn code sau sẽ so sánh giá trị của variable a với 7, nếu đúng thì Flash sẽ chơi ở frame 10

```
ActionScript
if (a == 7) {
  gotoAndPlay(10);
}
```

Từ khoá **if** luôn bắt đầu cho 1 điều kiện và tiếp theo sẽ là sự so sánh. Luôn đặt code so sánh ở giữa ngoặc đơn (). Tất cả các code được kích hoạt nếu điều kiện được thoả mãn sẽ để trong đâu ngoặc móc {}.

Từ khoá **else** bổ sung cho **if** trong trường hợp bạn muốn thực hiện một đoạn code nếu điều kiện của **if** không được thỏa mãn. Ví dụ:

```
ActionScript
if (a == 7) {
  gotoAndPlay(10);
}  else {
  gotoAndPlay(15);
}
```

Nếu trường hợp cần thỏa mãn nhiều điều kiện thì có thể dùng cú pháp **if .... else if ... else**. Bạn có thể có bao nhiều cái **else if** cũng được.

Nãy giờ ta chi nói tới điều kiện dựa trên 1 sự so sánh, những AS cũng cho phép dùng so sánh đa hợp (compound comparision). Trong điều kiện của Flash ta có thể dùng nhiều so sánh để đi tới một kết quả chính xác hơn. Ví dụ nếu như a lớn 10 và a phải nhỏ hơn 15 thì ra lệnh cho Flash chơi ở frame 10.

```
ActionScript
if ((a > 10) and (a < 15)) {
  gotoAndPlay(10);
}
```

Từ khoá **and** (còn có thể được viết là **&&**) yêu cầu phải thoả mãn điều kiện của 2 phép so sánh. Bạn cũng có thể dùng từ khoá **or** (còn có thể được viết là ||)nếu như chỉ cần thoả mãn 1 điều kiện trong 2 phép so sánh thôi.

```
ActionScript
if ((a > 10) or (a < 15)) {
  gotoAndPlay(10);
}
```

Vòng lặp

Cú pháp của vòng lặp (loop) thì hơi rắc rối hơn so với cú pháp của điều kiên **if**. nhưng

nó tương tự như C, C++, Java. Gồm có 3 loại vòng lặp chính là **for loop, while loop** và **do-while loop**

for loop được coi là vòng lặp chính với từ khoá **for** và cú pháp của nó nhìn như sau:

```
ActionScript
for(var i=0;i<10;i++) {
  trace(i);
}
```

Nếu bạn cho chạy đoạn code trên thì bạn sẽ có được từ số 0 tới 9 ở output window. **for loop** tăng giá trị của variable **i** trong suốt quá trình thực hiện loop. Điều kiện của for loop có 3 phần chính, và được cách biệt bởi dấu chấm phẩy ( . Đầu tiên là điều kiện bắt đầu với việc tạo local variable cho for loop **var i=0**. phần này for loop chỉ thực hiện có 1 lần duy nhất lúc bắt đâu. Phần thứ hai là điều kiện chính của for loop **i<10**, sẽ được thực hiện ở mỗi vòng lặp. Nếu điều kiện này thoả mãn thì mới tiếp tục vòng lặp. Và phần cuối cùng là phép tính cho sự tiếp nối của for loop ở mỗi vòng lặp, ở đây là **i++** nên giá trị của i sẽ được tăng sau mỗi vòng lặp trước khi vòng lặp mới bắt đầu. 3 phần này đều được bỏ vào trong ngoặc đơn (). Còn những code thực hiện trong mỗi vòng lặp thì ở trong ngoặc móc {}. Bây giờ chúng ta giả làm computer và chạy cái for loop này.

1. Tạo local variable và ấn định giá trị cho nó bằng 0
2. Kiểm tra giá trị của i có nhỏ hơn 10 không, nếu nhỏ hơn 10 thì vòng lặp thực hiện các code ở trong {}, còn không thì tới bước thứ 5
3. hàm trace viết giá trị của i ở output window
4. cộng 1 vào giá trị của i, và trở về bước thứ 1
5. ra khỏi for loop

Tới bước thứ 5 thì giá trị của i sẽ bằng 10

while loop với từ khoá **while** có cú pháp nhìn tương tự như sau:

```
ActionScript
while (a > 0) {
  // code thực hiện trong while loop
}
```

Đơn giản hơn for loop, while loop tương tự như điều kiện if đơn giản, thực hiện các code trong vòng lặp nếu điều kiện của while loop được thoã mãn, vì vậy bạn dễ dàng rồi vào trường hợp vòng lặp vô hạn, và dẫn đến tình trạng treo máy. Trong đoạn code trên, bạn phải làm thế nào để cho co 1 lúc a sẽ nhỏ hay bằng 0 để cho vòng lặp ngừng lại.

Giống như while loop, **do-while loop** có cú pháp nhìn như sau:

```
ActionScript
```

```
do {
  // code thực hiện trong do-while loop
}  while (a > 0);
```

Chỉ khác với while loop là do-while loop thực hiện code trong vòng lặp trước rồi mới kiểm tra điều kiện. (while loop kiểm tra điều kiện rồi mới chạy code trong vòng lặp)

Muốn phá ra khỏi quá trình tự hành của 3 loại loop trên thì dùng lệnh **break** và **continue**. Lệnh **break** sẽ phá ra khỏi loop hoàn toàn trong khi lệnh **continue** thì chỉ phá ra khỏi vòng lặp hiện tại và bắt đầu vòng lặp mới. Tạo ra một ví dụ cụ thể rất phức tạp, nên chúng ta tạm khoan bàn tới 2 lệnh này cho tới các bài sau.

Hàm

Cho tới bây giờ các script của chúng ta đều được viết vào frame đầu tiên của moive, cách này chỉ tốt cho nhưng chương trình đơn giản, nhưng nếu mà chương trình trở nên phức tạp thì đây không phải là cách. Hàm (function) sẽ cho phép chúng ta cấu tạo và quản lý code dễ dàng hơn trong các chương trình phức tạp. Dưới đây là một function đơn giản:

ActionScript
```
function myFunction(num) {
  var newNum = num + 3;
  return newNum;
}
```

Fuction bắt đầu bằng từ khoá **function**, tên của function có thể là bất cứ chữ gì như cách bạn đặt tên cho variable, nhưng lưu ý cách đặt tên cho fuction làm sao để người ta đọc tên có thể biết được chức năng của function. Theo sau tên của function sẽ là thông số (parameter, cho ngắn mình sẽ gọi thông số là param trong các bài viết này) được để trong ngoặc đơn (). Function có thể co 1
hay nhiều param hay không cần param cũng được. Param chẳng qua chỉ la variable được dùng trong function, nhưng variable này được truyền tự ngoài vào khi function được gọi. Đoạn code được function thực hiện sẽ nằm giữa ngoặc móc {}. Bây giờ chúng ta phân tích đoạn code trên. Khi function myFunction được gọi, thì param num được truyền vào, sau đó function myFunction tạo ra một local variable tên là newNum, và ấn định giá trị của newNum bằng giá trị của param num cộng với 3. Sau đó myFunction dùng lệnh **return** để ấn định giá trị của newNum là kết quả của myFunction. Lệnh **return** là lệnh đặc biệt chỉ được dùng ở trong function mà thôi. Lệnh này sẽ kết thúc function.

Để sử dùng function này, dùng nó như một lệnh hay hàm của AS, giống như là trace() vậy đó. Đây là ví dụ: **var a= myFunction(7);**. Trước tiên tạo một local var rồi sau ấn định giá trị của local var này bằng kết quả của myFunction với param là 7, và cuối cùng thi local var này sẽ bằng 10.

Một trong ưu điểm của function là bạn có thể tái sự dụng. Dưới đây là hàng code dùng chung 1 fuction và cho 3 kết quả khác nhau

ActionScript

```
trace(myFunction(7));
trace(myFunction(13));
trace(myFunction(2));
```

Khi chạy đoạn code này thì output windown sẽ là 10, 16 và 5. Ví có thể tái sử dụng nên chúng ta chỉ cần thay đổi code ở trong myFunction thì tất cả các kết qủa có được từ gọi function này cũng thay đổi theo luôn.

Dot Syntax

Một điều bạn sẽ thấy rất nhiều trong quá trình học AS là **dot syntax** (hỗng biết dịch sao bây giờ) Dot syntax là phương pháp được dùng trong lập trình theo hướng đối tượng (oob: object oriented programming).

Đây là một ví dụ của dot syntax. Nếu bạn muốn lấy căn bình phương của một số, và trong Flash đã có sẵn function để làm chuyện này rồi, và hàm này thuộc về đối tượng toán, tên là Math. Vậy muốn gọi function này thì trước tiên bạn phải gọi tên đối tượng mà fucntion này trực thuộc, đó là **Math**, theo sau đó là dấu chấm (dot), và rồi tới tên của function đó là **sqrt**. Cách viết như sau:

ActionScript
```
var a = Math.sqrt(4);
```

Một cách dùng dot syntax khác là để truy cập đặc tính hay variable của một đối tượng, như là movie clip.

ActionScript
```
var a = myClip._x;
var a = myClip.myVariable;
```

Chúng ta sẽ đề cập đến đối tượng Math và MovieClip trong các bài tới, còn bây giờ thì quan trọng là bạn nắm được khái niệm về dot syntax.

Chú giải

Một trong đức tính cần có của lập trình viên là viết code làm sao cho dễ đọc và dễ hiểu. Nhưng nhiều khi dù khi viết có cố gắng cách mấy thì những đoạn code đó vẫn khó hiểu cho người xem, vì vậy mới cần những lời chú giải. Muốn chú giải trong AS thì chỉ cần dùng ký hiệu// trước câu chú giải đó. Nếu chú giải nhiều hơn vài dòng thì có thể dùng kiểu sau**/* ..... chú giải ....*/**.

ActionScript
```
/*
  chú giải:
  cộng 2 cho a
  cộng 2 cho b
*/
a += 2; // add 2 a
```

```
// add 2 to b
b += 2;
```

Debugging

Thường thì thời gian diệt bọ tốn gần 1/3 thời gian phát triển chương trình. Nếu bạn nắm vững được "nghê thuật" diệt bọ thì bạn sẽ rút ngắn được rất nhiều thời gian. Có 3 cách diệt bọ hữu hiện là: phân tích, viết message ra output window và dùng AS debugger.

Với các loại bọ đơn giản thì phân tích các đoạn code sẽ giúp bạn bắt trúng con bọ cần diệt. Dùng output window để theo dõi các đối tượng, vòng lặp, điều kiện, và nhờ vào đó bạn có thể biết được chổ nào trong code có vấn đề, cuối cùng là dùng AS debugger có sẵn trong Flash. Debugger window cho phép bạn có thể coi hết tất cả moi thành phần trong movie của bạn, cho phép bạn dùng AS bất chứ khi nào và chổ nào bạn muốn. Tham khảo thêm phần trợ giúp của Flash để biết cách dùng debugger.

Giờ thứ 05: Điều khiển luồng movie, Control the flow of the movie

Sử dụng AScript là cách điều khiển movie đơn giản nhất, nhung lại hiệu quả nhất. Chương này bạn sẽ học cách :

- làm sao để dừng Movie trên frame
- Sao để nhảy từ frame này đến frame kia
- tạo nút cho phép điều khiển movie
- tạo một slide show đơn giản
- tạo một biểu diễn đầy đủ..

Dừng movie
Đây là câu lệnh stop:

ActionScript
```
stop();
```

Khi bạn dùng câu lệnh này , movie chỉ tạm dừng ở frame mà bạn đặt câu lệnh. Các animation bên trong movieClip và các file đồ họa vẫn tiếp tục chạy trên frame đó. Nhưng animation của time line sẽ bị dừng.

Để cho animation của timeline tiếp tục chạy, ta sử dụng một lênh đơn giản đó là lệnh play.. ta sẽ bàn đến nó sau.

Ví dụ
Tại frame 1, bạn hãy thả một text box và viết vào đó Chữ A
Tại frame 2, bạn hãy thả một text box và viết vào đó Chữ B
Tại frame 3, bạn hãy thả một text box và viết vào đó Chữ C

Bây giờ bạn hay đặt câu lệnh stop(); vào Frame 2, bạn se thấy chữ A xuất hiện rất

nhanh rồi đến chữ B, nhưng chữ C sẽ ko thấy xuất hiện,, Đơn giản vì câu lệnh stop() đã dừng animation của timeline tại Frame 2.

Nhảy từ frame này đến frame kia

Một câu lệnh cơ bản của AS là câu lệnh gotoAndPlay. Câu lệnh này giúp bạn có thể nhảy từ frame này đến frame mà bạn muốn. Bạn có thể sư dụng số thứ tự của frame hay là tên của frame

ActionScript
```
gotoAndPlay(7);
gotoAndPlay(20);
gotoAndPlay("my frame label")
```

bạn có thể sử dụng nhiều movies , hay còn gọi là cảnh(scene) . Nếu Câu lệnh gotoAndPlay chỉ có một tham số, thì tham số đó là frame. Nhưng nếu có 2 tham số,thì tham số đầu tiền là tên của scene, còn tham sso thứ 2 là số thứ tự của frame hay tên frame.

ActionScript
```
gotoAndPlay("My Scene","My Frame");
```

Khi bạn dùng gotoandPlay,movies sẽ nhảy đến frame mà bạn gọi , và tiếp tục chạy, nhưng nếu bạn muốn nó nhảy đến frame và dừng lại, bạn co thể sử dụng gotoAndStop . Câu lệnh này dùng y hệt như gotoAndPlay, với lựa chọn 1 tham số hay 2 tham số.

Ngoài ra có thêm 2 câu lệnh cũng hoạt động giống gotoAndStop , đó là nextFrame and prevFrame

Để hiểu rõ hơn về các câu lệnh này, ta sẽ đi tiếp phần sau...

Tạo nút (button)

Nút là một trong 3 biểu tượng (symbol) chính trong Flash, 2 cái kia là movie cilps và hình

Tạo một button

Có nhiều cách để tạo nút. Một trong nhưng cách đó là chọn Insert, New Symbol từ Menu. Một hộp thoại sẽ hiện lên và hỏi bạn đặt tên và lựa chọn biểu tượng của bạn là loại Movie clip, button, hay đồ họa. Bạn hãy chọn Button.

Bây giờ trong cửa sổ chính của Flash sẽ thay đổi, timeline của button sẽ thay thế timeline của movie chính. Có 4 frames trong timeLine của button. Chúng có tên là Up, Over, Down, HIt. CHúng thể hiện 3 trạng thái của button, và vùng hoạt động của button.

Nếu bạn đặt một đồ họa, chẳng hạn là một vòng tròn, trong frame đầu tiên(UP) và ko có gì trong 3 frame sau, trạng thái Over và Down của button sẽ giống như trạng thái của Up. VÙng hoạt động của button cũng tương tự như vậy.

Mặt khác , bạn có thể tạo các hiệu ứng khác nhau cho button của bạn. VÍ dụ nếu bạn đổi mầu của vòng tròn là màu xanh ở Frame OVER,có nghĩa là khi bạn đưa chuột qua button, button sẽ chuyển sang màu xanh.

Khi bạn tạo xong 1 button, bạn trở lai movie timeline chính. Và button của bạn đã sẵn sàng trong thư viện của FLash. Bạn chỉ viêc bấm F11, tìm button của bạn và kéo thả nó vào nơi nào bạn cần trên cửa sổ thiết kế.

Tạo script cho button

Để đặt script cho button, đầu tiên hãy chọm một button. Sau đó vào cửa sổ Action bằng cách bấm chuột phải lên button và chọn Action từ menu xuất hiện.

Hãy Chắc chắn rằng cửa sổ Action đã được đặt ở chế đọ Expert. Bạn có thể kiểm tra băng một popup menu ở góc trên cùng bên phải của cửa sổ.

Đây là một đoạn mã thông thường cho button. Bạn có thể đặt nó trong cửa sổ mã và sau đó kiểm tra movie xem nó hoạt động thế nào.

ActionScript
```
on (release) {
  trace("You clicked the button!")
}
```

on : là một keyword, được gọi ra để xử lí một sự kiện
Release : Là một sự kiện, khi người sử dụng click vào button và nhả chuột ra. Đoạn mã trace sẽ được thực thi.

Ngoài ra Release có thể thay thế bằng sự kiện PRESS, sự kiện này chỉ khác Release ở chỗ khi người sử dụng bấm vào button, đoạn mã trace sẽ đươc thực thi mà ko cần phải thả chuột ra.

Trace : là in ra màn hình dòng chữ YOU CLICKED THE BUTTON.

Thực Hành: Tạo một Slide Show đơn giản...

Có lẽ ko phải diễn tả nhiều về Slide Show, các bạn hãy tưởng tượng giông Power Point.. Khi chúng ta bấm chuột, các slide sẽ chuyển sang slide khác.

1. Mở Flash, chọn New

2. Trong Layer 1, từ Frame 1 đến Frame 4, bạn hãy đăt các biểu tượng hay hình ảnh, text khác nhau để phân biệt được các frame

3. Sau đó bạn tạo một layer mới, hãy đặt một button lên layer này.Tiếp theo bạn chọn

Frame 4 của layer 2, và bấm F6. Điều này đảm bảo cho button của bạn sẽ xuất hiện trong tất cả các Frame của Layer 1.

Bây giờ copy đoạn mã sau vào button của bạn, chú ý nhớ là copy vào action của button, chứ ko phải vào bất kì frame nào

ActionScript
```
on (release) {
  nextFrame();
}
```

Cuối cùng chọn Frame 1 Layer 1 action và chèn đoạn code sau

ActionScript
```
stop();
```

Hãy chạy thử slide show của bạn = cách bấm Ctrl Enter.

5) Xây dựng một presentation đơn giản:
Presentation này sử dụng nguyên tắc giông hệt như ví dụ Slide show ở trên, các bạn hay down load file FLA để xem .. easy

Giờ thứ 06: Điều khiển movie clip, Controlling Movie Clips

6th Hour Điều khiển movie clip
Bạn sẽ học
1- ra lệnh cho movie clip
2-Tạo playback cho animation
3-TÌm hiểu đích(target) của movie clip
4-Tập viết code cho movie clip
5-Tạo movie clip có thể chạy lùi lại.

1) Ra lệnh cho movie clip :

Một movie chính có thể rất đơn giản. NÓ chỉ cần 1 frame và một movie clip. Nhưng bản thân movie clip lại có thể là một animation dài. Việc này dễ dàng thực hiện mà ko cần Actioncript.

Để điều khiển được movie, bạn phải đặt tên cho movie clip. Chú ý rằng một movies clip có 2 tên, một tên sử dụng cho timeline . Một tên sử dụng cho actioncript. 2 tên này có thể đặt trùng tên nhau hoặc khác tên nhau.

Làm sao để đặt tên cho movie clip :
Chọn insert -----> New Symbol (hoặc bấm Ctrl F8), đặt tên cho movies clip
(nhớ lựa chọn option movies clip, chứ ko phải là button hay graphic)
Tên này được sử dụng cho timeline. vd: gear animation

Thả movie gear animation vào cửa sổ thiết kế.. PHía dưới cửa sổ properties, ban sẽ

thấy box instance name.. Đặt tên cho movie
tên này được sử dụng cho Action script , vd : gears

và như vậy, khi bạn lập trình , tên sử dụng là gears , chứ ko phải gear animation.
bạn có thể sử dụng dấu chấm để đưa ra các lênh cho movie của bạn. ví dụ

ActionScript
```
gears.stop();
gears.gotoAndStop(5);
```

Nhớ rằng movie clip luôn có level. Nếu đoạn mã được đặt trên chính movieclip, bạn ko cần sử dụng tên movies, chỉ cần gotoAndStop();. Nếu bạn đặt cả tên movie vào, Flash sẽ tìm movie của bạn từ timeline chính.. Tức là sẽ phải qua 2 level. ---> chậm hơn chút.

2)Animation Playback Controller
Hãy down load source code từ trên phần download và xem. Về cơ bản, chúng ta làm các việc sau:

- Tạo một Movie ,đặt tên là gear animation
- Đặt instance name là gear(hướng dẫn ở trên)
- Frame đầu tiên của time line chính, chèn code

ActionScript
```
gears.stop();
```

-Tạo các button Advance, Previous, Play, Stop, and Rewind
-Code cho các button lần lượt là
Advance
ActionScript
```
on (release) {
   gears.nextFrame();
}
```

Previous
ActionScript
```
on (release) {
   gears.prevFrame();
}
```

Play
ActionScript
```
on (release) {
   gears.play();
}
```

Stop
ActionScript

```
on (release) {
```

```
    gears.stop();
}
```

Rewind

ActionScript
```
on (release) {
    gears.gotoAndStop (1);
}
```

3) Target một movie

Level cơ bản đầu tiên của Flash là time line chính(Level 0). Nếu bạn muốn ra lệnh cho time line này, bạn dùng code sau

```
_root.gotoAndStop();
```
thậm chí nếu bạn đặt code ở time line chính, bạn ko cần _root. Chỉ cần gotoAndStop();

Giả sử nếu bạn có một movie trên time line, tức là bạn có một level sâu hơn(level 1). Nếu bạn từ time line chính, muốn goi movie gears , bạn phải dùng câu lệnh

```
gears.gotoAndStop(7);
_root["gears"].gotoAndStop(7);
this["gears"].gotoAndStop(7);
```

Cả 3 cách trên đều giống nhau.. Tuy nhiên this có nghĩa là level hiện hành. VÍ dụ trên, level hiện hành của this là level 0. Nhưng nếu code đặt trong movie, level hiện hành sẽ là level 1. Bạn phải chú ý.

Giả sử bạn có một movie khác bên trong movie, tức là bạn có level 2. Nếu bạn muốn từ level 1 gọi đến level 2, bạn phải dùng _parent.
Parent dùng giống như root, nhưng khác ở chỗ, nó được gọi từ level ở trên. Còn root được gọi từ level 0 (tức là level gốc).

Thuận lợi từ việc sử dụng root, this là bạn có thể gán biến cho movie clips

ActionScript
```
var whichClipToUse = "gears";
this[whichClipToUse].stop();
```

Thuận lợi của việc sử dụng this nhiều hơn root ở chỗ , ko phải lúc nào mọi thứ cũng xảy ra ở time line chính, đôi khi ta chỉ muốn nó xảy ra ở một level nhất định. Bởi vậy this là cách tốt nhất để goi level của một movie. Tuy nhiên trong những trương hợp đơn giản, hay nhất là cứ đặt tên cho movies.

4) Mã cho movie clip

Chúng ta đã bàn về button ở trên, vậy về cơ bản movie clip script cũng giống button, đó là

```
ActionScript
onClipEvent (load) {
  trace("This clip has been loaded.");
}
```

onClipEvent : key word gọi sự kiện (giống on của button)
load : sự kiện này xảy ra khi movie clip xuất hiện lần đầu tiên trên màn hình . NÓ chỉ xảy ra một lần.

Nhớ rằng ngay cả khi time line chính bị đừng, movie clip nằm trên time line vẫn tiếp tục chạy. Sự kiên enterFrame xảy ra liên tục bên trong movie clip mỗi khi đến một frame mới. Sự kiện này sẽ thực thi đoạn mã một cách liên tục cho đến khi nó bị kết thúc.

```
ActionScript

onClipEvent (enterFrame) {
  trace("This clip has entered a new frame.");
}
```

bây giờ hãy copy 2 đoạn code trên vào của sổ action của timeline chính và run .. bạn sẽ thấy

```
ActionScript
This clip has been loaded
  This clip has entered a new frame
  This clip has entered a new frame
  This clip has entered a new frame
  This clip has entered a new frame
```

dòng chữ sẽ tiếp tục hiện ra cho đến khi bạn tắt movie.

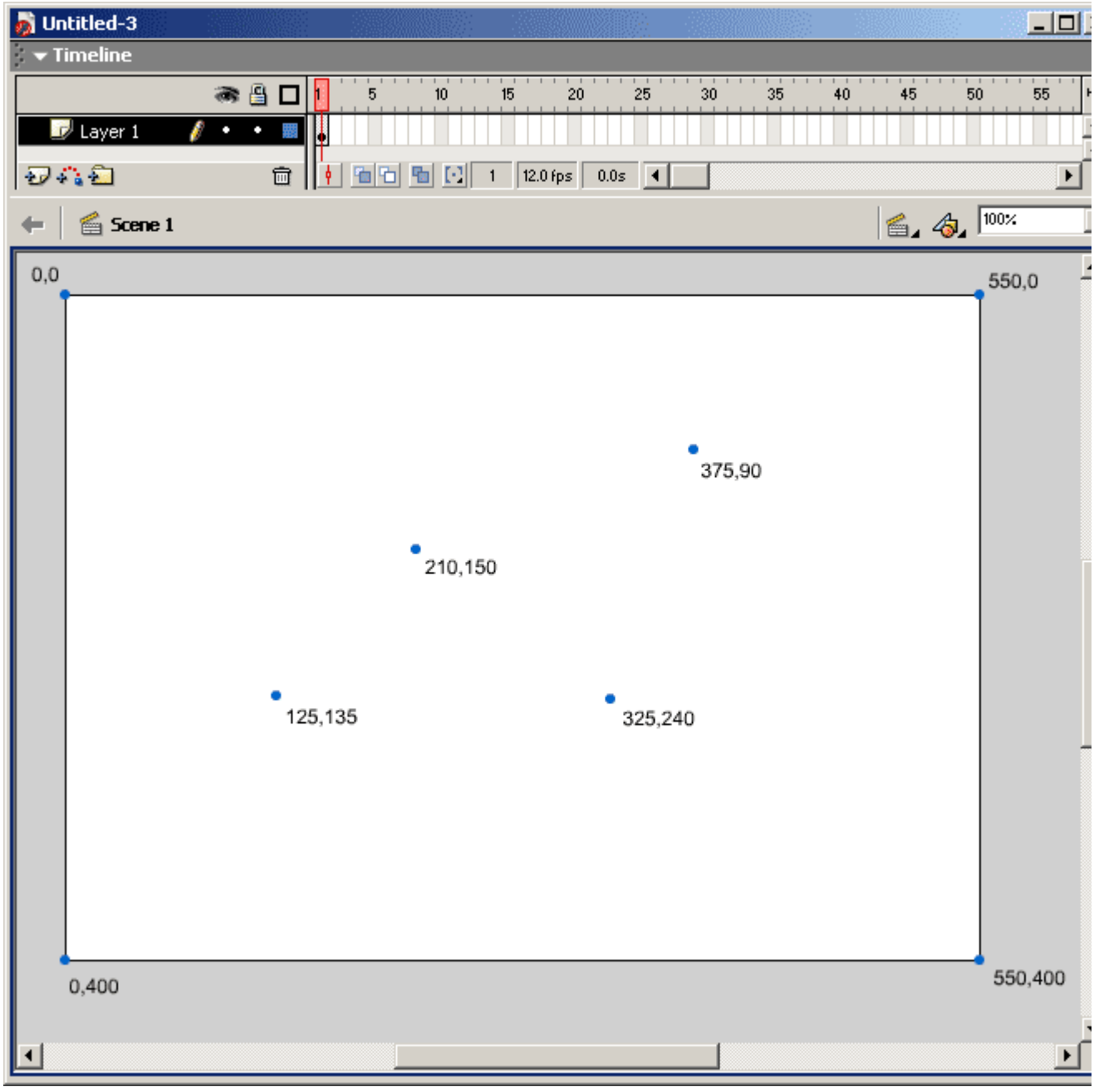
Giờ thứ 07: dịch chuyển và thay đổi movie, Moving and Changing Movie Clips

Một movie clip (MC) bao giờ cũng có các thuộc tính, cho phép bạn xác định vị trí, phóng to thu nhỏ, quay, thậm chí làm nó biến mất.
Chương này các bạn sẽ biết cách:
-thay đổi vị trí của mc
-xác đinh vị trí chuột
-làm sao để quay một mc
-làm sao để phóng to mc
-làm sao để movie invisible

1) Vị trí của MC:
Mọi thứ trong Flash đều có vị trí. VỊ trí này được đo bằng PIXEL. Góc trên cùng bên trái của màn hình là vị trí (0,0) .Nó là vị trí giao của hàng ngang và cột dọc.

Nếu bạn sử dụng một movie mặc định, thì góc dưới bên phải sẽ là 550,400. Có nghia là 550 pixel từ bên trái sang và 400 pixel từ bên trên xuống.

nhìn hình sau để rõ hơn.



2) Vị trí của Mouse:

Cũng như movie, mouse cũng có thể xác đinh được tọa độ.. CHú ý thuật ngữ mouse ở đây là nói đến con trỏ (cursor), chứ không phải con chuột bạn đang cầm trên tay.. ^^

Có hai thuộc tính cơ bản của con trỏ , đó là _xmouse và _ymouse.

Vậy _xmouse và _ymouse là thuộc tính của chính đối tượng mà chúng được gán. Nếu bạn sử dụng trên time line chính, thì chúng là thuộc tính của time line. Nếu bạn sử dụng chúng trên movie clip, thì nó là thuộc tính của movie clip. Nếu xmous y mouse là thuộc tính của time line chính, nó sẽ mặc định là góc trái trên cùng. Nếu xmouse ymouse là thuộcc tính của 1 movie clip, nó sẽ mặc định ở trung tâm của MC.

Hầu hết các trường hợp thuộc tính liên quan đến movie chính, bạn nên dùng cách _root._xmouse và _root._ymouse

Đây là một đoạn code ví du
CODE
```
onClipEvent (enterFrame) {
  trace(_root._xmouse);
  trace(_root._ymouse);
  trace("");
}
```

Khi bạn chạy movie, bạn sẽ thấy các cặp số được in ra, đó chính là tọa độ x,y mouse .Hãy dịch chuyển mouse lên góc trái trên, bạn sẽ thấy ở đó vị trí mouse là 0.0. hoặc góc phải dưới, tọa độ là 550 400.

Chú ý: khi bạn đưa chuột ra ngoài Flash window, tọa độ chuôt xmouse y mouse không thay đổi. Nếu bạn di chuyển chuột thật nhanh từ trung tâm của cửa sổ ra ngoài cửa sổ, các giá trị cũ vẫn ở đó cho đến khi bạn quay trở lại cửa sổ. VÌ vậy, bạn phải luôn lên kế hoạc cẩn thẩn trước khi dùng xmouse, y mouse

3) Làm movie quay

Một thuộc tính khac đó là thuộc tính _rotation

Thuộc tính rotation chấp nhận giá trị là góc độ. Một vòng tròn chia ra 360 độ, miền giá trị cả thuộc tính rotation là -180 và 180. Bạn có thể sử dụng giá trị integer hoạc floating point.

Giá trị của _rotation luôn luôn ở trong vòng -180 và 180, ví dụ 179, hoặc -179. Nhưng nếu bạn đặt nó là 181, nó sẽ hiểu thành góc độ -179.

vậy để quay một movie, đơn giản là đặt giá trị cho nó, bạn cũng có thể sử dụng các biểu toán ++, += để thay đổi giá trị . Hãy xem ví dụ sau
CODE

```
myClip._rotation = 90;
myClip._rotation++;
_root["myClip"]._rotation = 45;
this._rotation += 0.5;
```

4) Co giãn đàn hồi một MC:

Bạn có thể làm co giãn, thay đổi chiều dài chiều rộng của movie clip
Scale thuộc tính

THuộc tính để làm việc này là _xscale cho chiều ngang và _yscale cho chiều dọc.

Các giá trị được gán cho x,y scale là phần trăm. Có nghĩa giá trị 100 là 100 phần trăm , đây là thuộc tính mặc đinh cho một movie clip gốc. Bạn có thể sử dụng các số nhỏ hơn như 50 để làm movie co lại. Hoặc có thể sử dụng số to hơn, 200 để giãn movie ra. Thậm có thể dùng các giá trị âm để lật movie.

ví dụ 07mousesclae.fla chứa một đoạn mã mà chúgn ta sẽ gặp rất nhiều sau này. Nó kiểm tra tọa độ của mouse,. Sau đó xác đinh từ mouse đến trung tâm movie xa bao nhiêu. Rồi nó sử dụng khoảng cách ,cả chiều cao chiều rộng để tính phần trăm tỉ lệ cho movie clip.
CODE

```
onClipEvent (load) {
  // get the original width and height of the mc
  origWidth = this._width;
  origHeight = this._height;
}
onClipEvent (enterFrame) {
  // get the distance from the center of the mc to the mouse
  dx = _root._xmouse-this._x;
  dy = _root._ymouse-this._y;

  // calculate the percentage of scale
  sx = 100*dx/(origWidth/2);
  sy = 100*dy/(origHeight/2);

  // set the scale of the mc
  this._xscale = sx;
  this._yscale = sy;
}
```

Chú ý ở đây ta sử dụng 2 thuộc tính mới, _width và _height trả về chiều cao và chiều rộng của movie clip bằng Pixel. CHúng được lưu giú trong onClipEvent (load) vì để khi ta cần lấy lại giá trị gốc của movie.

Thuộc tính _width, _height
Sự khác nhau giữa **scale** và **width,height** í scale sử dụng giá trị phần trăm. Còn width height sử dụng pixel . Movie có thể có giá trị sau: width 75, height 45, nhưng scale cho cả xscale and yscale là 100 phần trăm.

Dưới đây là một ví dụ sử dụng width height thấy thế cho xscale, yscale.

CODE

```
onClipEvent (enterFrame) {
  // get the distance from the center of the mc to the mouse
  dx = _root._xmouse-this._x;
  dy = _root._ymouse-this._y;

  // set the scale of the mc
  this._width = dx*2;
  this._height = dy*2;
}
```

bạn có thể thấy code này ngắn gọn hơn ơ trên. bởi vì nó ko sử dụng sự kiên onClipEvent(load) bởi gái trị chiều dài chiều cao của movie gốc ko cần lưu trữ.Đây là một ví dụ cho thấy sử dụng width và height tiện hơn dụng xscale ,y scale.

5) Visibility

Một thuộc tính khác của MC là thuộc tính _visible , giá trị của thuộc tính này là giá trị boolean true , false.

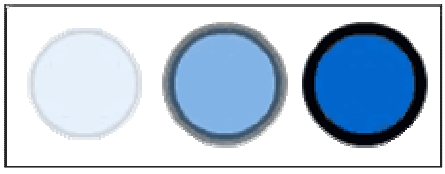
CODE
```
myClip._visible = false;
```

hãy xem ví dụ 07visible.fla để rõ hơn.

Giả sử ta không muốn movie biến mất hẳn, mà chỉ bị mờ đi, ta sử dụng thuộc tính _alpha . Thuộc tính có giá trị từ 0 đến 100
CODE
```
myClip._alpha = 50;
```
Thuộc tính alpha là kênh màu thứ 4, tên là ***alpha channel*** . 3 kênh đầu tiên là red ,green, blue đủ để tạo ra 7 màu cầu vồng. Khi bạn nghe đến đồ họa 32 bit, có nghĩa là nó đủ 4 kênh màu.. Còn 24 bit có nghĩa chỉ có 3 kênh đầu tiên.

Nếu giá trị của _visible là 0 thì movie là trong suốt hoàn tòan và có thể nhìn thấy tất cả mọi thứ phía sau movie.

Nhìn hình sau để xem cùng 1 movieclip nhưng với 3 giá trị alpha khác nhau, 10, 50 và 100 (từ trái sang)



Giờ thứ 08: chọn, kéo thả movie clip, Moving and Changing Movie Clips

1. Selection

Bạn đã biết cách học sao để tạo một button và cho phép người sử dụng click vào làm sự kiện hoạt động. Tuy nhiên ở đây ta sẽ học cách click vào để chọn lựa một đối tượng trên màn hình.
Tạo một chọn lựa, có nghĩa là user click vào item , nhưng không có gì xảy ra. Như vậy, người dùng có thể lựa chọn và thay đổi lựa chọn.
Chúng ta sẽ sử dụng lụa chọn như bước đầu tiên để học kéo thả movie.

Button ở bên trong một Movie method.

Một movie không thể phản ứng với mouse click. Không thể sử dụng các sự kiện như on (release) hay (Press). vì vậy ta phải tìm cách đánh lừa nó. Đó là sử dụng một button bên trong movie. Button có thể xử lý được mouse click, và miễn là nó có độ lớn đủ để bao trùm một movie.
Hãy xem ví dụ 08buttoninmc.fla . Ta chỉ nhìn thấy một movie clip trên màn hình.nhưng thực chất bên trong của movie clip là một button.
Để tạo ra khả năng select, ta phải tạo nhiều frame.Frame đầu tiên chứa một button tên là offbutton. Button này có mã là

CODE
```
on (release) {
  this.gotoAndStop(2);
}
```

bằng this, button sẽ gọi đến movie clip mà nó nằm trên đấy. Frame thứ 2 chứa nút tương tự nhu frame 1 , tên là onbutton, nút này có mầu hơi sáng hơn để người dùng nhận ra khi nó được chọn.

CODE
```
on (release) {
  this.gotoAndStop(1);
}
```

cuối cùng đặt vào frame 1
CODE
```
stop();
```

Bây giờ hãy chạy để xem.

Phương thức **hitTest**

Có cách khác để làm movie có thể sử lý mouse click mà không cần button, đó là sử dụng sự kiện onClipEvent(mouseDown) hoặc là onClipEvent(mouseUp)ví dụ
CODE
```
onClipEvent (mouseUp) {
  this.gotoAndStop(2);
}
```
Hãy chạy ví dụ 08twomcs1.fla để xem.Bạn sẽ thấy tại sao onClipEvent(mouseUp) khác với on (Release) . Nếu bạn click vào movie, cả hai đều phản ứng.Bởi vì tất cả movie clip nhận sự kiện mouse up được gửi đến cho chúng.

Quyết đinh movie nào được click.
Có một cách để click vào movie mình muốn. Đó là sử dụng vị trí chuột để xác định. Sửa đoạn mã thành như sau

CODE
```
onClipEvent (mouseUp) {
  if (this.hitTest(_root._xmouse, _root._ymouse)) {
    this.gotoAndStop(2);
  }
}
```
vậy bằng cách ràng buộc movie clip với mouse location , ta có thể click movie ta muốn.

VIết mã cho selection:
Chúng ta phải cho phép User click lên movies nhiều lần và thay đổi trạng thái của MC từ bật sang tắt, hay tắt sang bật.
Đoạn mã sau có thể xác định movie đang ở trạng thái nào và gửi nó đến frame khác. VIệc này thực hiện bằng thuộc tính _currentFrame . Thuộc tính này sẽ đọc giá trị 1 khỉ nó ở frame thứ nhất, giá trị 2 khi nó ở frame thứ 2.

CODE
```
onClipEvent (mouseUp) {
  if (this.hitTest(_root._xmouse, _root._ymouse)) {
    if (this._currentFrame == 1) {
      this.gotoAndStop(2);
    } else {
      this.gotoAndStop(1);
    }
  }
}
```
Bây giờ bạn đã thấy các cách để tạo sự chọn lựa movie. Cách thư nhất tiện lợi nếu bạn muốn dùng cả các trang thái khác như Over,up , down.. Cách thứ hai thì không cần nhiều biểu tượng trong thư viện.

Enjoy..

2) Kéo một movie clip
Bây giờ ta sẽ học kéo một movie.
Có 2 cách để kéo movie, cách thứ nhất là sử dụng lệnh . Cách này rất dễ sử dụng nhưng nó không cho bạn nhiều lựa chọn trong việc sửa đổi và giám sát movie.

Cách Drag cơ bản:
2 câu lệnh drag co bản là :startDrag và stopDrag. bạn dùng startDrag khi bạn muốn movie của bạn đi theo chuột, còn stopDrag khi bạn muốn movie của bạn không theo chuột nữa. Đây là đoạn code cơ bản :

CODE
```
onClipEvent (mouseDown) {
  if (this.hitTest(_root._xmouse, _root._ymouse)) {
```

```
    this.startDrag();
  }
}

onClipEvent (mouseUp) {
  if (this.hitTest(_root._xmouse, _root._ymouse)) {
    this.stopDrag();
  }
}
```

Sự kiện onClipEvent nhìn tương tự như ta đã làm ở trên. Chúng sử dụng hitTest để nhận biết movie nào đang được click. Và sau đó câu lệnh startDrag và stopDrag được sử dụng. Khi chuột được bấm xuống, câu lệnh startDrag thực thi. Và ngược lại, khi chuột thả ra, stopDrag thực thi.

Hãy xem movie, bạn sẽ thấy với các câu lệnh drag, bạn có thể kéo các movie một cách độc lập. Nhưng chú ý rằng bạn không thể kéo 2 movie cùng một lúc. Chỉ một movie được kéo tại một thời điểm. Vì vậy bạn nên cần cách drag phức tạp và linh động hơn để làm ứng dụng hay game.

Note: startDrag có 5 thám số. Tham số thứ nhất là giá trị True(hoặc False) để quyết định việc trung tâm của movie clip có khóa với vị trí chuột hay không. Các tham số còn lại là left, top, right, bottom. Nếu bạn đặt chúng, movie của bạn chỉ có thể được kéo theo các ràng buộc đó.

Kéo nâng cao

Bây giờ ta học cách kéo movie mà ko dùng câu lệnh startDrag và stopDrag. Vậy làm thế nào??

Bạn cần có 4 phần mã sau trong movie của bạn:
Thứ nhất là sự kiện onClipEvent(load), sự kiện này sẽ đặt cho biến ***global*** tên là dragging giá trị false. Khi biến này true, tức là báo hiệu rằng movie có thể được kéo.

Thứ hai, người dùng click vào một movie. Có nghĩa là dragging = true .
Thứ ba, quá trình kéo sẽ tiếp tục diễn ra bởi sự kiện onClipEvent(enterFrame). Sự kiện này chỉ đơn giản gán _x và _y (thuộc tính của movie clip) cho _root._xmouse và _root._ymouse(thuộc tính của con trỏ).
Thứ 4, nếu người dùng thả chuột , thì biến dragging được gán false.
Sau đây là đoạn code đầy đủ
CODE

```
onClipEvent (load) {
  // start out not dragging
  dragging = false;
}
```

```
onClipEvent (mouseDown) {
  if (this.hitTest(_root._xmouse, _root._ymouse)) {
     // follow the mouse from now on
     dragging = true;
  }
}

onClipEvent (enterFrame) {
  if (dragging) {
     // set to location of the mouse
     this._x = _root._xmouse;
     this._y = _root._ymouse;
  }
}

onClipEvent (mouseUp) {
  if (this.hitTest(_root._xmouse, _root._ymouse)) {
     // don't follow the mouse any longer
     dragging = false;
  }
}
```

Trong ví dụ 08complexdrag.fla chứa 2 MC. Kiểm tra movie, nhưng chỉ sử dụng MC bên trái. Đây là MC làm theo đoạn code trên. Chú ý xem trung tâm movie được khóa với vị trí chuột như thế nào. Nó làm movie nhảy ngay lập tức nếu bạn click vào nó.

Thông thường, khi bạn muốn kéo MC đó lên màn hình, bạn không muốn nó nhảy tới để làm khớp vị trí chuột với trung tâm của MC. Hãy kéo file xung quanh màn hình.Bạn sẽ thấy khi bạn click chuột vào bất kì điểm nào trên movie, movie sẽ lập tức làm khớp vị trí click chuột với trung tâm movie. Điều này cũng tương tự với câu lệnh startDrag .

Để tránh điều này, chúng ta chỉ cần thêm một đoạn code nhỏ. Khi người sử dụng bấm chuột, chúng ta sẽ lấy offset, đây là khoảng cách từ chuột cho đến trung tâm của movie. Sau đó, thay thế bằng việc gán vị trí chuột đến trung tâm movie, ta se thêm vào offset để cho MC luôn luôn xuất hiện offset bởi cùng một giá trị khi người dùng kéo.

Tưởng tượng , ví dụ người dùng click vào một điểm cách trung tâm movie 5 pixel. Sau đó ta luôn muốn khoảng cách 5 pixel đó được duy trì. Điều này làm cho người dùng có thể click vào bất cứ điểm nào để kéo movie.
Đây là code cho movie 2(bên phải)

CODE

```
onClipEvent (load) {
  // start out not dragging
```

```
  dragging = false;
}

onClipEvent (mouseDown) {
  if (this.hitTest(_root._xmouse, _root._ymouse)) {
     // follow the mouse from now on
     dragging = true;

     // get the mouse offset
     xOffset = this._x - _root._xmouse;
     yOffset = this._y - _root._ymouse;
  }
}

onClipEvent (enterFrame) {
  if (dragging) {
     // set to location of the mouse
     this._x = _root._xmouse + xOffset;
     this._y = _root._ymouse + yOffset;
  }
}

onClipEvent (mouseUp) {
  if (this.hitTest(_root._xmouse, _root._ymouse)) {
     // don't follow the mouse any longer
     dragging = false;
  }
}
```

Hãy xem movie để thấy khác biệt giữa 2 đoạn code.

3) Kéo và thả

Bây giờ ta đã biết kéo movie, vậy ta sẽ thả nó ở đâu.?
Vấn đề là bạn muốn theo dõi hành động của người sử dụng và xác đinh người sử dung đặt movie ở đâu.
Basic Drop Zone

Chức năng hitTest có thể được sử dụng để xác định khi MC bị đè lên nhau. Trong đoạn mã sau, chúgn ta sử dụng lệnh startDrag để đoạn mã kéo thả đơn giản đi, nhờ đó ta có thể tập trung vào các chức năng mới.

Trong đoạn mã sau, chức năng hitTest được sử dụng để so sánh MC được kéo với một movie khác tên là dragZone ở trên một lêvl(root level)

CODE

```
onClipEvent (mouseDown) {
  if (this.hitTest(_root._xmouse, _root._ymouse)) {
```

```
      this.startDrag();
   }
}
onClipEvent (mouseUp) {
   if (this.hitTest(_root._xmouse, _root._ymouse)) {
      this.stopDrag();

      // see if this mc is inside the dropZone mc
      if (this.hitTest(_parent.dropZone)) {
         trace("Dropped in zone");
      } else {
         trace("Dropped outside zone");
      }
   }
}
```

Chú ý hitTest không được sử dụng như các bài trước, chúng ta thay thế vị trí _x và _y bởi một movie khác.

Loại hitTest này so sánh vị trí và vùng được bao phủ của 2 movie clip. Trong trường hợp này, nó được so sánh với _parent.dropZone. Nếu có 2 movie đè lên nhau, hitTest trả giá trị True. Để xác đinh xem MC nào đang tồn tại, hình chữ nhật của cả 2 movie đều được sử dụng. Điều này có nghĩa là khi bạn sử dụng 2 hình tròng, như trong ví dụ, bản thân chính các vòng tròn đó không được chạm vào miễn là các hình chứ nhật của 2 movie được dùng.

Hãy xem hình minh họa. Tất cả các movie Dragme đều nằm đè trên DropZone.

Có 2 cách sử dụng hitTest. Thứ nhất là sử dụng tọa độ x,y. Chúng ta có thể sử dụng movie clip như vị trí x,y, rồi sau đó sử dụng dropzone như một movie clip thứ nhất.

CODE
```
if (_parent.dropZone.hitTest(this._x,this._y)) {
```

Với đoạn code này, thay thế ví dụ trên, movie clip sẽ nằm ở bên trong hình chữ nhật của vùng dropzone.
Chúng ta có thể thêm một bước nữa với hittest. Bằng cách thêm tham số thứ 3 cho hitTest, chúng ta có thể dùng hitTest để xem hình dạng chính xác của movie clip và xác đinh vị trí x,y. Biến thứ 3 này cần gán true nếu bạn muốn hành vi này. Nếu nó false, nó se hành động giống như chức năng hitTest bình thường. Đây là đoạn code

CODE
```
if (_parent.dropZone.hitTest(this._x,this._y,true)) {
```

Bây giờ movie kéo thả của ta hoạt động tốt hơn. Nếu bạn sử dụng ví dụ 08drop.fla, movie bên phải sẽ sử dụng đoạn mã gôc, và nó nằm đè ở trên hình chữ nhật của dropzone. Các movie khác sử dụng đoạn code phức tạp hitTest và chúng có trung tâm nàm bên trong hình dạng gốc của dropzone.

Giờ thứ 09: Lấy thông tin từ người dùng, Getting input from the user

Cho tới bây giờ, qua 8 giờ làm quen với AS, tất cả các tác động của bạn lên trên flash đều được làm qua chuột. Nhưng ngoài chuột, bạn có thể dùng bàn phím để nạp thông tin vào Flash.

Làm sao để phát hiện khi người dùng bấm phím

Có 3 cách để phát hiện khi một phím tên bàn phím bị nhấn. 1 là dùng nút, 2 là dùng đối tượng Key, và 3 là dùng "listener" (chỉ có ở Flash MX và MX 2004)

1. Phát hiện qua nút:

Bạn có thể dùng nút để phát hiện khi người dùng sử dụng bàn phím. Bạn chi cần dùng lệnh xử lý sự kiện **on** cho đoạn code của nút . Ví dụ đoạn code sau sẽ kích hoạt khi mà người dùng nhấn phím "a".

```
ActionScript
on (keyPress "a"){
   trace ("Key 'a' pressed");
}
```

Trong movie mẫu 09keybutton.fla có một nút đơn giản nằm trên màn hình và vài ví dụ trong đó. Nếu bạn cho chạy thử movie và nhấn phím "a" thì Output window sẽ có viết ra dòng chữ **"Key 'a' pressed."**. Xử lý dữ kiện **on(keyPress)** nhớ là phải phân biệt dạng chữ (case-sensitive) nên nếu bạn chỉ code cho nó nhận phím "a" thì nó sẽ không thể nhận được phím "A". May mà chúng ta có thể dùng bao nhiều bộ xứ lý dữ kiện cũng được.

Nếu bạn muốn xữ lý các phím khác, như các phím mũi tên, enter, thanh dài (space bar) ... thì bạn phải dùng các code dành riêng cho chúng. Vì dụ bạn muốn xữ lý phím mũi tên trái (left arrow key) thì dùng đoạn code sau:

```
ActionScript
on (keyPress "<Left>") {
   trace("Left pressed.");
}
```

Và sau đây là các code dành cho các phím đặc biệt:

- <Right>
- <Left>
- <End>
- <PageUp>

- <Insert>
- <PageDown>
- <Down>
- <Up>
- <Delete>
- <Tab>
- <Backspace>
- <Escape>
- <Home>
- <Enter>
- <Space>

Bạn có thể hợp nhiều dữ kiện lại chung với nhau. Ví dụ bạn có một nút và nút đó có phím tắt là "b". Vậy khi bạn nhấn nút đó hay là nhấn phím "b" thì code của nút đó sẽ được kích hoạt.

ActionScript
```
on (keyPress "b", release) {
    trace("'b' pressed or button clicked.");
}
```

Các ví dụ trên đều có trong movie mẫu 09keybutton.fla

2. Phát hiện qua Đối tượng phím (key object)

Mặc dù nút rất hữu ích trong việc phát hiện khi người dùng nhấn phím, nhưng nút lại không thể phát hiện được trường hợp khi người dùng nhấn phím và không nhất tay. Ví dụ nếu bạn làm một game bằng Flash, và trong đó người chơi cho thể làm cho các nhân vật trong game di chuyển liên tục nếu như họ nhấn phím mũi tên và không nhất tay lên.

Cho những trường hợp như vậy, bạn phải dùng đối tượng Key. Đối tượng Key là tập hợp của một số hàm (function) và hằng số (constant) được xây dựng sẵn trong Flash. Bạn có thể dùng các hàm và hằng số này để biết được các phím đang bị nhấn hay không. Ví dụ, nếu muốn kiểm tra coi phím mũi tên trái có bị **đang** nhấn hay không thì dùng đoạn code sau:

ActionScript
```
if (Key.isDown(Key.LEFT)) {
  trace("The left arrow is down");
}
```

Hàm **Key.isDown** sẽ cho ra kết quả đúng hay sai phụ thuộc vào thông số có phải là phím đang bị nhấn hay không. Hằng số **Key.LEFT** tượng trưng cho phím mũi tên trái. vậy khi mũi tên trái bị nhấn thì output window sẽ cho ra hàng chữ "The left arrow is down". Dưới đây là các hằng số tương tự như **Key.LEFT** đề cập ở trên

- Key.BACKSPACE
- Key.ENTER
- Key.PGDN

- Key.CAPSLOCK
- Key.ESCAPE
- Key.RIGHT
- Key.CONTROL
- Key.HOME K
- ey.SHIFT
- Key.DELETEKEY
- Key.INSERT
- Key.SPACE
- Key.DOWN
- Key.LEFT
- Key.TAB
- Key.END
- Key.PGUP
- Key.UP

Nếu bạn muốn kiểm tra xem nếu những phím bình thường đang bị nhấn thì bạn phải dùng **Key.getCode** để lấy mã số của phím đó để dùng trong hàm **Key.isDown**. Đây là code mẫu khi bạn muốn kiểm tra phím "a"

ActionScript
```
if (Key.isDown(Key.getCode("a")) {
  trace("The left arrow is down");
}
```

Cho 2 ví dụ trên, bạn có thể coi source Fla 09keyobject.fla

3. Key Listener

Key listener được dùng để quan sát bàn phím và thông báo cho Flash khi phím được nhấn.

Ở cách thứ 2 chúng ta dùng đối tượng Key để kiểm tra coi phím có bị nhấn hay không, nhưng nó không thể biết chính xác lúc nào thì phím bị nhấn. Nếu mà người dùng nhấn phím quá nhanh, trước khi **onClipEvent(enterFrame)** kịp kích hoạt đối tượng key để kiểm tra thì Flash sẽ không bao giờ biết được người dùng đã nhấn phím. Một bất tiện nữa khi dùng cách thứ 2 là nếu bạn muốn người dùng nhấn phím nhiều lần thì đối tượng Key sẽ không phân biệt được đó là 1 lần nhấn dài hay là nhiều cái nhấn thật nhanh.

Ví vậy ở Flash MX, MM cho chúng ta thêm một lựa chọn nữa là Key listeners. bạn có thể dùng "listener" (có thể hiểu nó như một quan sát viên) để theo dõi sự kiện của bàn phím và thông báo cho Flash ngay khi phím được nhấn.

Listener có 2 phần. Phần đầu là phần tạo listener. Bạn phải ra lệnh cho listener này chú ý vào sự kiện của bàn phím. Đây là code cho tạo listener

ActionScript
```
Key.addListener(_root);
```

Lệnh **Key.addlistner** tạo listener cho đối tượng trong thông số là **_root**. Và đối tượng này sẽ được thông báo về sự kiện của bàn phím.

Trong đoạn code trên, **_root** là đối tượng được thông báo về sự kiện của bàn phím. Nhưng khi nhận được thông báo thì **_root** phải làm gì? Vì vậy chúng ta cần phải viết code xử lý sự kiện cho **_root**. Ví dụ đoạn code sau sẽ được thực hiện khi **_root** nhận được thông báo về sự kiện bàn phím.

ActionScript
```
_root.onKeyUp = function() {
  trace(Key.getAscii());
}
```

Khi người dùng nhấn phím, rồi nhất tay lên thì sẽ tạo ra sự kiện **onKeyUP**, và sự kiện này được thông báo tới **_root** (bạn có thể thay thế _root bằng bất kỳ đối tượng nào) và sau đó thì hàm **Key.getAscii()** sẽ cho ra kết quả là mã số ASCII tương ứng với phím vừa được nhấn, ví dụ A = 65, B = 66 .v.v

Bạn có thể xem cái fla mẫu 09keylistener.fla, trogn đó có đoạn code trên (lưu ý là trong fla này sẽ không có gì hết ngoài đoạn code AS ở frame đầu tiên)

Nếu bạn muốn biết phím được nhất là gì thay vì ASCII code thì bạn có thể dùng **String.fromCharCode()** để cho ra kết quả bạn muốn tìm. Thay dòng **trace(Key.getAscii());** bằng **trace(String.fromCharCode(Key.getAscii()));**

Thật ra ngoài 3 cách trên, còn một cách nữa là dùng bộ xử lý sự kiện **onClipEven** của movie clip để theo dỏi sự kiện keyDown, keyUp, nhưng cách này không còn được dùng trong Flash MX nữa, vì vậy bạn chỉ có thể cách này trong các Flash trước MX thôi.

Bài Tập: Dùng phím để di chuyển movie clip

1. Tạo một file Flash mới
2. Tạo một movie clip đơn giản
3. Gắn đoạn code này vào movie clip đó

ActionScript
```
onClipEvent(enterFrame) {
  if (Key.isDown(Key.LEFT)) this._x -= 5;
  if (Key.isDown(Key.RIGHT)) this._x += 5;
  if (Key.isDown(Key.UP)) this._y -= 5;
  if (Key.isDown(Key.DOWN)) this._y += 5;
}[/b]
```
Đoạn code trên kiểm tra 4 phím mũi tên và di chuyển movie clip theo hướng của mũi tên. Bạn hãy thử movie coi sao

4. Thay đổi đoạn code trên cho hoàn chỉnh hơn: tạo 2 biến [b]x, y[/b] có giá trị tương đương với vị trí ban đầu của movie clip, và lập giá trị cho biến [b]speed[/b] bằng 5. Mỗi frame của movie, mình sẽ kiểm tra 4 phím mũi tên và điều chỉnh giá trị của 2

biến [b]x,y[/b] chứ không điều chỉnh vị trí của movie clip, sau đó phối hợp với biến [b]speed[/b] để tìm vị trí mới cho movie clip, rồi mới di chuyện movie clip tới đó:
[code]onClipEvent(load) {
  x = this._x;
  y = this._y;
  speed = 5;
}

onClipEvent(enterFrame) {
  if (Key.isDown(Key.LEFT)) {
    x -= speed;
  }
    if (Key.isDown(Key.RIGHT)) {
    x += speed;
  }
    if (Key.isDown(Key.UP)) {
    y -= speed;
  }
    if (Key.isDown(Key.DOWN)) {
    y += speed;
  }

  this._x = x;
  this._y = y;
}

Đoạn code trên có 2 lơi điểm. Thứ nhất, chúng ta tìm vị trí mới của movie clip trước khi chúng ta di chuyển movie clip tới đó, và như vậy chúng ta có thể kiểm tra vị trí mới có hợp lý không (áo dụng rất nhiều trong game), và thứ hai là rất tiện cho chúng ta thay đổi tốc độ di chuyển của movie clip, chỉ cần thay đổi giá trị của biến **speed**

Nhập văn bản

Trong Flash, người dùng có thể nhẫp văn bản vào các khung, và bạn có thể dùng AS để lấy những văn bản đó. Để cho người dùng có thể nhập văn bản, bạn phải tạo khung **input text** và tạo cho 1 varible để tương ứng với gía trị trong khung input text đó. (nếu bạn nào chưa biết tạo input text thì có thể download 3 cái CD hướng dẫn ơ bên box tài liệu để tham khảo thêm)

Các chức năng và thao tác làm việc với chuỗi ký tự (string

Chúng ta có rất nhèiu thao tác với chuỗi ký tự:

1. Ghép 2 chuỗi lại với nhau: dùng ký hiệu +. Ví dụ bạn có variable tên là myVariable có giá trị là "Hello", và muốn ghép chữ "world" vào sau đó thì dùng như sau: **myVariable = myVariable + "world"** và kết quả có được sẽ là "Hello world". Bạn cũng có thể ghép nhiều hơn 2 chuỗi lại với nhau bằng cách trên.

2. Substrings. Substring là một chuỗi ký tự nhỏ trong 1 chuỗi ký tự khác. Ví dụ "ell"

là substring của "hello world" hay là "hello" hay là "elle" hay "hell".

Bạn có thể lấy bất cứ substring nào của 1 string với cú pháp sau: **String.substring(start, end));**. String là đối tượng chuỗi mà bạn muốn trích ra một phần, start là số thứ tự của ký tự bắt đầu cho chuỗi bạn muốn lây, và end là số thự tự bắt đâu cho phần bạn không muốn lấy. Lưu ý rằng ký tự đầu tiê sẽ có số thự tự là 0. Ví dụ với đoạn code sau, output window sẽ cho ra hàng chữ "lo W"

ActionScript
```
var myString = "Hello World.";
trace(myString.substring(3,7));
```

Ngoài ra còn 1 cách nữa dùng tương tự như cách trên nhưng chỉ khác về thông số và có cú pháp như sau: **String.substr(start, length);**. thông số 1 cho số thứ tự của chữ cái đầu tiên và thứ hai cho chiều dài của chuỗi. Đoạn code sau cũng sẽ có kết quả tượng tự như ở cách 1

ActionScript
```
var myString = "Hello World.";
trace(myString.charAt(6));
```

Các hàm của đối tượng String

Sau đây là một số hàm tiêu biểu của đối tượng String.

1. indexOf: dùng để tìm số thứ tự của một ký tự hay một chuỗi nhỏ trong đối tượng String. Nếu hàm indexOf tìm không thấy ký tự hay chuỗi nhỏ thì sẽ cho ra kết quả -1. Đây là cú pháp:**myString.indexOf(substring,start));** substring là ký tự hay chuỗi mình muốn tìm, start là số thự tự mình bắt đầu tìm trong đối tượng String. Ví dụ:

ActionScript
```
var myString = "Hello World.";
//output: 6
trace(myString.indexOf("W",0));
//output: 2
trace(myString.indexOf("llo",0));
```

2. lastIndexOf: tương tự như indexOf nhưng chúng ta tìm ký tự cuối cùng đi ngược lên tớiký tự đầu tiên.

ActionScript
```
var myString = "Hello World.";
//output: 2
trace(myString.indexOf("l",0));
//output: 9
trace(myString.lastIndexOf("l"));
```

3. toUpperCase/toLowerCase: dùng để đổi chuỗi từ viết thường sang viết hoa và ngược lại

ActionScript
```
var myString = "Hello World.";
//output: HELLO WORLD
trace(myString.toUpperCase());
//output: hello world
trace(myString.toLowerCase());
```

4. length: là đặc tính của Sring dùng để tính chiều dài của chuỗi.

ActionScript
```
var myString = "Hello World.";
//output: 12
trace(myString.length);
```

Bài tập:

1. mở file 09form-noscripts.fla
2. kiểm tra tên của variable của mỗi input text field qua property panel
3. Thêm đoạn mã sau vào nút **CLEAR**

ActionScript
```
on (release) {
  clearForm();
}
```

4. thêm đoạn mã sau vào nút **SUBMIT**

ActionScript
```
on (release) {
  submitForm();
}
```

5. Thêm đoạn code sau vào frame đầu tiên của movie

ActionScript
```
function clearForm() {
  firstName = "";
  middleInitial = "";
  lastName = "";
  address = "";
  city = "";
  state = "";
  zip = "";
  phone = "";
  comments = "";
```

```
}

function submitForm() {
  if (middleInitial.length == 1) {
     trace("Name: "+firstName+" "+middleInitial+". "+lastName);
  }  else {
     trace("Name: "+firstName+" "+lastName);
  }
}
```

Rồi bạn test movie.

Giờ thứ 10: Tạo Và Điều Khiển Văn Bản, Creating and Controlling Text

Dynamic Text

Bạn cũng có thể hiển thị văn bản bằng dynamic text field. Khác với Input field mà bạn làm quen trong giờ số 9, dynamic text field chỉ cho phép AS thay đổi nội dung văn bản chứ không phải là người dùng. Vì nội dung, hình thức của văn bản có thể thay đổi bất cứ lúc nào nên mới gọi là dynamic text

Trước tiên bạn tạo dynamic text bằng dụng cụ text trên thanh toolbar, nhưng thay vì chọn "Input Text" ở trong property panel, bạn chọn "Dynamic Text"

Bước kế tiếp là nối nội dung của văn bản trong dynamic text field với một variable bằng cách viết tên của variable đó vào trong khung **Var** ở trong propery panel. Ví dụ như bạn đặt variable myText vào trong khung Var thì khi bạn thay đổi giá trị của myText thì văn bản trong dynamic text field cũng thay đổi giống như giá trị của biến myText. Ngoài ra bạn còn có thể thay đổi một số lựa chọn cho dynamic text field như "Single", "Multiline", or "Multiline No Wrap" để xử lý trường hợp nội dung của văn bản quá dài và cònn nhiều lựa chọn khác nữa. (Nếu bạn chưa nắm vững phần này thì có thể tham khảo thêm về sử dụng Flash trong CD được post box tài liệu)

Định dạng văn bản theo HTML
(HTML format)

Để định dạng văn bản kiểu HTML, bản phải cho phép dynamic text field chấp nhận định dạng HTML (click vào HTML icon ở trên property panel) và sau đó bạn có thể dùng thẻ HTML để định dạng văn bản của bạn. Ví dụ đoạn code này sẽ tạo văn bản sau:

CODE

```
myText = "This text is <B>bold</B>.<BR>";
myText += "This text is <I>italic</I>.<BR>";
myText += "This text is <U>underlined</U>.<BR>";
myText += "This text is <FONT COLOR='#FF0000'>red</FONT>.<BR>";
myText += "This text is <FONT FACE='Arial Black'>Arial Black</FONT>.<BR>";
```

```
myText += "This text is <FONT SIZE='24'>large</FONT>.<BR>";
myText += "This text is <A HREF='link.html'>linked</A>.<BR>";
```

QUOTE
This text is <B>bold</B>.
This text is <I>italic</I>.
This text is <U>underlined</U>.
This text is <FONT COLOR='#FF0000'>red</FONT>.
This text is <FONT FACE='Arial Black'>Arial Black</FONT>.
This text is <FONT SIZE='24'>large</FONT>.
This text is linked.

Siêu liên kết (hyper link) trên có tác dụng giống như trong HTML như khác 1 điều là không có gạch dưới
Sau đây là những thẻ HTML có thể dùng trong Flash MX

- <B></B>: viết đậm
- <I></I>: viết nghiêng
- <U></U>: gạch dưới
- <FONT FACE='Arial Black'></FONT>: kiểu chữ
- <FONT SIZE='24'></FONT>: cỡ chữ
- <FONT COLOR='#XXXXXX></FONT>: màu chữ
- <A HREF=''></A>: link
- <P></P>: Đọan văn
-
: Xuống hàng

Đối Tượng TextFormat

Còn một cách khác để định dạng cho văn bản là dùng đối tượng TextFormat. Để làm được điều này, bạn cần phải tạo ra 1 biến từ đối tượng TextFormat. Sau đó bạn có thể định giá trị cho các thuộc tính của nó. Ví dụ bạn muốn kiểu chữ Arial Black, cỡ 26, và màu đỏ thì dùng đọan code sau:
CODE
```
myFormat  = new TextFormat();
myFormat.font = "Arial Black";
myFormat.size = 36;
myFormat.color = 0xFF0000;

textInstance.setTextFormat(myFormat);
```

Bạn có thể dùng đối tượng TextFormat như là stylesheet trong các file HTML, một khi bạn tạo nó ra thì bạn có thể dùng nó ở bất cứ chổ nào trong movie
Bạn có thể coi các đọan code trên trong file mẫu: 10formattext.fla

Variable ở ngoài Flash

Bạn có thể lấy biến từ ngoài trang HTML vào trong Flash để thay đổi nội dung của Flash hay dùng vào bất cứ chuyện gì. Ví dụ bạn phải làm 30 cái banner bằng Flash cho 30 trang web, thì bạn có thể chỉ làm 1 cái banner thôi, nhưng thay đỗi nội dung

tuỳ theo từng trang web. Muốn nhập văn bản hay giá trị của biến nào từ HTML, bạn chỉ cần thêm vài chữ vào sau cái tên của movie trong phần <object> và <embed> trong HTML.. Đoạn mã HTML sau đây nhập giá trị của biến txtName vào trong Flash

HTML
<PARAM NAME=movie VALUE="10banner.swf?txtName=Dominico Savio!">
và trong thẻ <embed> thì bạn thêm phần variable và giá trị vào sau tên file
HTML
src="10banner.swf?txtName=Dominico Savio!">
Khi movie load thì nó sẽ tự tạo variable "txtname" và đặt giá trị cho biến này bằng "Dominico Savio". Bạn có thể xem trang 10banner.html, 10banner.fla trong cái source file để xem chi tiết hơn.
Bạn có thể làm nhiều giá trị một lúc cũng bằng cách này
10banner.swf?txtName=Dominico Savio!&gender=male"
Ngoài lấy biến từ HTML, bạn còn có thể lấy biến từ các text file. Cách này thích hợp khi bạn có nhiều biến để nhập vào Flash, hay giá trị của các biến quá dài để gắn vào thẻ HTML. Bạn chỉ cần dùng lệnh **loadVariables()** để làm được điều này. Ví dụ:
CODE
loadVariables("10external.txt", _root);
Bạn có thể coi thêm về cách dùng loadVariables() ở bên box ActionScript

[**Bài Tập**
Một ví dụ đơn giản áp dụng các điều mà bạn vừa học trong bài này là làm một cái "news ticker" như dưới đây:

Dòng chữ sẽ chạy từ bên phải qua bên trái cho suốt bản tin. Bạn có thể tham khảo source fla: 10ticker.fla và test nó để có khái niệm thêm về "news ticker"
1. Tạo movie mới

2. Tạo dynamic text field, với độ cao chỉ vừa 1 hàng, và dài bằng chiều ngang của stage, để cho dễ nhận, bạn có thể viết vài chư trong đó, như là "text goes here"

3. Tạo variable cho text field này là **text**

4. Chọn font, nên dùng kiểu chữ tương tự như Courier New, sao cho bề ngang của mỗi chư bằng nhau

5. Con text field rồi nhấn F8 để đổi thành Movie Clip, và đặt cho nó 1 cái tên như thế nào tuỳ bạn

6. Sau đó thì gắn đoạn script sau vào movie
CODE
```
onClipEvent(load) {
  // đặt bản tin
  tickerText = "News Alert: ";
  tickerText += "Stock prices shoot up sharply with good earnings reports. ";
  tickerText += "The first manned flight to Mars prepares to leave Earth orbit. ";
  tickerText += "Your favorite sports team wins championship. ";
  tickerText += "Scientists find cure for major diseases. ";
```

```
  firstChar = 0; // bắt đầu ở chữ cái đầu tiên
  lineLength = 50; // số chữ cái tối đa trên news ticker
  // thêm các khoảng trống vào trước dòng tin
  for(var i=0;i<lineLength;i++) {
     tickerText = " " + tickerText;
  }
}
onClipEvent(enterFrame) {
  // phát bản tin
  text = tickerText.substr(firstChar,lineLength);
  // thêm chữ cái kê tiếp
  firstChar++;
  // phát lại bản tin sau mỗi lần kết thúc
  if (firstChar > tickerText.length) {
     firstChar = 0;
  }
}
```

Trước tiên bạn tạo ra biến **tickerText** để chứa bản văn news ticker, biến **firstChar** để chứa giá trị của chữ cái đầu tiên sẽ thấy trong news ticker, và **lineLenght** là số chữ cái có thể xuất hiện một lúc trên news ticker. Và tất cả các đều được cho vào trong bộ sử lý dữ kiện **onLoad**
Trong **onEnterFrame**, Flash sẽ lấy 50 chữ cái đầu tiên bỏ vào biến **text** (ở trong dynamic text field) để cho nó hiển thị lên màn hình. Sau đó di chuyển sang chữ cái thứ 2 bằng câu **firstChar++;**, và sau cùng là kiểm tra, nếu bản tin đã được phát hết thì phát lại từ đầu.

6. Bạn có thể thay đổi giá trị của lineLenght và giá trị của bản văn trong tickerText để hiểu hơn về 2 biến này

7. Bây giờ bạn hãy thữ dùng loadVariable() để nhập bản tin từ text file vào coi có được không. Nếu không thì hãy trở lại đây hỏi mình nhe 

Happy flashing

Giờ thứ 11: Làm việc với các con số, Chương này rất hay và bổ ích!

Chúng ta đã nghiên cứu về text và strings trong các chương trước. Bây giờ chúng ta sẽ nghiên cứu về những con số. Bạn sẽ phải đụng độ với những con số trong Action Script trong nhiều trường hợp, vì vậy bạn cần phải hiểu làm sao để sử dụng chúng.

Những phép toán (operator) và những hàm (function) về số

Những phép toán đơn giản

Chúng ta cũng đã học qua các phép toán đơn giản rồi. Bây giờ hãy xem lại nhé. Dấu + và dấu - sẽ thực hiện phép toán cộng và trừ, dấu * thực hiện phép nhân, dấu / thực hiện phép chia.
Ngoài ra, bạn cũng có thể sử dụng những phép toán như +=, -=, *=, /= để làm đơn giản cho code của bạn. Lấy ví dụ hai câu lệnh dưới đây sẽ thực hiện cùng một công

việc là cộng thêm 7 và biến a

CODE
```
a = a + 7;
a += 7;
```

Những phép so sánh

Những phép toán so sánh như == để so sánh hai số xem có bằng nhau không, > (lớn hơn), < (nhỏ hơn), <= (nhỏ hơn hoặc bằng), >= (lớn hơn hoặc bằng). Trong Action Script chúng ta có thể so sánh một số nguyên với một số thực. Ví dụ như 7.2 sẽ lớn hơn 7.

Math.abs

Đối tượng Math chứa một tập hợp các hàm xử lý các con số. Hàm Math.abs sẽ trả về giá trị tuyệt đối của một số. Bạn hãy thử đoạn code này xem kết quả thế nào nhé:

CODE
```
trace(Math.abs(-7));
```

Math.round

Nếu bạn có một số thực, nhưng bạn lại muốn hiển thị cho người dùng xem dưới dạng số nguyên thay vì phải cho họ xem các con số phần thập phân, bạn hãy sử dụng hàm Math.round. Thử nhé:

CODE
```
trace(Math.round(7.2));
```

Math.ceil, Math.floor

Hai hàm khác dùng để làm tròn số thực thành số nguyên là Math.ceil và Math.floor. Hàm Math.ceil sẽ làm tròn số lên và Math.floor sẽ làm tròn số xuống. Thử nhé:

CODE
```
trace(Math.ceil(7.2));
trace(Math.floor(8.3));
```

Math.min, Math.max

Các bạn có thể sử dụng hai hàm này để tìm số lớn hơn và số nhỏ hơn trong hai số. Ví dụ hàm Math.min(4, 5); sẽ trả về 4, Min.math(4, 5); sẽ trả về 5

Math.pow

Đây là hàm để lấy lũy thừa, cũng đơn giản thôi, tham số thứ nhất là cơ số, tham số thứ hai là lũy thừa. Ví dụ bạn muốn lấy 4 luỹ thừa 3 thì viết thế này

CODE
Math.pow(4, 3);

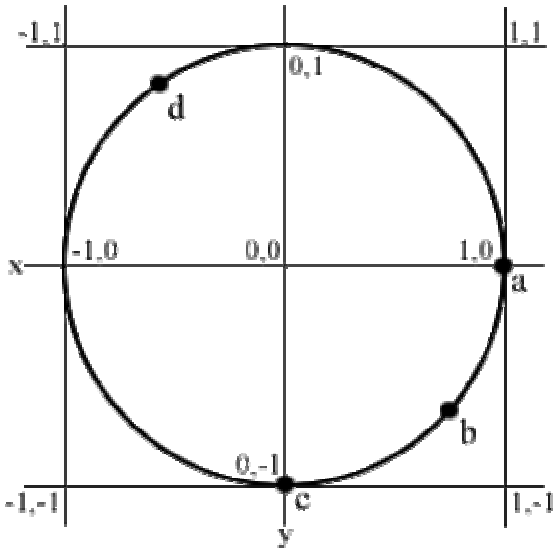
Math.sqrtĐây là hàm để tính căn, nhưng mà mình xin nói trước luôn là ngoài hàm Math.sqrt, bạn còn có thể sử dụng hàm Math.pow với lũy thừa nhỏ hơn 1 để tính căn. Ví dụ muốn lấy căn 2 của 4 thì có thể sử dụng

CODE
Math.pow(4, .5);
hoặc
Math.sqrt(4);

Cả hai cách trên đều ra cùng một kết quả là 2.

Hàm số lượng giác

Những hàm của object class mà các bạn được biết ở trên có vẻ như rất dễ học. Nhưng còn những hàm về lượng giác như sin, cosin thì "khó nuốt" hơn nhiều. Mặc dù bạn có thể bỏ qua những hàm này nhưng những hàm lượng giác này rất hữu ích và hay trong việc tạo những ứng dụng trong Flash. Vậy chúng ta hãy cùng tìm hiểu cách nó làm việc nhé!
Hàm sin và cosin sử dụng qua Math.sin và Math.cosin, biểu diễn mối quan hệ giữa đường thẳng với đường cong của một đường tròn.
Hình dưới đây biểu diễn một vòng tròn với một số điểm được đánh dấu bằng các chữ. Hãy tưởng tượng tâm của vòng tròn ở toạ độ 0,0; bán kính của đường tròn bằng 1. Vì thế, điểm trên cùng sẽ có toạ độ 0,1 và điểm tận cùng bên phải sẽ có toạ độ 1,0



Bây giờ bạn hãy tưởng tượng vòng tròn trên là một đường thẳng. Bắt đầu từ điểm "a"

có toạ độ 1,0 đến điểm 0,-1 qua -1,0 rồi qua 0,1 và cuối cùng trở về 1,0.
Trong Flash, và cũng như trong các ngôn ngữ lập trình khác, chúng ta định vị một vật bất kỳ bằng toạ độ x, y của chúng. Vì thế, việc chuyển một điểm trên đường thẳng thành một điểm có toạ độ x, y trên đường tròn rất hữu dụng. Đó là những gì mà hàm sin và cosin thực hiện. Ví dụ như chúng ta đặt điểm "a" là điểm bắt đầu của đường thẳng của đường tròn, bạn có sử dụng hàm sin và cosin của 0 để tính toạ độ x, y của điểm "a" trên đường tròn. Tất nhiên, Math.cos(0) sẽ trả về giá trị 1, Math.sin(0) sẽ trả về giá trị 0. Kết quả sẽ trả về toạ độ của điểm "a" 1,0
Vậy điểm "c" sẽ nằm ở đâu trên đường tròn? Nếu duỗi đường tròn ra sẽ có độ dài bằng 6.28 lần bán kính, do bán kính bằng 1 nên độ dài sẽ bằng 6.28. Đây là xuất xứ của hằng số **pi**. Pi bằng 3.14, bằng một nửa độ dài của đường tròn.
Độ dài của đường tròn là 6.28 thì một phần tư độ dài đường tròn là 1.57. Điểm đó tương ứng với điểm "c". Hàm Math.cos(1.57) sẽ trả về một giá trị rất nhỏ, gần bằng 0. Hàm Math.sin(1.57) sẽ trả về giá trị gần bằng -1. Chỉ gần bằng chứ không bằng vì hằng số pi không chính xác bằng 3.14, chỉ là gần bằng thôi.
Bạn có thể sử dụng cách này để chuyển các điểm trên đường tròn thành điểm có toạ độ x,y
Vậy thì nó có ích như thế nào? Giả sử như bạn muốn tạo một movieclip bay quanh màn hình trong một vòng tròn. Làm sao để làm được điều đó? Nếu làm bằng frame có thể phải tốn hàng trăm frame  . Hoặc cách khác là có thể dùng hàm Math.cos và Math.sin để tính chuyển sang toạ độ x,y quanh vòng tròn.
Trong đoạn code dưới đây, khi cộng thêm n, movie clip sẽ chuyển động dọc theo đường tròn. Hàm Math.cos sẽ tính ra giá trị x và hàm Math.sin sẽ tính ra giá trị y, chúng sẽ được nhân thêm với giá trị của radius để tăng kích thước của đường tròn. Giá trị của centerX và centerY sẽ được cộng thêm cho các toạ độ x, y, vì thế tâm đường tròn bây giờ không còn ở 0,0 nữa mà sẽ ở vị trí 150,150

CODE
```
onClipEvent(load) {
  n = 0;
  radius = 100;
  centerX = 150;
  centerY = 150;
}

onClipEvent(enterFrame) {
  n += .1;
  this._x = Math.cos(n) * radius + centerX;
  this._y = Math.sin(n) * radius + centerY;
}
```

Chữ (string) và số (number)

Có rất nhiều cách để chuyển đổi từ chuỗi sang số và ngược lại. Ví dụ như trường hợp người dùng nhập một số vào text field, bạn sẽ nhận giá trị đó và cộng thêm một. Hãy xem đoạn code sau đây, num là một biến giá trị text field

CODE

b = num + 1;

Nếu num = 42, vậy thì b sẽ là 43, đúng không? Xin trả lời là sai, kết quả b là 421, đơn giản bởi vì num là một chuỗi (string) chứ không phải là một số, vì thế khi chúng ta thực hiện phép cộng num với 1 có nghĩa là cộng ký tự "1" vào chuỗi num đang có giá trị "42".
Để làm cho Flash hiểu num là một số, bạn có thể sử dụng một trong hai hàm sau đây để chuyển một chuỗi thành số. Hàm parseInt chuyển một chuỗi thành kiểu số nguyên, hàm parseFloat chuyển một chuỗi thành kiểu số thực. Ví dụ hàm parseInt("42") sẽ trả về giá trị là 42, nếu chúng ta sử dụng parseInt("42.9") thì cũng sẽ nhận được giá trị là 42 bởi vì hàm parseInt không làm tròn số, nó chỉ lấy phần nguyên mà thôi. Còn nếu sử dụng hàm parseFloat("42.9") bạn sẽ nhận được giá trị là 42.9, muốn làm tròn các bạn có thể sử dụng hàm Math.round(parseFloat("42.9")). Hàm parseFloat cũng sẽ trả về một một số nguyên nếu tham số truyền vào là một chuỗi số nguyên. Ví dụ parseFloat("42") sẽ cho ta số 42. Trừ trường hợp bạn muốn trả về một giá trị số nguyên, còn không bạn nên sử dụng hàm parseFloat.
Một nét rất đặc biệt và rất hay của hàm parseInt là có thể chuyển một chuỗi thành một số nhưng thành nhiều dạng hệ số khác nhau. Hãy xem một ví dụ cho dễ hiểu:

CODE
parseInt("FF", 16);

Hàm này sẽ trả về giá trị là 255, ý nghĩa của nó là chuyển chuỗi chứa số FF là một số hexa hệ số 16 thành một số hệ nguyên hệ số 10.
Ngược lại, để chuyển từ một số sang một chuỗi, hãy sử dụng hàm toString. Hàm này sẽ hoạt động khác với hàm parse, bởi vì nó hoạt động bên trong mỗi biến, gọi hàm này sau dấu chấm (.). Xem ví dụ nhé:

CODE
a = 135;
trace(a.toString() + 1);

Kết quả sẽ là 1351.
Chúng ta cũng có thể sử dụng hàm toString để chuyển một số thành một chuỗi chứa số dưới một hệ số khác. Ví dụ a.toString(16) sẽ trả về kết quả là chuỗi "ff"

Số ngẫu nhiên

Tạo số ngẫu nhiên là một phần quan trọng trong công việc thiết kế game và một số hoạt hình. Bởi vì nếu không có sô ngẫu nhiên thì những đoạn phim của bạn sẽ chạy giống nhau, không còn gì thú vị nữa!
Để tạo số ngẫu nhiên, bạn có thể sử dụng hàm Math.random(). Hàm này sẽ trả về một giá trị từ 0.0 đến 1.0 nhưng thường thì không trả về giá trị bằng 1.0
Hãy xem ví dụ dưới đây, bạn sẽ nhận được con số ngẫu nhiên trong cửa sổ OutPut

CODE

```
trace(Math.random());
```

Kết quả trả về đại loại giống như 0.023268056102097, nhưng mỗi lần lại khác nhau. Một ví dụ khác là bạn cần tạo một số ngẫu nhiên từ 1 đến 10, việc này rất đơn giản, chỉ cần nhân thêm số ngẫu nhiên cho 10. Đoạn code sau sẽ cho ta con số ngẫu nhiên từ 0.0 đến 10.0

CODE
```
trace(Math.random() * 10);
```

Nhưng cái chúng ta cần là con số ngẫu nhiên từ 1.0 đến 10.0 chứ không phải từ 0.0 đến 10.0, vì thế, hãy cộng thêm 1

CODE
```
trace(Math.random() * 10 + 1);
```

Bây giờ thì kết quả trả về là 1.0 đến 11.0 nhưng sẽ không có kết quả 11.0. Hãy sử dụng hàm Math.floor để làm tròn xuống.

CODE
```
trace(Math.floor(Math.random() * 10 + 1));
```

Thật ra thì con số ngẫu nhiên trong máy tính cũng chưa thật sự ngẫu nhiên. Bởi vì nó không thật sự thay đổi trong bộ vi xử lý. Thay vào đó sẽ có một con số chuẩn, một con số nào đó không biết trước được như là giờ hoặc phút… của hệ thống, con số này sẽ được đưa và một biểu thức rất phức tạp, rất rất phức tạp mà chúng ta không thể đoán được. Kết quả trả về cho chúng ta kết quả dường như là ngẫu nhiên. Kết quả này sẽ được đưa và một hàm tính một lần nữa là lưu lại thành con số chuẩn để tính ngẫu nhiên cho lần tiếp theo.
Hãy nghĩ về điều này, số ngẫu nhiên trong đời sống thật sự cũng không thật sự ngẫu nhiên. Nếu chúng ta giữ một mặt của con xúc xắc, thẩy đúng theo một hướng thật chính xác, chúng ta sẽ có cùng một kết quả 
Được rồi, bây giờ hãy thử tạo một số ngẫu nhiên từ 3 đến 7. Làm thế nào đây? Ah, có một thủ thuật cho bạn đây:

CODE
```
trace(Math.floor(Math.random() * 5 + 3);
```

Trong phạm vi từ 3 đến 7 sẽ có tất cả là 5 số nguyên 3, 4, 5, 6, và 7. Thế còn trong phạm vi từ 50 đến 100 thì sao

CODE
```
trace(Math.floor(Math.random() * 51 + 50);
```

Là số 51 bởi vì trong phạm vi từ 50 đến 100 có 51 số, còn nếu từ 51 đến 100 sẽ là

CODE
```
trace(Math.floor(Math.random() * 50 + 50);
```

Một cách để thử xem phạm vi của bạn có đúng như ý của mình không, đó là hãy thử với số nhỏ nhất và số lớn nhất thay cho Math.random(). Giá trị nhỏ nhất của hàm Math.random() là 0, hãy thử với số 0 nhé

CODE
```
trace(0 * 51 + 50);
```

Giá trị lớn nhất của Math.random() sẽ không chính xác bằng 1.0 mà sẽ là gần bằng, vậy chúng ta hãy thử với giá trị là 0.9999

CODE
```
trace(.9999 * 51 + 50);
```

Hãy thử xem có đúng không nhé! 
Dưới đây là một ví dụ nữa rất thú vị, movie clip của bạn sẽ nhảy đến những vị trí bất kỳ

CODE
```
onClipEvent(enterFrame) {
  this._x = Math.random()*550;
  this._y = Math.random()*400;
}
```

Luyện tập: Tạo một chương trình máy tính đơn giản

Hãy bắt tay vào việc thiết kế một chương trình máy tính đơn giản nhé.
- Mở một movie mới trong Flash. Movie của chúng ta sẽ giống như hình dưới đây

Chúng ta sẽ có 10 nút để nhập 10 số từ 0 đến 9, các nút các phép toán, phím =, dấu chấm thập phân, phím C để xóa màn hình. Một text field đặt ở trên để hiển thị những số người dùng bấm, text field này sẽ liên kết với biến **display**.
- Trong mỗi movie clip của mỗi nút bấm, chèn đoạn code sau

CODE
```
on (release) {
  _parent.keyPressed(this._name);
}
```

Khi một nút bất kỳ được nhấn thì hàm keyPressed sẽ được gọi ở level ngoài mà ở đây sẽ là level root. Hàm này sẽ truyền tên của movie clip.
- Quay trở lại level root, mỗi movie clip phải có một tên riêng, không được trùng. Đặt tên của movie clip theo số của nó, ví dụ nút số 5 sẽ có tên là 5, dấu chấm thập phân có tên là ".". Dấu cộng, dấu trừ, dấu nhân, dấu chia, dấu bằng, nút C lần lượt đặt tên là plus, minus, multiply, divide, equals, clear.
- Ok, bây giờ bắt đầu viết code nhé! Code này sẽ nằm ở ngoài movie frame

CODE
```
// trước tiên, xoá màn hình cũ
clearAll();

function clearAll() {
  display = "0"; // giá trị mặc định là 0
  memory = 0; // bộ nhớ
  operation = "none"; // chưa có phép toán nào cả
  newNum = true; // đánh dấu khi nào đã nhập xong một số
}
```

Biến display là một chuỗi (string), sẽ liên kết với text field hiển thị nội dung người dùng bấm. Nó sẽ bắt đầu với giá trị bằng 0, có nghĩa là khi người dùng mới bật máy lên thì sẽ mặc định là số 0
Biến memory lưu lại con số trước đó để thực hiện phép tính. Điều này rất cần thiết, ví dụ bạn bấm số 5, +, 7 thì số 5 nhập vào, khi bạn bấm dấu cộng, màn hình sẽ được xóa và số 5 được lưu vào memory, số 7 được nhập tiếp và sẽ thực hiện phép tính 5 và 7. Phép toán người dùng chọn cũng cần được lưu vào bộ nhớ. Khi người dùng đã nhấn 5, +, 7 rồi, sau đó bấm dấu = hoặc một phép toán khác thì chương trình sẽ gọi lại phép toán cũ đã được lưu trong biến operation để thực hiện phép toán trước của 5 với 7. Biến newNum như là một biến cờ hiệu để báo cho chương trình biết khi nào sẽ kết thúc một phép toán. Ví dụ khi người dùng bấm 5, +, 7 rồi sau đó bấm một phép toán khác thì kết quả sẽ được lưu lại để tính toán tiếp, lúc này phép toán chưa kết thúc, newNum = false. Còn nếu người dùng bấm dấu = thì màn hình sẽ hiện ra kết quả và kết thúc luôn phép toán để chuyển sang phép toán mới, newNum = true.
- Phần tiếp theo là hàm keyPressed để xử lý khi mỗi nút được nhấn. Hàm keyPressed sẽ sử dụng cấu trúc switch thay cho cấu trúc if, then, else. Cả hai cách đều hoạt động tương tự nhưng cách viết hơi khác, bạn hãy xem nhé:

CODE
```
// hàm này được gọi bởi nút nhấn
function keyPressed(keyName) {

  // do something different for different keys
  switch (keyName) {
  case "clear" : // khi nhấn nút C
     clearAll();
     break;
  case "plus" : // các phép toán cộng, trừ, nhân, chia
     operate(keyName);
```

```
      break;
  case "minus" :
      operate(keyName);
      break;
  case "multiply" :
      operate(keyName);
      break;
  case "divide" :
      operate(keyName);
      break;
  case "equals" :
      operate(keyName);
      break;

  default : // các số
      if (newNum) { // hiển thị số mới trên màn hình
          display = keyName;
          newNum = false;
          if (display == "0") newNum = true; // số không bắt đầu với số 0
      } else {
          display += keyName; // nối thêm số vào màn hình
      }
      break;
  }
}
```

Khi người dùng nhấn nút C thì hàm clearAll() sẽ được gọi, tương tự khi người dùng nhấn các phép toán thì hàm operate sẽ được gọi

- Hàm operate sẽ thực hiện tính toán. Nó sẽ tìm ra phép toán ở giữa con số trướcđó và con số hiện thời, sử dụng hàm parseFloat để chuyển chuỗi thành số.Operation có giá trị bằng none khi nhập một số mới sau khi thực hiện xong phép toán và màn hình được xóa.

CODE

```
// thực hiện phép toán trước
function operate(keyName) {
  switch (operation) {
  case "none" : // số đầu tiên
      memory = parseFloat(display); // lưu lại số trước
      break;

  case "plus" : // thực hiện phép toán
      memory += parseFloat(display);
      break;
  case "minus" :
      memory -= parseFloat(display);
      break;
 case "multiply" :
      memory *= parseFloat(display);
```

```
      break;
   case "divide" :
      memory /= parseFloat(display);
      break;
   }

   // equals operation is like a clear, but results are displayed
   if (keyName == "equals") {
      operation = "none";
   }  else {
      operation = keyName; // remember this operation for next time
   }

   display = memory.toString(); // display result
   newNum = true; // prepare for next number
}
```

Luyện tập: Làm những hành tinh bay quanh quỹ đạo

Trong ví dụ về lượng giác trong phần trước, bạn đã biết cách làm một movie clip quanh quay một vòng tròn. Bây giờ hãy áp dụng những kiến thức đó để làm một movie 4 hành tinh xoay quang mặt trời, hành tinh thứ 3 sẽ có một vệ tinh.
- Tạo một movie mới trong Flash
Tạo 6 movie clip là các vòng tròn, đặt tên lần lượt là sun, mercury, venus, earth, mars, and moon, và đặt instance name của chúng giống vậy luôn. Hãy làm cho kích thước của chúng ta khác nhau một chút nhé!
- Đặt movie clip sun (mặt trời) vào giữa màn hình. Các hành tinh khác sẽ quay quanh mặt trời. Vị trí của các movie clip các hành tinh khác không quan trọng, vì chúng ta sẽ điều khiển vị trí của chúng bằng Action Script
- Đặt đoạn code sau vào movie clip mercurcy:

CODE
```
onClipEvent(load) {
   speed = .4;
   radius = 40;
   orbit = 0;
}

onClipEvent(enterFrame) {
   orbit += speed;
   this._x = Math.cos(orbit) * radius + _root.sun._x;
   this._y = Math.sin(orbit) * radius + _root.sun._y;
}
```

Trong event load sẽ thiết lập tốc độ quay của hành tinh, khoảng cách giữa hành tinh với mặt trời. Biến speed điều khiển tốc độ quay của hành tinh, có nghĩa là số vòng quay trong một frame. Nếu speed = 6.28 thì hành tinh sẽ quay được một vòng trong

đúng một frame, và nếu speed = .4 như trong đoạn code sau thì hành tinh sẽ quay một vòng quỹ đạo mất 15.7 frame (6.28/.4).
Trong mỗi lần event enterFrame xảy ra thì orbit (quỹ đạo) sẽ được cộng thêm với speed hiện thời, và hành tinh di chuyển đến vị trí mới. Toạ độ x, y của các hành tinh sẽ được hiệu chỉnh theo vị trí của mặt trời, vì vậy tâm quỹ đạo của các hành tinh chính là mặt trời.
- Chạy thử movie. Lúc này, các hành tinh vẫn đứng yên, chỉ có mercurcy là di chuyển. Nếu quỹ đạo bị nghiêng thì bạn phải xem lại xem mặt trời có nằm đúng ở giữa quỹ đạo hay không.
- Đặt đoạn code tương tự vào các movie clip venus, earth, and mars movie clips, nhưng mà bạn hãy thay đổi giá trị của hai biến speed và radius để làm cho hành tinh xa hoặc gần mặt trời hơn. Gợi ý cho các bạn nhé, speed của các hành tinh mercury, venus, earth, and mars là .4, .2, .1, và .05, radius cho các hành tinh là 40, 90, 150, và 210.
- Chạy thử lại movie. Bây giờ thì cả bốn hành tinh đã chuyển động, chỉ còn moon là chưa chuyển động vì chưa được viết code điều khiển.
- Bây giờ hãy viết code cho moon, moon sẽ có speed nhanh hơn và radius nhỏ hơn, và moon sẽ quay quanh earth thay vì quay quanh sun

CODE
```
onClipEvent(load) {
  speed = .5;
  radius = 15;
  orbit = 0;
}

onClipEvent(enterFrame) {
  orbit += speed;
  this._x = Math.cos(orbit) * radius + _root.earth._x;
  this._y = Math.sin(orbit) * radius + _root.earth._y;
}
```

- Bây giờ hãy chạy thử movie của bạn một lần nữa. Bây giờ tất cả các hành tinh đều đã quay rồi, nhưng còn một vấn đề. Earth không chính xác nằm giữa quỹ đạo của moon, vì sao thế? Bởi vì moon sẽ quay trước earth, vì moon nằm ở một layer trên layer của earth. Chúng ta cần phải làm cho earth quay trước rồi mới đến moon. Để làm như vậy, chọn movie clip moon, rồi chọn Modify -> Arrange -> Send to Back.

Luyện tập: Làm tuyết rơi

Đây là một ví dụ rất hữu ích cho việc tạo số ngẫu nhiên. Bạn có thể không cần phải dùng Action Script nhưng lúc đó bạn phải làm hàng trăm movie clip bông tuyết, mỗi cái chuyển động theo một đường.
Bằng cách sử dụng Action Script và số ngẫu nhiên, bạn có thể làm cho các bông tuyết rơi với tốc độ và hướng rơi ngẫu nhiên. Bắt tay vào làm nhé!
- Tạo một movie mới trong Flash
- Tạo một movie clip mới, đặt tên là Snowflake và đặt tên instance của nó là snowflake
- Đặt đoạn code sau vào movie clip Snowflake. Đoạn code này sẽ khởi tạo những giá

trị ban đầu như vị trí của các bông tuyết, tốc độ rơi, tốc độ bị thổi ngang, độ xoay của bông tuyết.

CODE

```
onClipEvent(load) {
  this._x = Math.random()*550; // 0 to 550
  this._y = Math.random()*400; // 0 to 400
  speed = Math.random()*3+3; // 3 to 6
  drift = Math.random()*2-1; // -1 to 1
  rotate = Math.random()*6-3; // -3 to 3
}

onClipEvent(enterFrame) {
  this._y += speed;
  this._x += drift;
  this._rotation += rotate;

  // đưa những bông tuyết lên đầu
  if (this._y > 400) this._y = 0;

  // kiểm tra hai biên
  if (this._x < 0) this._x = 550;
  if (this._x > 550) this._x = 0;
}
```

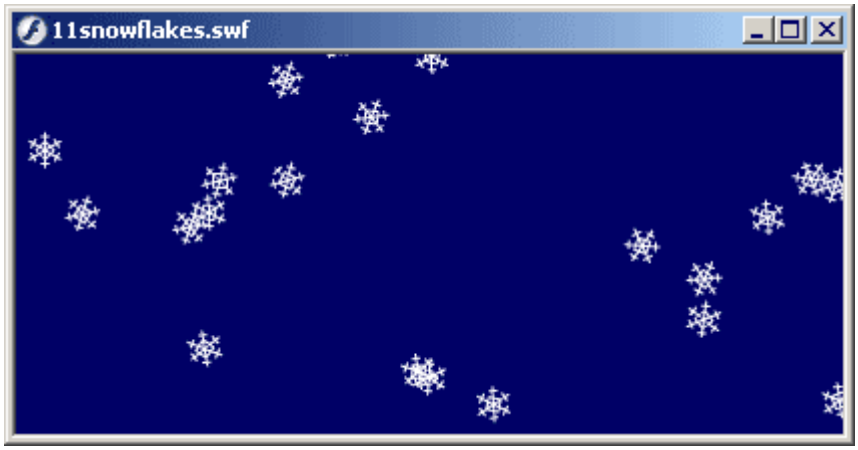
Khi event enterFrame xảy ra thì bông tuyết sẽ được điều khiển rơi xuống bởi speed và bị dạt theo hướng ngang bởi drift. Bông tuyết cũng sẽ bị xoay theo giá trị của rotate. Tiếp theo là sẽ kiểm tra xem nếu bông tuyết đã rơi xuống đất rồi thì sẽ cho nó rơi lại, nếu bông tuyết bị dạt qua bên phải thì nó sẽ được đưa quay lại về bên trái…

- Chạy thử movie của bạn. Các bông tuyết sẽ rơi tự do. Hãy click vào cuối movie xem sao, những bông tuyết ở dưới sẽ được rơi lại.
- Được rồi, bây giờ hãy làm cho bông tuyết nhiều nhiều một tí bằng đoạn code sau, đặt vào timeline chính nhé!

CODE

```
// tạo 50 bông tuyết
for(var i=0;i<50;i++) {
  snowflake.duplicateMovieClip("snowflake"+i,i);
}
```

Hàm duplicateMovieClip hoạt động cũng giống như hàm attachMovie để đưa movie clip vào trong lúc đang chạy. Điểm khác biệt là hàm duplicateMovieClip sẽ chèn từ một Movie clip có sẵn, sử dụng lại tất cả hình ảnh và code, nhưng bạn phải chắc chắn rằng mỗi movie clip phải có một tên riêng và một level riêng



Giờ thứ 12: Đối tượng và Mảng, Objects and Arrays

Cho đến bây giờ, mỗi biến đều chứa dữ liệu riêng lẻ, mỗi biến chứa một dữ liệu. Đối với những chương trình nhỏ và đơn giản thì không có gì để nói, nhưng đến khi bạn cần sử dụng thật nhiều biến. Cũng có những lúc bạn gặp khó khăn khi nhận ra rằng bạn cần lưu rất rất nhiều dữ liệu, những biến bình thường muốn làm thì không phải dễ.
ActionScript cung cấp cho ta hai thứ để lưu những loại nhiều dữ liệu. Một là Custom Object (Đối tượng), với Custom Object bạn có thể nhóm những dữ liệu riêng lẻ lại với nhau. Cách khác là Array (Mảng), một trong những phần cơ bản của các ngôn ngữ lập trình cao cấp.
Trong giờ thứ 12 này, chúng ta sẽ học:
- Cách tạo Custom Object
- Cách sử dụng những đối tượng ActionScript được dựng sẵn
- Tìm hiểu về Array (Mảng)
- Làm chữ chuyển động
- Tạo những vật thể chạy theo con trỏ chuột trên màn hình

Tạo Custom Object

Chắc hẳn các bạn đã quen thuộc với những tên như x và y để điều khiển vị trí trên movie clip. Có bao giờ bạn cảm thấy nhàm chán với những chữ x, y hay muốn thay thế nó bằng những chữ khác như positionX, positionY hay bất cứ gì bạn thích không? Việc đó quá đơn giản, chỉ cần tạo biến của bạn rồi gán giá trị cho biến của bạn thôi . Nhưng dù sao thì đó cũng chỉ là những biến riêng lẻ thôi, chỉ với một cái tên khác .
Trong phần này, chúng tôi không muốn chỉ cho các bạn làm những điều như vậy mà sẽ nói về cách tạo một custom variable Object. Ví dụ, bạn có thể lưu biến x, y như sau:

CODE
```
pos = {x:10, y:20};
```

Bằng cách sử dụng hai dấu ngoặc nhọn, bạn đã tạo ra một object. Bạn có thể truy cập dữ liệu bằng cách:

CODE
```
trace(pos.x);
```

Bạn có thể sử dụng nó như bất cứ những biến nào khác.
Hãy tưởng tượng đến một cấu trúc phức tạp hơn nhiều như một record trong một cơ sở dữ liệu. Ví dụ như ta có một custom object tên record, và trong nó sẽ có những property như là name, address, phone…
Bạn có thể tạo object từng bước một, tạo thêm property. Xem ví dụ này nhé:

CODE
```
record = new Object();
record.name = "Gary";
record.age = 32;
record.state = "Colorado";
trace(record.name);
```

Cũng với mục đích làm cho dữ liệu dễ tổ chức hơn, custom object cũng giống như những đối tượng dựng sẵn. Hai ví dụ về những đối tượng dựng sẵn là Color và đối tượng Date.

Đối tượng Color

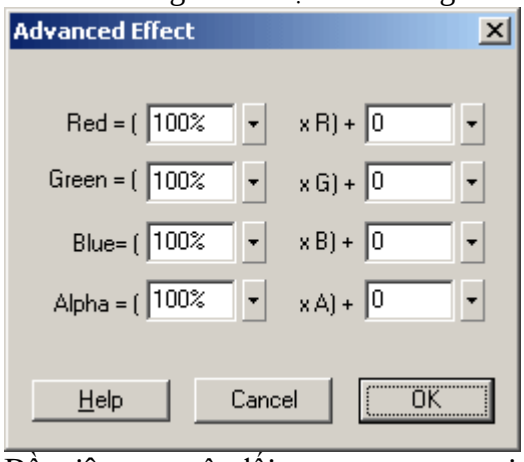
Có thể dùng ActionScript để đổi màu của một movie clip bằng cách sử dụng câu lệnh setRGB. Đó là cách dễ nhất nhưng nó sẽ không thực hiện được đối với instance của movie clip. Thay vào đó, chúng ta sẽ chuyển đến đối tượng Color của movie clip. Thực hiện cách đó bằng hàm new Color(), sau đó thì có thể sử dụng câu lệnh setRGB để thay đổi màu.
Dưới đây là một ví dụ, sử dụng setRGB từ đối tượng Color của movie clip để đặt lại màu cho movie clip sang màu 0xFF0000 (màu đỏ)

CODE
```
circleColor = new Color("circle");
circleColor.setRGB(0xFF0000);
```

Chúng ta cũng có thể lấy màu của một movie clip. Sử dụng hàm getRGB() để lấy màu của movie, nhưng nhớ sử dụng hàm toString(16) để chuyển thành giá trị hexa.

CODE
```
circleColor = new Color("circle");
trace(circleColor.getRGB(0xFF0000).toString(16));
```

Một cách khác để đặt màu cho movie clip bằng cách tạo một đối tượng color transform ứng với 8 mục như trong hình dưới đây



Đầu tiên, tạo một đối tượng custom variable. Rồi tạo các property ra, ga, ba, aa, rb, gb, bb, ab. Các ký tự đầu tiên r, g, b, a là đại diện cho từ red, green, blue và alpha. Còn ký tự thứ hai a và b là đại diện cho 2 cột bên trái và bên phải
Thay vì sử dụng setRGB thì chúng ta sẽ sử dụng setTransform như trong ví dụ dưới đây:

CODE
```
circleColor = new Color("circle");
myObject = new Object();
myObject = {ra:100, rb:255, ga:0, gb:0, ba:0, bb: 0, aa: 100, ab: 0};
circleColor.setTransform(myObject);
```

Đối tượng Date

Một đối tượng dựng sẵn khác là đối tượng Date. Đối tượng này cũng có thể được xem như là một biến bình thường, nó đại diện cho một thời điểm của thời gian.
Đối tượng Date được chia ra 7 phần: năm (year), tháng (month), ngày (date), giờ (hour), phút (minute), giây (second) và phần trăm giây (milisecond). Bạn có thể tạo một đối tượng Date mới bằng cách cung cấp cho nó 7 thông tin trên

CODE
```
myDate = new Date(2002,3,29,10,30,15,500);
```

Hãy thử sử dụng câu lệnh trace để kiểm tra nội dung của myDate xem nào. Có phải bạn nhận được kết quả này không?

CODE
Mon Apr 29 10:30:15 GMT

Rất dễ hiểu, đúng không nào? Nhưng mà còn một vấn đề: tháng là Apr trong khi chúng ta truyền tham số vào cho tháng là 3, vậy phải là March chứ!?
Đối tượng Date của Flash cũng hoạt động tương tự như đối tượng Date của các ngôn ngữ lập trình khác. Tháng được quy định từ 0 đến 11, rất đặc biệt, bởi ngày thì được quy định từ 1 đến 31  . Nhưng dù sao thì bạn cũng phải làm quen đi 
Bạn cũng có thể lấy những thành phần từ đối tượng Date. Ví dụ dưới đây sẽ lấy năm.

CODE
myDate.getYear()

Bạn có thể lấy giờ hiện tại bằng cách đơn giản là tạo một đối tượng Date rỗng  .
Giờ hiện tại sẽ được đưa vào đối tượng. Ví dụ:

CODE
myDate = new Date();
trace(myDate);

Chú ý: Flash sẽ lấy giờ của đồng hồ hệ thống trong máy bạn. Vì thế đồng hồ trong máy tính của bạn chạy sai thì giờ lấy ra sẽ sai 

Mảng (Array)

Mảng là một phần rất quan trọng trong lập trình. Bạn sẽ gặp mảng trong hầu hết các ngôn ngữ lập trình bởi vì nó là một công cụ rất cần thiết cho những ngôn ngữ lập trình phức tạp.

Mảng (Array) là một dãy dữ liệu. Trong đó thì các dữ liệu sẽ có cùng một kiểu dữ liệu với nhau, như là cùng là tên của movie clip hoặc đều cùng là vị trí của movie clip

Tạo mảng

Dưới đây là một ví dụ về một mảng. Sử dụng dấu ngoặc vuông và những dấu phẩy để

tạo mảng:

CODE
```
myArray = [36,23,63,71,25];
```

Mảng myArray có chứa 5 phần tử là số nguyên. Muốn lấy giá trị của một phần tử trong mảng thì sử dụng như sau:

CODE
```
trace(myArray[0]);
```

Phần tử đầu tiên trong mảng luôn luôn được đánh số 0. Vì vậy, như ví dụ trên có 5 phần tử thì phần tử thứ 0 chứa giá trị 36 và phần tử thứ 4 mang giá trị 25.
Một cách khác để tạo mảng là sử dụng new Array()

CODE
```
myArray = new Array();
```

Muốn thêm một phần tử vào cuối mảng, sử dụng câu lệnh push. Ví dụ dưới đây sẽ tạo ra một mảng giống mảng ở ví dụ trên:

CODE
```
myArray = new Array();
myArray.push(36);
myArray.push(23);
myArray.push(63);
myArray.push(71);
myArray.push(25);
```

Những thao tác trên mảng

Để kiểm tra xem mảng có bao nhiêu phần tử thì bạn có thể sử dụng thuộc tính length

CODE
```
myArray = [36,23,63,71,25];
trace(myArray.length);
```

Còn nếu muốn lấy phần tử cuối cùng của mảng và bỏ phần tử này ra khỏi mảng thì sử dụng câu lệnh pop

CODE
```
myArray = [36,23,63,71,25];
trace(myArray);
a = myArray.pop();
trace(a);
trace(myArray);
```

Đoạn code ví dụ trên sẽ trace 5 phần tử của mảng myArray. Sau đó, lấy phần tử cuối cùng của mảng tức là 25 đưa vào biến a, đồng thời bỏ phần tử 25 ra khỏi mảng. Cuối cùng trace các phần tử của mảng myArray, lúc này chỉ còn 4 phần tử.
Sử dụng kết hợp push và pop để tạo một hệ thống vào sau ra trước (last in first out), thường gọi là stack. Hãy tưởng tượng đến một chồng sách, chúng ta để cuốn sách đầu tiên xuống, rồi chồng lên cuốn sách thứ 2, cuốn thứ 3, cuốn thứ 4… Khi muốn lấy sách ra thì phải lấy cuốn trên cùng trước, tức là chồng lên cuối cùng.
Ngược lại với pop là shift. Nó sẽ bỏ ra phần tử đầu tiên trong mảng. Đoạn code dưới đây sẽ làm giống như đoạn trên nhưng sẽ không bỏ phần tử 25 mà sẽ bỏ phần tử 36:

CODE
```
myArray = [36,23,63,71,25];
trace(myArray);
a = myArray.shift();
trace(a);
trace(myArray);
```

Ngược lại với shift là unshift. Nó sẽ chèn thêm một phần tử vào đầu mảng.
Nếu bạn muốn lấy ra chỉ một phần của mảng thì có thể sử dụng hàm slice. Đối số truyền vào sẽ là vị trí đầu tiên và vị trí cuối cùng của phần cần lấy ra trong mảng.

CODE
```
myArray = [36,23,63,71,25]
trace(myArray.slice(1,3));
```

Đoạn code trên sẽ trả về 23, 63 vì nó không kể phần tử thứ 3. Nếu không có đối số thứ 2 thì nó sẽ lấy đến cuối mảng.
Một hàm khác nữa là hàm splice. Hàm này sẽ thay thế một số phần tử trong mảng bằng những phần tử khác. Sử dụng hàm này, đối số đầu tiên là vị trí của phần tử trong

mảng, đối số thứ hai là số phần tử muốn xóa kể từ phần tử trong đối số đầu tiên. Truyền đối số thứ 2 là số 0 nếu không muốn xóa phần tử nào cả. Những phần tử tiếp theo là danh sách những phần tử muốn chèn vào mảng. Nói vậy cũng hơi khó hiểu nhỉ, vậy hãy xét ví dụ này nhé. Ví dụ này sẽ xóa phần tử 23 và 63 và chèn vào phần tử 17.

CODE
```
myArray = [36,23,63,71,25];
myArray.splice(1,2,17);
trace(myArray);
```

Sắp xếp trong mảng

Chúng ta có thể sắp xếp một mảng bằng câu lệnh sort. Ví dụ dưới đây sẽ cho ta một mảng được sắp xếp theo thứ tự số:

CODE
```
myArray = [36,23,63,71,25];
myArray.sort();
trace(myArray);
```

Còn ví dụ này sẽ sắp xếp theo thứ tự chữ:

CODE
```
myArray = ["Gary","Will","Jay","Brian"];
myArray.sort();
trace(myArray);
```

Câu lệnh reverse để đảo vị trí sắp xếp của mảng. Ví dụ:

CODE
```
myArray = ["Gary","Will","Jay","Brian"];
myArray.reverse();
trace(myArray);
```

Muốn sắp xếp mảng theo vị trí giảm dần thì sử dụng sort rồi sử dụng reverse.
Sử dụng câu lệnh concat để nối hai mảng lại với nhau. Nó sẽ không làm thay đổi các mảng cũ, mà nó sẽ tạo ra một mảng mới.

CODE
```
myArray = [36,23,63,71,25]
otherArray = [58,97,16];
newArray = myArray.concat(otherArray);
trace(newArray);
```

Chuyển đổi giữa chuỗi và mảng

Chúng ta có thể sử dụng câu lệnh join để đổi từ một mảng thành chuỗi. Câu lệnh này cần một đối số duy nhất là ký tự ngăn cách giữa các phần tử của mảng trong chuỗi. Nếu bạn không truyền tham số này vào thì ký tự mặc định là dấu phẩy. Ví dụ dưới đây trả về 36:23:63:71:25.

CODE
```
myArray = [36,23,63,71,25]
myString = myArray.join(":");
trace(myString);
```

Câu lệnh join ít được sử dụng vì nó không cần thiết lắm, nhưng hàm split lại rất hữu dụng. Nó sẽ chuyển đổi từ một chuỗi sang mảng. Ví dụ như nó sẽ chuyển một chuỗi “36,23,63,71,25” thành một mảng trong ví dụ dưới đây:

CODE
```
myString = "36,23,63,71,25";
myArray = myString.split(",");
trace(myArray);
```

Hãy nghĩ đến chuyện chúng ta có một câu nói được lưu trong một chuỗi muốn chuyển sang mảng, mỗi phần tử trong mảng sẽ chứa một chữ. Xem ví dụ dưới đây nhé:

CODE
```
myString = "This is a test";
myArray = myString.split(" ");
trace(myArray);
```

Làm chữ chuyển động

Trong ví dụ này, chúng ta sẽ lấy từng chữ trong một câu dài và hiển thị vào textfield.
- Tạo movie Flash mới
- Tạo một text field dynamic, cho font chữ to, khoảng 64. Cho text field nằm ở giữa màn hình và canh giữa cho text field. Đặt variable = text.
- Vẽ một shape rồi chọn Insert -> Convert to Movie Clip. Đặt tên cho instance này là Actions rồi kéo nó ra ngoài vùng hiển thị.
- Chèn đoạn code sau vào movie clip đó. Đầu tiên sẽ sử dụng hàm split để tách từng chữ của câu vào trong mảng. Sau đó sẽ khai báo thêm 3 biến nữa. Biến wordNum sẽ lưu một con số là số thứ tự của chữ sẽ hiển thị. Biến frameDelay sẽ lưu số frame để mỗi chữ hiện ra. Biến frameCount sẽ đếm số frame mà một chữ đã đi qua.

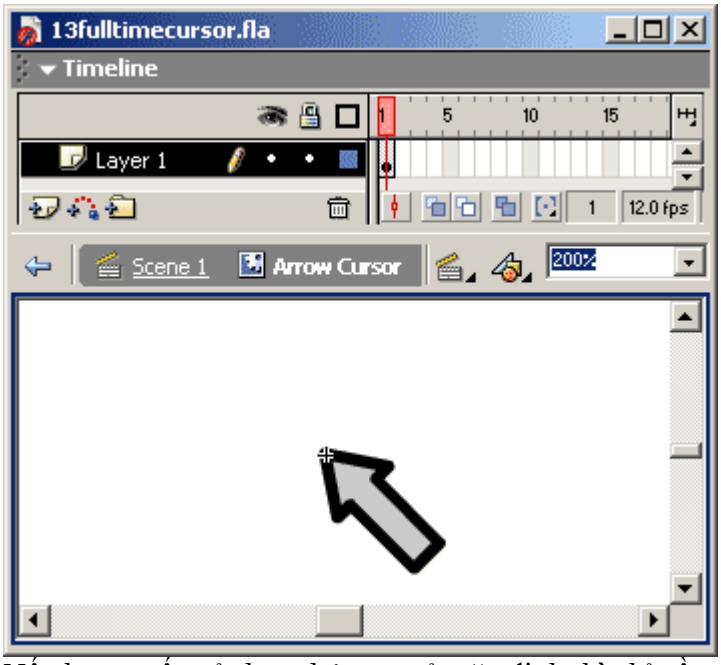
CODE
```
onClipEvent(load) {
  // get the words
  wordList = ("Imagination is more important than knowledge").split(" ");

  // set up variables
  wordNum = 0;
  frameDelay = 6;
  frameCount = frameDelay; // prime for first word
}

onClipEvent(enterFrame) {
  // time for new word
  if (frameCount == frameDelay) {
     _root.text = wordList[wordNum]; // display word
     wordNum++; // next word
     if (wordNum >= wordList.length) wordNum = 0;
     frameCount = 0;
  }
  frameCount++;
}
```

- Nào, bây giờ thì đã xong rồi, hãy chạy thử xem nào

Giờ thứ 13: Sử dụng Rollovers, Rollovers

Tạo hình dáng con trỏ theo ý thích

Việc thay con trỏ mặc định bằng một con trỏ theo ý thích của mình rất đơn giản, chỉ cần sử dụng hàm Mouse.hide() và đặt một movie clip của mình vào vị trí của con trỏ là xong. Con trỏ có thể là bất cứ hình dáng gì cũng được, như là hình mũi tên, hình bàn tay hay một movie clip.
Hình dưới đây là một ví dụ của một movie clip được dùng để làm con trỏ. Chỉ là một hình mũi tên đơn giản nhưng bạn phải chú ý rằng dấu cộng chính giữa movie clip phải nằm ngay đầu của mũi tên.



Nếu bạn muốn sử dụng lại con trỏ mặc định thì chỉ cần gọi Mouse.show()
Một điều cần lưu ý nữa là phải chắc rằng movie clip làm con trỏ của chúng ta phải ở trên tất cả các movie clip khác. Chúng ta có thể chọn Modify -> Arrage ->Bring To Front để đưa movie clip lên đầu nhưng chỉ là trên các movie clip trong layer đó mà thôi. Cho dù bạn có để movie clip của mình lên layer trên cùng thì cũng có thể bị che khuất bởi những movie clip được load vào bằng duplicateMovie và attachMovie. Vì vậy, chúng ta sẽ sử dụng swapDepths() để đưa movie clip này lên trên cùng.
Câu lệnh swapDepths() sẽ đưa movieclip lên một level mới, level có thể là một số nguyên 0, 1, 2… 9999. Vì thế chúng ta sẽ sử dụng lệnh
Cursor.swapDepths(9999);
để đưa movieclip của chúng ta lên trên cùng.

Luyện tập: Tạo con trỏ tĩnh
- Tạo một movie mới
- Tạo một movie clip mới để thay thế cho con trỏ
- Quay trở lại movie đầu tiên, và chúng ta sẽ thay thế con trỏ bằng cách

CODE

```
onClipEvent(load) {
  // hide the real cursor
  Mouse.hide();

  // bring this movie clip to the front
  this.swapDepths(99999);
}
```

- Tiếp theo chúng ta sẽ gắn vị trí của movie clip vào vị trí của con trỏ

CODE

```
onClipEvent(enterFrame) {
  // follow the mouse
  this._x = _root._xmouse;
  this._y = _root._ymouse;
}
```

- Sau cùng, chúng ta sẽ phục hồi lại con trỏ cũ khi kết thúc movie

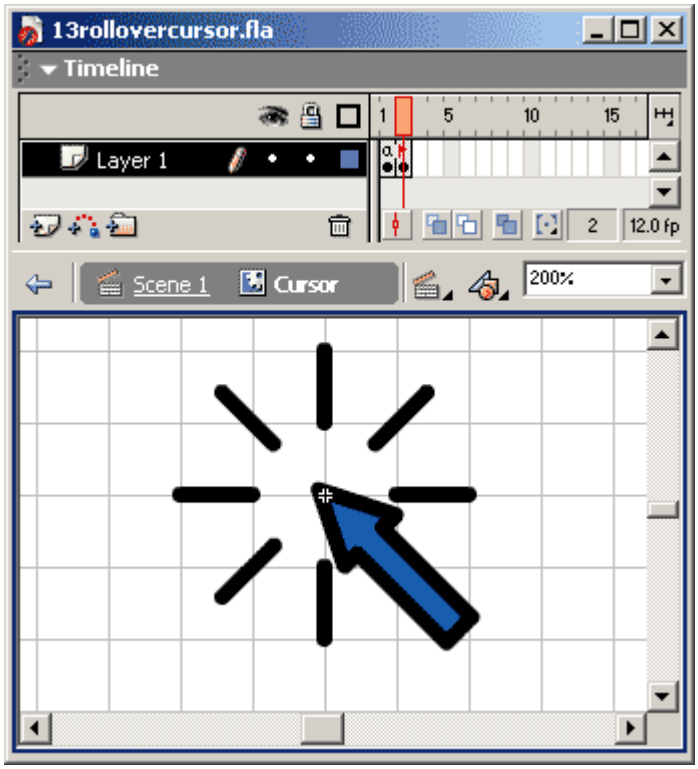
CODE

```
onClipEvent(unload) {
  // show the real cursor again
  Mouse.show();
    }
```

- Cuối cùng là việc chạy thử movie của mình

Luyện tập: Tạo con trỏ động

- Chúng ta có thể sử dụng lại movie trước
- Tạo một button mới trong root. Hãy thử làm cho button có sự thay đổi trong over và down để chúng ta có thể thấy được sự khác biệt.
- Chúng ta sẽ thay đổi một ít trong movie clip làm con trỏ. Tạo một frame thứ hai, vẽ thêm vài nét như hình vẽ



- Đặt tên hai frame là normal và over button
- Đặt câu lệnh stop(); vào frame đầu tiên của movie clip

- Kéo thả một button vào root
- Đặt tên movie clip làm con trỏ là cursor
- Thêm đoạn code sau vào button

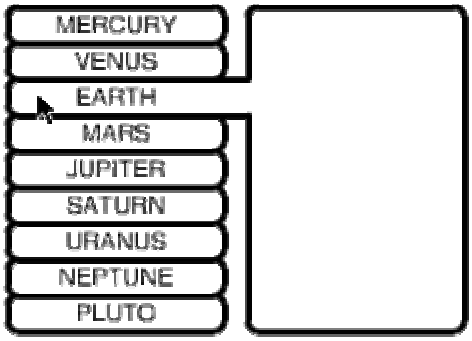
CODE

```
on (rollOver) {
  cursor.gotoAndStop("over button");
}

on (rollOut) {
  cursor.gotoAndStop("normal");
}
```

- Bây giờ hãy chạy thử xem nào , hãy thử click vào button xem sao. Xem kết quả bạn làm có giống hình này không nhé!

Rollovers

Một kỹ thuật thông dụng để hiển thị những thông tin dài là sử dụng Rollovers để đưa ra những thông tin thay vì sử dụng một button để người dùng click vào thì sẽ sang một trang khác.
Ý tưởng của kỹ thuật này là khi người dùng đưa chuột lướt những vùng nào đó. Mỗi vùng sẽ hiển thị cho người dùng xem một thông tin gì đó trên màn hình.
Trong ví dụ sau, sẽ có 9 vùng như thế, mỗi vùng sẽ là một tên của một hành tinh (ở bên trái). Khi chúng ta đưa trỏ chuột qua những vùng đó thì bên phải sẽ xuất hiện thông tin về hành tinh đó. Khi đưa trỏ chuột ra ngoài thì thông tin đó cũng biến mất.



Chúng ta có thể sử dụng AS để làm Rollovers bằng nhiều cách

Rollovers sử dụng button

Chúng ta sẽ sử dụng hai event của button là on(rollOver) và on(rollOut) để viết code

xử lý việc hiển thị thông tin. Hãy xem ví dụ dưới đây:

CODE

```
on (rollOver) {
  information.gotoAndStop("information 1");
}

on (rollOut) {
  information.gotoAndStop("none");
}
```

Rollovers sử dụng movie clip

Flash không có hàm onClipEvent(mouseOver), vì thế chúng ta sẽ sử dụng một hàm khác.
Hàm hitTest sẽ cho chúng ta biết rằng con trỏ chuột có đang ở trên movie clip hay không. Và chúng ta có thể làm như sau

CODE

```
onClipEvent (enterFrame) {
  if (this.hitTest(_root._xmouse,_root._ymouse, true)) {
     _root.information.gotoAndStop("information 1");
  }  else {
     _root.information.gotoAndStop("none");
  }
}
```

Nhưng làm như vậy vẫn còn một vấn đề nữa. Bởi vì movie clip của chúng ta sẽ vẫn tiếp tục chạy từ frame này sang frame khác, chúng ta đã gọi hàm gotoAndStop() để dừng lại. Hãy tưởng tượng chúng ta có nhiều rollovers. Cái đầu tiên sẽ đưa movie clip information về frame none, mặt khác thì rollovers khác lại đưa movie clip information về một nơi khác. Điều này sẽ làm xảy ra xung đột. Vì vậy, chúng ta sẽ làm như sau. Chúng ta sẽ ghi nhớ lại rằng con trỏ chuột có đang ở trên movie clip hay không. Nếu có, nó sẽ thi hành lệnh khi con trỏ chuột ra ngoài movie clip. Còn nếu không, nó sẽ thi hành lệnh khi con trỏ chuột đi vào movie clip.
Để làm điều này, chúng ta sẽ sử dụng biến over, phụ thuộc vào vị trí của con trỏ chuột mà nó sẽ mang giá trị true hoặc false. Đối với mỗi frame, chúng ta sẽ sử dụng hàm hitTest để kiểm tra vị trí của con trỏ chuột. Nếu vị trí con trỏ đối lập với over thì sẽ xảy ra sự thay đổi. Chúng ta hãy xem đoạn code sau:

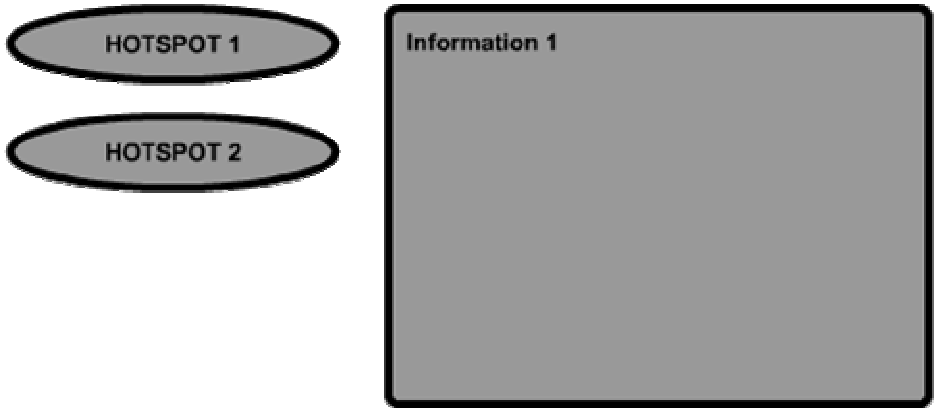
CODE

```
onClipEvent (load) {
  over = false;
}
```

```
onClipEvent (enterFrame) {
  // kiểm tra xem liệu con trỏ chuột có đang di chuyển qua movie clip không
  testOver = (this.hitTest(_root._xmouse,_root._ymouse, true));

  if (testOver and !over) {
     _root.information.gotoAndStop("information 1");
     over = true;
  }else if (!testOver and over) {
     _root.information.gotoAndStop("none");
     over = false;
  }
}
```

Rollovers sử dụng frame

Như đã nói ở trên, có rất nhiều cách để làm rollovers. Ở đây, chúng ta sẽ bàn về một cách khác.
Thay vì sử dụng movie clip cho mỗi thông tin cần hiển thị, chúng ta sẽ sử dụng timeline chính để lưu thông tin. Frame đầu tiên sẽ là frame none, và các frame tiếp theo sẽ chứa các thông tin.
Chúng ta hãy xem hình bên dưới



Ví dụ trên sử dụng button cho các hotspot, và chúng ta hãy lưu ý rằng các button sẽ xuất hiện trong tất cả các frame, nhưng thông tin cần hiển thị thì chỉ xuất hiện ở một vài frame.
Bây giờ công việc viết code của chúng ta gần giống như làm rollovers bằng button. Nhưng chúng ta không cần phải gọi hàm gotoAndStop từ movie clip information.

CODE

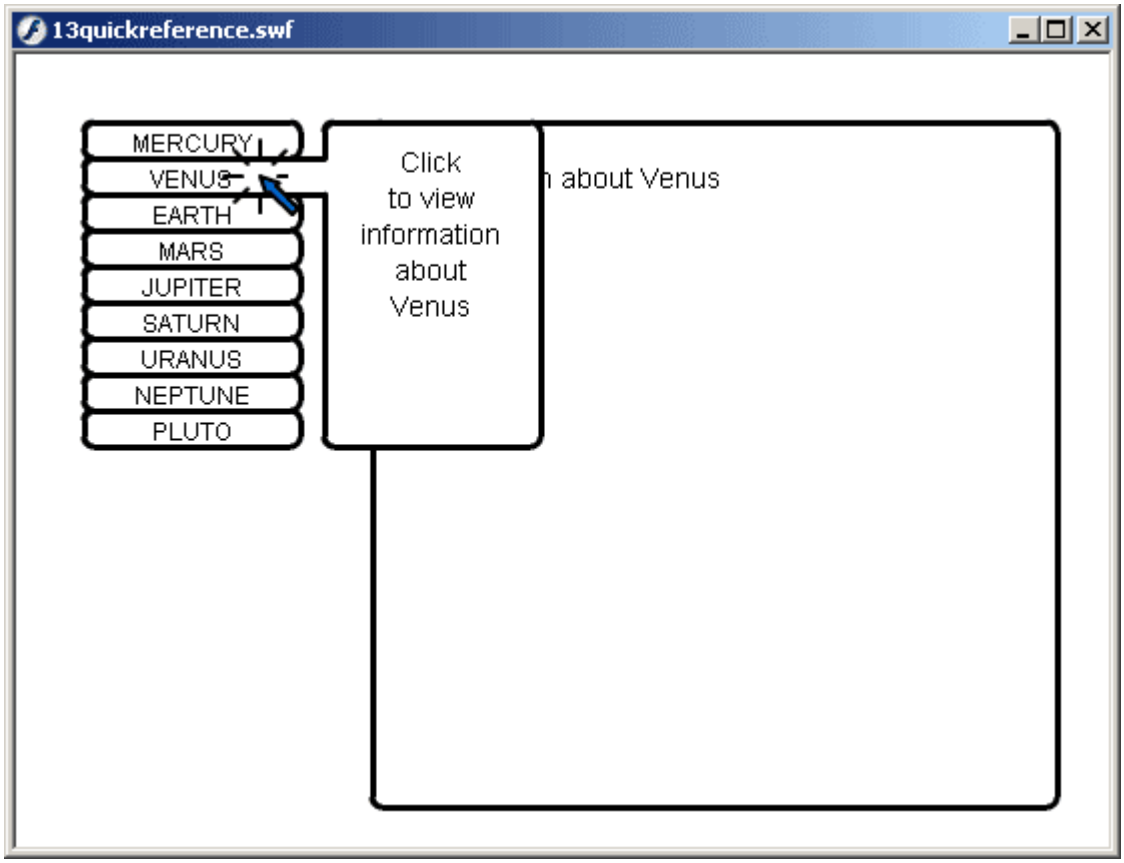
```
on (rollOver) {
  gotoAndStop("information 1");
}

on (rollOut) {
```

```
  gotoAndStop("none");
}
```

Lợi thế của việc sử dụng cách này là chúng ta có thể thay đổi những thông tin trong các frame rất dễ dàng, chúng ta không phải mổ xẻ vào từng movie clip. Nếu bạn quen sử dụng nhiều frame thì đây là một cách tốt cho bạn
Lưu ý rằng cả ba cách trên đều cho một kết quả như nhau, không có sự khác biệt. Ở đây, chúng tôi chỉ muốn trình bày cho các bạn thấy những cách làm khác nhau mà thôi!

Luyện tập: Hiện thị thông tin

Nào, bây giờ các bạn hãy thử sử dụng những hiểu biết của mình về cách thay đổi con trỏ và rollovers để làm một chương trình xem nào! Chương trình này sẽ hiển thị những thông tin về các hành tinh.
Chúng ta sẽ có 9 hotspot, mỗi hotspot là một hành tinh, mỗi hotspot sẽ hiển thị một hộp thông tin khi đưa trỏ chuột ngang qua. Mỗi hotspot là một button, vì thế người dùng có thể click vào để di chuyển đến một frame khác để xem những thông tin về hành tinh đó.



- Đầu tiên, hãy tạo một movie mới trong Flash. Movie này sẽ có 10 frame, từ frame 2 cho đến frame 10 sẽ chứa thông tin của các hành tinh. Đặt tên frame 1 là none và để trống vùng hiển thị thông tin.
Đặt 10 button ở bên trái, mỗi cái cho một hành tinh. Một movie clip summary sẽ xuất hiện để hiển thị thông tin vắn tắt về các hành tinh khi đưa trỏ chuột ngang qua. Movie clip này cũng sẽ chứa 10 frame: 1 frame trống và 9 frame chứa thông tin của 9 hành

tinh.
Chúng ta cũng phải chú ý việc sử dụng layer cũng rất quan trọng, trong ví dụ này thì chúng ta sẽ sử dụng 3 layer và movie clip summary sẽ được đặt ở layer trên cùng
- Đặt lệnh stop(); vào frame đầu tiên, và cũng đặt trong frame đầu tiên của movie clip summary
- Nào, hãy viết code nhé!

CODE

```
on (rollOver) {
  summary.gotoAndStop("mercury");
}

on (rollOut) {
  summary.gotoAndStop("none");
}
```

- Những button ở trên sẽ đưa người dùng đến những frame khác nhau để xem thông tin về hành tinh, chúng ta lại viết code cho các button

CODE

```
on (release) {
  gotoAndStop("mercury");
}
```

Lưu ý rằng chúng ta có hai frame tên mercury, một frame ở timeline chính và một ở trong movie clip summary
- Tạo một movie clip để thay thế con trỏ chuột, và viết code như sau

CODE

```
onClipEvent(load) {
  Mouse.hide();

  this.swapDepths(99999);
}

onClipEvent(enterFrame) {
  this._x = _root._xmouse;
  this._y = _root._ymouse;
}

onClipEvent(unload) {
  Mouse.show();
}
```

- Tiếp theo chúng ta sẽ làm cho con trỏ chuột thay đổi khi đưa con trỏ ngang qua các button. Hãy thêm đoạn code sau vào phần code của button

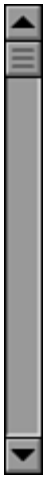
CODE

```
on (rollOver) {
  summary.gotoAndStop("mercury");
  cursor.gotoAndStop("over button");
}

on (rollOut) {
  summary.gotoAndStop("none");
  cursor.gotoAndStop("normal");
}

on (release) {
  gotoAndStop("mercury");
}
```

Bây giờ thì movie của bạn đã xong. Bạn hãy chạy thử xem sao

Giờ thứ 14: Thành phần Scroll, Scrolling

Mặc dù scrollbar rất quen thuộc trong các ứng dụng Mac, Windows, các trình duyệt web… nhưng mấy ai hiểu được nó hoạt động như thế nào. Lý do đơn giản là do đây là một trong những thành phần trực quan, rất dễ xây dựng mà không cần phải viết code nhiều. Mọi người sử dụng nó nhưng không suy nghĩ nhiều về nó.
Kết quả là khi những nhà phát triển sử dụng Flash để tạo ra những scrollbar của riêng họ thì gặp khó khăn. Vì vậy, chúng ta hãy cùng nhau tìm hiểu 4 thành phần cơ bản của một scrollbar và hãy tìm hiểu scrollbar là gì.
Hình dưới đây cho chúng ta thấy những thành phần cơ bản của một scrollbar là: mũi tên lên, mũi tên xuống, thanh trượt và khay trượt.



Thanh trượt (Slider)

Thanh trượt phục vụ nhiều mục đích. Đầu tiên, thanh trượt sẽ trượt trên khay trượt để cho chúng ta thấy vị trí của khối văn bản chúng ta đang xem. Nếu thanh trượt ở trên cùng thì chúng ta đang xem dòng đầu tiên của văn bản, còn nếu thanh trượt ở cuối thì chúng ta đang xem dòng cuối cùng.
Như đã nói, thanh trượt sẽ được kẹp chặt và trượt trên khay trượt. Khi chúng ta kéo thanh trượt trượt trên khay trượt thì khối văn bản sẽ được cập nhật vị trí thích hợp.
Mới đây, scrollbar có thêm một đặc tính mới. Thay vì kích thước của thanh trượt sẽ bị gắn sẵn với một giá trị thì kích thước này sẽ được thay đổi tùy vào độ dài văn bản. Vì thế, vị trí đầu tiên của thanh trượt sẽ ứng với dòng đầu tiên của văn bản và vị trí cuối cùng sẽ ứng với dòng cuối cùng. Ví dụ một textbox có scrollbar hiển thị được 10 dòng của một văn bản có 100 dòng thì chiều cao thanh trượt sẽ là 10% so với khay trượt. Nhưng chúng ta sẽ không bàn vấn đề này ở đây.

Khay trượt (Bar)

Khay trượt có chức năng chính là chứa thanh trượt và cho thanh trượt trượt trên nó. Chiều dài của khay trượt phụ thuộc vào độ dài của văn bản. Khay trượt còn có một chức năng nữa là khi ta click vào khay trượt thì khối văn bản sẽ di chuyển một trang. Khi click vào phần trên của thanh trượt thì khối văn bản sẽ di chuyển đến trang trước, còn nếu click vào phần dưới thanh trượt thì khối văn bản sẽ di chuyển đến trang sau.

Các mũi tên

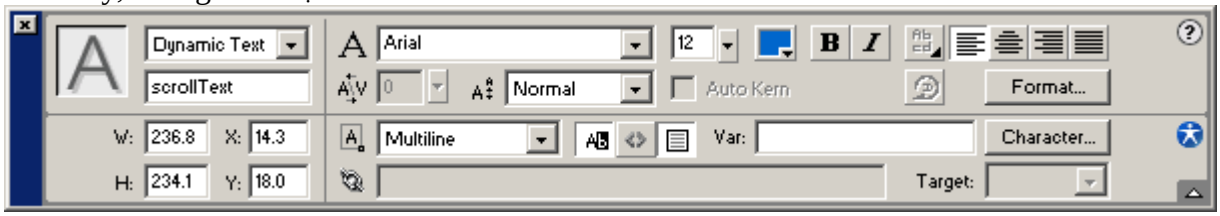
Mũi tên lên và xuống là hai thành phần đơn giản nhất của scrollbar, nó chỉ có chức năng cho người dùng di chuyển khối văn bản từng dòng một.

Những thuộc tính chung

Scrollbar có một số thuộc tính chung mà chúng ta cần phải xem xét. Đầu tiên, các thành phần của scrollbar chỉ hoạt động khi chúng ta click vào, và sẽ tiếp tục hoạt động cho đến khi chúng ta thả nút chuột ra. Lấy ví dụ: nếu người dùng click vào mũi tên xuống để di chuyển một khối văn bản thì khối văn bản sẽ di chuyển từng dòng một cho đến khi chúng ta thả nút chuột ra.
Một vấn đề nữa là vị trí thanh trượt phải được cập nhật liên tục khi những thành phần khác được kích hoạt.

Scroll một văn bản

Đầu tiên, chúng ta mở khung Properties ra để đặt tên lại cho text field. Như trong hình dưới đây, chúng ta sẽ đặt tên cho text field là scrollText

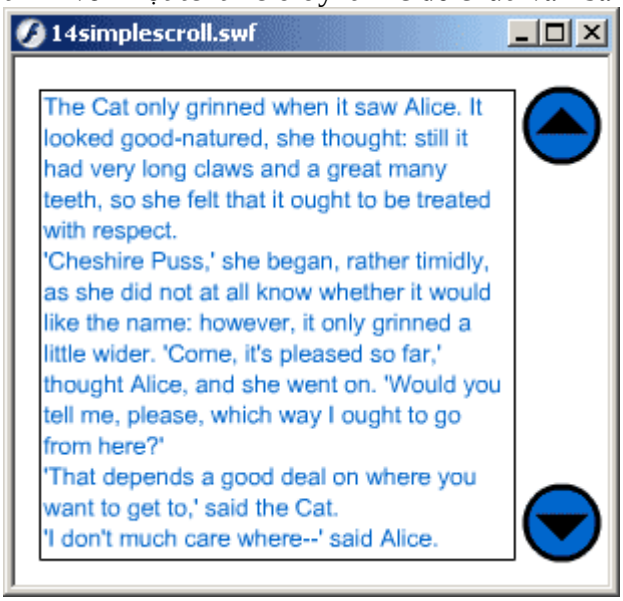


Sau đó, chúng ta có thể viết code cho nó lấy dữ liệu vào. Nhưng trước hết cần đặt cho text field của chúng ta một scroll.
Khi đã chọn vào text field rồi thì bạn có thể thay đổi kích thước của text field bằng cách kéo hình ô vuông ở góc dưới bên phải của text field. Còn nếu không đặt kích

thước thì text field sẽ vừa đủ để hiểu thị nội dung bên trong.
Thay vì điều chỉnh bằng hình vuông màu trắng thì lúc đó, chiều cao của text field sẽ được tự điều chỉnh theo kích thước nội dung bên trong, có nghĩa là nếu văn bản bên trong text field đó có nhiều dòng thì chúng ta không thể làm cho text field chỉ hiện một vài dòng để scroll. Muốn là xuất hiện scroll thì chúng ta hãy giữ phím Shift và click vào hình vuông trắng đó để trở thành hình vuông đen, lúc này chúng ta có thể điều chỉnh kích thước text field theo ý muốn. Như vậy là chúng ta đã kích hoạt scroll cho text field. Còn một số thuộc tính khác lien quan đến scroll như
Thuộc tính scroll sẽ cho chúng ta biết dòng nào trong văn bản sẽ xuất hiện đầu tiên. VD như scroll = 1 thì dòng đầu tiên sẽ được hiển thị đầu tiên, nếu scroll = 2 thì dòng thứ hai sẽ hiển thị và lúc này thì dòng 1 sẽ không thấy được.
Thuộc tính maxscroll cho chúng ta biết giá trị lớn nhất của scroll
Thuộc tính scroll và bottomscroll cho chúng ta biết chính xác dòng văn bản nào ở trên đầu và dòng nào ở cuối
Để scroll văn bản lên hay xuống bạn cần phải tăng hoặc giảm giá trị của scroll. Thế là xong!

Luyện tập: Thiết kế một chương trình scroll văn bản đơn giản

Bài tập này thật ra rất dễ. Trong ví dụ dưới đây, chúng ta sẽ xây dựng một chương trình với một text field dynamic để chứa văn bản và hai button như hình dưới đây



- Tìm một đoạn văn bản nào đó để dán vào text field (tìm đoạn nào dài dài tí)
- Bây giờ hãy tạo một movie mới trong Flash
- Sử dụng công cụ Text Tool để tạo một text field
- Mở phần Properties ra và đặt tên cho text field của chúng ta là scrollText. Đặt thuộc tính Mutiline và Show Border Around Text
- Dán đoạn văn bản mà bạn đã chuẩn bị vào text field
- Tạo hai button giống như hai hình vẽ ở trên. Một button để điều khiển đi lên, một để đi xuống.
- Nhập đoạn code sau cho button đi lên:

CODE

```
on (press) {
```

```
  scrollText.scroll--;
}
```

- Và nhập đoạn code này cho button đi xuống:

CODE

```
on (press) {
  scrollText.scroll++;
}
```

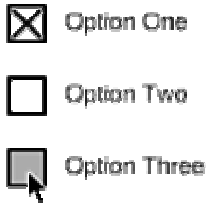
- Ok, bây giờ bạn hãy chạy thử movie của mình xem sao. Bạn hãy thử click vào button để scroll văn bản

Giờ thứ 15: Các thành phần nhập liệu

Các bạn có thể tạo được nhiều thành phần nhập liệu bằng Action Script, chắc hẳn các bạn đã gặp các thành phần này trong các thẻ HTML. Trong chương này, các bạn sẽ học cách làm checkbox, radiobutton bằng Action Script. Bạn cũng sẽ học được cách làm sao để chuyển từ thành phần này sang thành phần khác trong form bằng cách nhấn nút TAB, và làm sao để hạn chế nội dung người dùng nhập vào
Các nội dung trong chương này:
- Cách tạo checkbox
- Cách tạo radiobutton
- Sử dụng TAB để chuyển từ thành phần này sang thành phần khác
- Hạn chế nội dung nhập liệu

Tạo Checkbox

Trong Hour 8, các bạn đã biết cách tạo một selectable movie clip. CheckBox cũng là một Selectable movie clip giống với các checkbox chuẩn sử dụng trong các hệ điều hành Mac và Windows.
Để tạo ra một Checkbox bạn cần 2 button và 1 movie clip. Button đầu tiên thể hiện trạng thái Off của Checkbox, có nghĩa là khi checkbox chưa được chọn. Một button thể hiện trạng thái On, có nghĩa là cũng checkbox đó nhưng đã được chọn.
Hình dưới đây là 3 checkbox làm ví dụ.



Check box đầu tiên đang được chọn, cái thứ 2 chưa được chọn, cái thứ 3 thì người dùng đang chuẩn bị chọn.
Các bạn có biết cách nào để cho 3 thành phần của checkbox hoạt động chung như vậy được không? Rất đơn giản, button Off sẽ được đưa vào một movie clip riêng đặt ở frame đầu tiên, bấm F6 để tạo frame kế tiếp và đặt button On vào frame thứ 2. Đặt tên

frame 1 là Off, frame 2 là On. Tiếp theo, cho một câu lệnh stop(); vào frame 1 để dùng movie clip lại ngay đó. Trong mỗi button đó sẽ có một đoạn code gọi một hàm ở ngoài time line của movie clip có chứa 2 button để xử lý khi mỗi nút được nhấn.

CODE
```
on (release) {
  pressButton();
}
```

Ngoài ra, trong frame Off (frame 1) sẽ còn chứa một đoạn code nữa ngoài câu lệnh stop();

CODE
```
state = false;

function pressButton() {
  state = !state;
  if (state) {
     gotoAndStop("on");
  }  else {
     gotoAndStop("off");
  }
}
```

Biến state để kiểm tra trạng thái của checkbox. Khi click một button thì hàm pressButton sẽ được gọi. Trong đó, sẽ đổi lại trạng thái của state, có nghĩa là lúc đầu là false (chưa chọn), sau khi click sẽ thành true (đã chọn). Tiếp theo, hàm pressButton sẽ kiểm tra giá trị của biến state để đưa người dùng đến frame thích hợp. Nếu state = true thì sẽ nhảy đến frame On, còn nếu state = false thì sẽ nhảy đến frame Off.

Tạo RadioButton

Việc tạo một Radiobutton sẽ phức tạp hơn một chút so với checkbox. Các Radiobutton sẽ được gom vào một nhóm có quan hệ với nhau.
Checkbox được sử dụng trong những trường hợp chọn lựa không có tính loại trừ, có nghĩa là người dùng có thể chọn nhiều checkbox. Ngược lại, Radiobutton được sử dụng trong những trường hợp có tính loại trừ, có nghĩa là trong một nhóm Radiobutton thì chỉ có một checkbox được chọn tại một thời điểm, không có chuyện 2 Radiobutton đều được chọn. Nếu bạn đã chọn một RadioButton, khi bạn chọn qua một Radiobutton khác thì chọn lựa cũ sẽ tự động mất đi, chuyển qua Radiobutton mới.
Hình dưới đây là một nhóm Radiobutton

RadioButton đầu tiên đang được chọn, nhưng người dùng đang chuẩn bị chọn

RadioButton thứ 2, nếu người dùng chọn RadioButton 2 hoặc bất kỳ cái nào khác thì RadioButton đầu tiên sẽ tự động mất chọn lựa.
Một RadioButton đơn giản cũng tương tự như một Checkbox, một movie clip có 2 frame để chứa 2 button biểu hiện 2 trạng thái của RadioButton. Frame đầu tiên chứa một vòng tròn rỗng, frame thứ 2 chứ một vòng tròn với dấu chấm tròn ở giữa.
Điểm khác nhau giữa Checkbox và RadioButton chính là code của chúng. Code của RadioButton sẽ phức tạp hơn code của CheckBox.
Phần đầu code của Radiobutton sẽ được viết ở frame đầu tiên. Nó được viết ở ngoài, không nằm trong hàm nào cả, điều này có nghĩa là đoạn code này sẽ chạy khi load movie clip.

CODE

```
stop();

// kiểm tra có phải là RadioButton đầu tiên trong nhóm hay không
if (_parent.radioButtons == undefined) {
  // tạo một array RadioButton
  _parent.radioButtons = new Array();

  // RadioButton đầu tiên mặc định được chọn
  gotoAndStop("on");
  state = true;
}  else {
  // các RadioButton khác không được chọn
  state = false;
}

// chèn array RadioButton ra ngoài level ngoài
_parent.radioButtons.push(this);
```

Khi người dùng click vào button, thì hàm turnOn sẽ được gọi. Điều đầu tiên là hàm turnOn sẽ duyệt qua tất cả các RadioButton (các movie clip), gọi hàm turnOff cho từng RadioButton. Nói đơn giản có nghĩa là khi click vào một RadioButton thì trước tiên tất cả các RadioButton trong nhóm đều quay về trạng thái Off, sau đó sẽ chuyển trạng thái của RadioButton được chọn thành On.

CODE

```
function turnOn() {
  // chuyển tất cả thành OFF
  for(var i=0;i<_parent.radioButtons.length;i++) {
     _parent.radioButtons[i].turnOff();
  }

  // chuyển RadioButtond được chọn thành ON
  gotoAndStop("on");
  state = true;
}
```

Tiếp theo là đoạn code cho hàm turnOff()

CODE

```
function turnOff() {
  gotoAndStop("off");
  state = false;
}
```

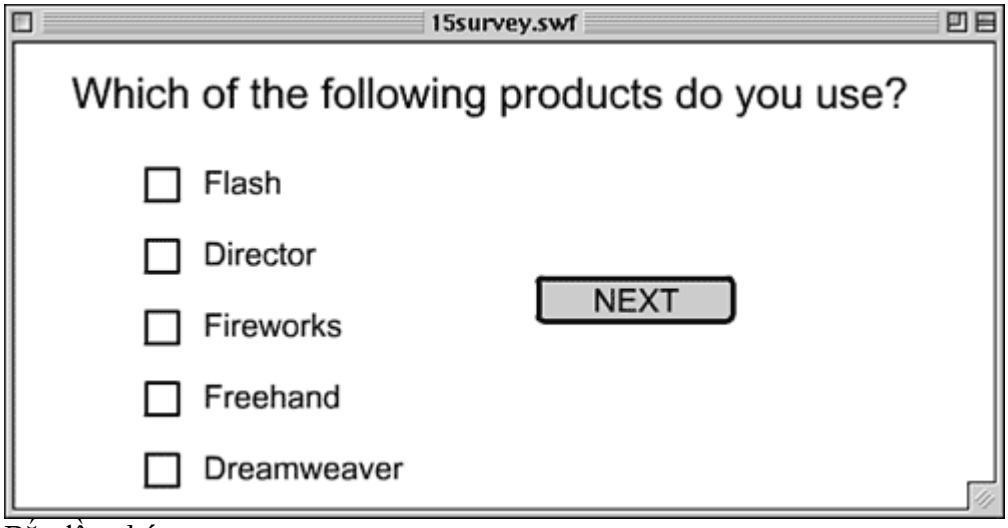
Tiếp theo là xây dựng hàm getValue()để kiểm tra xem RadioButton nào đang được chọn. Hàm này sẽ được một movie khác gọi. Hàm này rất đơn giản, nó sẽ duyệt qua tất cả các RadioButton trong array RadioButton xem cái nào đang được chọn.

CODE

```
function getValue() {
  // duyệt tất cả các RadioButton
  for(var i=0;i<_parent.radioButtons.length;i++) {
     // tìm RadioButton nào đang được chọn
     if (_parent.radioButtons[i].state) {
        return(_parent.radioButtons[i]._name);
     }
  }

  // nếu không có cái nào được chọn thì trả về một chuỗi rỗng “”
  return "";
}
```

Luyện tập: Chương trình trắc nghiệm

Bây giờ hãy cùng nhau áp dụng những thứ đã học được về CheckBox và RadioButton để làm một chương trình trắc nghiệm đơn giản nhé. Mỗi frame sẽ chứa một câu hỏi riêng.
Ví dụ frame đầu tiên sẽ chứa câu hỏi như hình dưới đây, và có những câu trả lời ở dưới



Bắt đầu nhé

- Tạo một movie mới trong Flash
- Tạo Checkbox như đã được học rồi đó.
- Rồi sau đó kéo tạo 5 bản của movie clip checkbox vào, đặt tên là: Flash, Director, Fireworks, Freehand, and Dreamweaver.
- Tạo các câu trả lời và một static text kế bên Checkbox như hình ở trên.
- Tạo một button Next để chuyển đến câu hỏi tiếp theo.
- Chèn đoạn code sau vào frame đầu tiên

CODE

```
results = new Array();
stop();
```

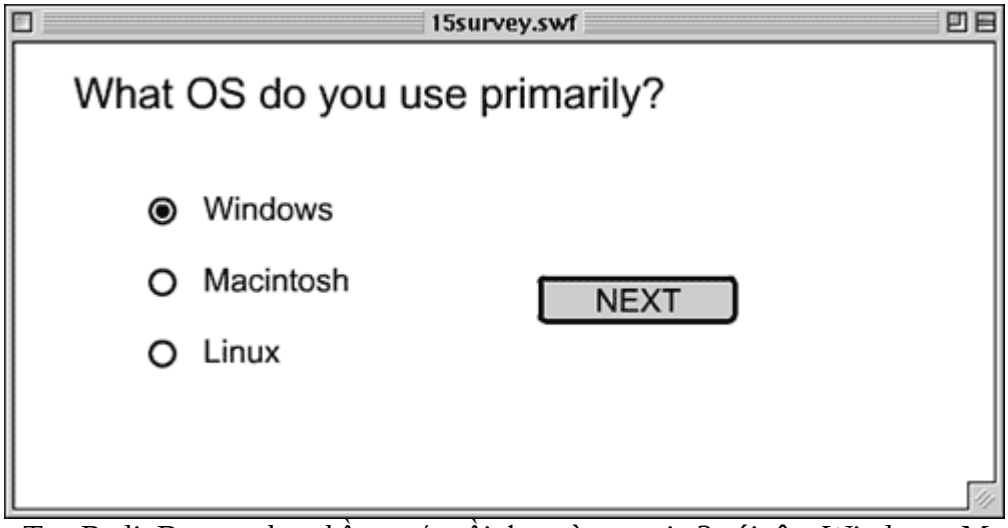
Đoạn code trên có nhiệm vụ tạo một mảng results để lưu kết quả của các câu trả lời, đồng thời cũng dùng movie tại đây.

- Khi nhấn nút Next thì sẽ chuyển sang câu hỏi tiếp theo và lưu câu trả lời hiện thời vào mảng results

CODE

```
on (release) {
  if (Flash.state) results.push("Flash");
  if (Director.state) results.push("Director");
  if (Fireworks.state) results.push("Fireworks");
  if (Freehand.state) results.push("Freehand");
  if (Dreamweaver.state) results.push("Dreamweaver");
  nextFrame();
}
```

- Tiếp theo, trong frame thứ hai sẽ là câu hỏi thứ hai. Vì câu hỏi thứ hai chỉ cho chọn một lựa chọn nên chúng ta sẽ sử dụng RadioButton như hình dưới đây



- Tạo RadioButton như phần trước rồi đưa vào movie 3 cái tên: Windows, Macintosh, and Linux
- Chèn nội dung câu trả lời vào luôn như hình ở trên
- Copy nút Next vào frame 2 nhưng chúng ta sẽ thay đoạn code bằng một đoạn code khác.

CODE

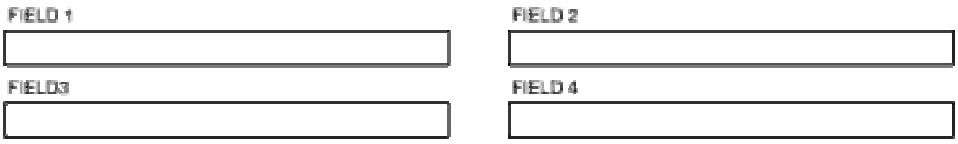
```
on (release) {
  results.push(Windows.getValue());
  nextFrame();
}
```

Đoạn code trên sử dụng hàm getValue để kiểm tra xem RadioButton nào đang được chọn

- Ok, đến đây thì bạn tự làm tiếp những câu hỏi tiếp theo nhé, tương tự như vậy thôi. Nhưng mà hãy nhớ là tên các thành phần phải khác nhau nhé. Các bạn cũng có thể sử dụng hàm trace để đưa kết quả những câu trả lời ra cửa sổ Output để xem.
- Trong Hour 18, các bạn sẽ học về cách đưa dữ liệu lên server, khi đó các bạn có thể cải tiến chương trình này để đưa câu trả lời lên server để kiểm tra.

Sử dụng Tab để chuyển từ thành phần này sang thành phần khác

Những người sử dụng Flash 5 luôn than phiền về việc không thể sử dụng tab để chuyển đổi giữa các thành phần như vậy. Công việc chuyển đổi bằng cách nhấn phím Tab này được gọi là Tab Order, chúng ta thường thấy việc này trong hầu hết các chương trình chuyên nghiệp, ví dụ như trong một trình duyệt web, điều này sẽ mang lại nhiều thuận tiện cho người sử dụng, nó sẽ chuyển đến thành phần logic tiếp theo trong chương trình.
Tuy nhiên, sự lựa chọn thành phần logic tiếp theo của Flash không phải lúc nào cũng đúng. Vì thế, Flash chỉ có thể chứa vị trí của thành phần mà thôi. Lấy ví dụ như trong hình bên dưới, Flash sẽ hiểu rằng thành phần tiếp theo thành phần đầu tiên sẽ là cái ở dưới nó chứ không phải là cái kế bên.



May mắn là chúng ta có thể đặt lại Tab order cho các thành phần bằng cách đặt lại thuộc tính tabIndex. Nếu chúng ta có 4 text field như hình trên: text1, text2, text3, text4 thì chúng ta có thể sử dụng đoạn code sau:

CODE
```
text1.tabIndex = 1;
text2.tabIndex = 2;
text3.tabIndex = 3;
text4.tabIndex = 4;
```

Điều chú ý khi sử dụng tabIndex là phải chú ý đến label của các textfield và tránh sử dụng lại một số nhiều lần, điều này sẽ làm cho Flash bị lẫn lộn.
Một điều nữa là cho dù là mặc định của Flash hay là bạn sử dụng tabIndex thì Flash cũng không tự động đặt focus cho thành phần đầu tiên, vì thế, bạn phải tự làm điều này. Để làm được như vậy, bạn sử dụng một lệnh trong đối tượng Selection để báo cho Flash biết thành phần mặc địnhđược đặt focus

CODE
```
Selection.setFocus(text1);
```

Bạn có thể sử dụng lệnh Selection.setFocus bất cứ lúc nào cũng được để chuyển đến một thành phần mong muốn. Việc này sẽ rất quan trọng, bạn sẽ đặt con trỏ vào textfield thay vì bắt người dùng phải click vào textfield trước khi gõ.
Bạn có thể sử dụng Selection.getFocus để kiểm tra xem thành phần nào đang được focus. Đoạn code dưới đây sẽ là một ví dụ. Khi người dùng chuyển focus sang một thành phần khác thì bạn sẽ biết người dùng chuyển đến đâu.

CODE
```
Selection.addListener(this);
this.onSetFocus = function(oldFocus, newFocus) {
  trace(oldFocus+","+newFocus);
}
```

Hạn chế nhập liệu

Khi người dùng nhập dữ liệu vào một textfield, cũng có lúc bạn muốn hạn chế việc nhập liệu đó. Ví dụ trong ô năm sinh, bạn chỉ muốn người dùng nhập số vào, không cần phải nhập chữ. Bạn có thể hạn chế những ký tự được phép nhập vào textfield bằng cách đặt giá trị của thuộc tính restrict của textfield đó. Nếu không đặt giá trị thì textfield có thể nhận tất cả các ký tự. Nhưng nếu thuộc tính restrict của textfield là một chuỗi ký tự thì chỉ có những ký tự trong chuỗi đó mới được chấp nhận. Dưới đây

là một ví dụ về hạn chế nhập liệu, người dùng chỉ có thể nhập số mà thôi

CODE
```
text1.restrict = "01234567890";
```

Còn dưới đây là một ví dụ nữa nếu ô nhập liệu là email

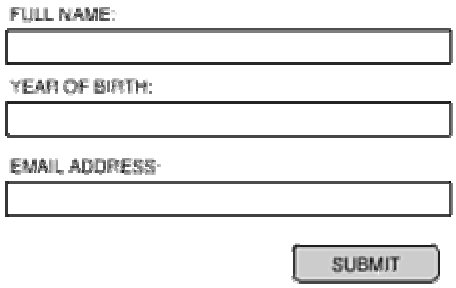
CODE
```
text2.restrict = "abcdefghijklmnopqrstuvwxyz0123456789@.-_";
```

Một điều chú ý là cả các ký tự in hoa và in thường đều đươc chấp nhận trong text2
Bạn cũng có thể hạn chế số ký tự được phép nhập vào một textfield. Cái này bạn cũng có thể không cần phải dùng AS, có thể đặt thuộc tính trực tiếp trong khung Properties

CODE
```
text1.restrict = "01234567890";
text1.maxChars = 4;
```

Luyện tập: Kiểm tra dữ liệu nhập

Bây giờ chúng ta sẽ làm một chương trình gồm một form nhập liệu, yêu cầu nhập vào tên, năm sinh, email. Và chúng ta sẽ kiểm tra các thông tin nhập vào này
Tên người dùng ít nhất phải có 3 ký tự. Năm sinh phải có 4 số và đó là những năm trong khoảng 100 năm trước đến nay. Còn email ít nhất phải có 7 ký tự và có dạng a@b.c, a, b có thể là tuỳ ý nhưng c phải có ít nhất là 3 ký tự và bắt buộc phải có ký hiệu @. Đó là những yêu cầu cơ bản. Nào, bắt đầu nhé!
- Tạo một movie mới
- Tạo 3 textfield cho các nội dung nêu trên, đặt tên là userName, userYear, và userEmail. Đặt các variable tương ứng là userNameText, userYearText, and userEmailText. Bạn cũng cần phải tạo một dynamic textfield liên kết với variable feedback. Tạo một nút Submit. Movie của bạn sẽ giống như hình dưới này nhé



- Đặt đoạn code sau vào frame đầu tiên để dừng lại và thiết lập các thuộc tính để hạn chế nhập liệu

CODE
```
stop();

// hạn chế chiều dài tối đa của tên là 64 ký tự
```

```
userName.maxChars = 64;

// năm sinh phải có 4 số
userYear.restrict = "01234567890";
userYear.maxChars = 4;

// hạn chế dữ liệu email
userEmail.restrict = "abcdefghijklmnopqrstuvwxyz0123456789@.-_";
userEmail.maxChars = 128;
```

- Tiếp theo là đặt con trỏ vào text field userName lúc movie mới bắt đầu

CODE
```
Selection.setFocus(userName);
```

- Để kiểm tra khi nào người dùng đã nhập liệu xong, chúng ta sẽ thêm một listener để bắt event như đoạn code dưới đây

CODE
```
Selection.addListener(this);
```

Nó sẽ báo cho chúng ta biết khi nào xảy ra event onSetFocus
- Tiếp theo đặt giá trị của biến ignoreSetFocus là False, chúng ta sẽ sử dụng đến biến này sau

CODE
```
ignoreSetFocus = false;
```

- Tiếp theo chúng ta viết hàm onSetFocus để bắt event.

CODE
```
this.onSetFocus = function(oldFocus, newFocus) {
  // this is a focus reset, so ignore
  if (ignoreSetFocus) {
     ignoreSetFocus = false;
     return(0);
  }
  // use the appropriate check function
  if (oldFocus == userName) {
     ret = checkUserName();
  }  else if (oldFocus == userYear) {
     ret = checkUserYear();
  }  else if (oldFocus == userEmail) {
     ret = checkUserEmail();
  }
```

```
  if (!ret) {
     // ignore this focus change and go back
     ignoreSetFocus = true;
     Selection.setFocus(oldFocus);
  }
}
```

Hàm này sẽ nhận hai đối số. Đối số thứ nhất là textfield trứơc khi chuyển focus và đối số thứ hai là textfield sau khi chuyển focus. Hàm checkUserName sẽ kiểm tra tên người dùng

CODE

```
// tên phải có ít nhất 3 ký tự
function checkUserName() {
  if (userNameText.length < 3) {
     feedback = "Bạn phải nhập ít nhất 3 ký tự"
     return(false);
  }

  // quay trở lại feedback
  feedback = "";
  return(true);
}
```

Hàm checkUserYear sẽ kiểm tra năm nhập vào

CODE

```
// năm phải từ khoảng 100 năm đến nay
function checkUserYear() {
  // lấy năm
  today = new Date();
  thisYear = 1900+today.getYear();

  // kiểm tra đã nhập
  if (parseInt(userYearText) == Math.NaN) {
     feedback = "Bạn phải nhập năm sinh.";
     return(false);

  // nếu năm sinh quá sớm (không thật)
  }  else if (parseInt(userYearText) < thisYear-100) {
     feedback = "Bạn phải nhập đúng năm sinh, từ 100 năm trước đến nay";
     return(false);

  // nếu năm sinh là ở trong tương lai :)
  }  else if (parseInt(userYearText) > thisYear) {
  feedback = "Bạn phải nhập đúng năm sinh";
  return(false);
  }
```

```
  // quay trở lại
  feedback = "";
  return(true);
}
```

Tiếp theo là hàm chechUserEmail để kiểm tra email

CODE

```
// kiểm tra email
function checkUserEmail() {
  if (userEmailText.length < 7) {
     feedback = "Email quá ngắn";
     return(false);
  } else if (userEmailText.indexOf("@") == -1) {
     feedback = "Thiếu ký tự @";
     return(false);
  } else if (userEmailText.indexOf(".") == -1) {
     feedback = "Thiếu dấu chấm (.)";
     return(false);
  } else if (userEmailText.indexOf("@") > userEmailText.indexOf(".")) {
     feedback = "@ và dấu chấm không đúng";
     return(false);
  } else if (userEmailText.lastIndexOf(".") > userEmailText.length-3) {
     feedback = "Domain không hợp lệ"
     return(false);
  }

  // quay trở lại
  feedback = "";
  return(true);
}
```

- Thế là kiểm tra dữ liệu đã xong. Bây giờ chúng ta sử dụng các hàm này để kiểm tra dữ liệu nhập vào và xuất ra kết quả. Hàm sau sẽ kiểm tra từng ô nhập liệu, nếu một trong những ô trên sai thì sẽ trả về kết quả false, còn đúng hết sẽ là true

CODE

```
function checkAll() {
  if (!checkUserName()) {
     return(false);
  } else if (!checkUserYear()) {
     return(false);
  } else if (!checkUserEmail()) {
     return(false);
  }

  return(true);
```

}

- Bây giờ chúng ta viết code cho nút Submit. Khi nhấn nút Submit thì sẽ gọi hàm CheckAll và đưa sang frame tiếp theo với lời cám ơn.

CODE

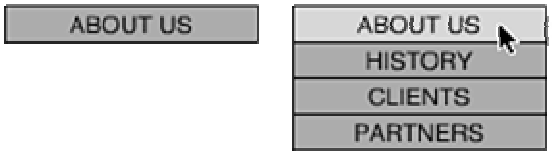
```
on (release) {
  if (checkAll()) {
     nextFrame();
  }
}
```

Giờ thứ 16: Menu và button động

Menu hệ thống đã rất quen thuộc với các chương trình máy tính hiện nay. Hệ điều hành của chúng ta và ngay cả Flash đều có một hệ thống menu ở trên đầu. Menu là một cách tốt để đưa ra nhiều lựa chọn cho người dùng mà lại ít tốn diện tích màn hình.
Trong giờ thứ 16 này, chúng ta sẽ học về:
- Cách làm một menu đơn giản
- Sử dụng menu trong movie
- Tạo một menu xổ xuống khi chúng ta click vào một nút
- Cách tạo button động
- Sử dụng button động trong movie

Cách làm một menu đơn giản

Những bạn mới làm quen với AS thường muốn biết cách tạo menu. Thật ra để tạo một menu rất đơn giản, chúng ta đã từng biết qua rồi, hoặc có thể các bạn không để ý. Cách thức hoạt động của nó là có một button, và khi người dùng đưa con trỏ qua button đó thì một loạt những button khác sẽ xuất hiện lần lượt bên dưới button đó tạo thành một dãy menu. Hình dưới đây là một ví dụ



Khi chúng ta đưa con trỏ ngang qua nút About Us thì một loạt những button khác sẽ xuất hiện như hình bên phải. Như vậy, chúng ta cần có 2 frame trong movie clip làm menu. Frame thứ nhất sẽ chỉ chứa button About Us, frame thứ hai sẽ chứa button About Us và 3 button còn lại. Ở frame thứ nhất, khi người dùng đưa con trỏ ngang qua button About Us thì sẽ nhảy sang frame thứ 2 và dừng lại ở frame 2 cho đến khi người dùng đưa con trỏ chuột ra ngoài, khi đó thì sẽ trở về frame 1. Nếu để ý, các bạn sẽ thấy cách này giống như chúng ta đã được học ở Giờ thứ 13 về RollOver. Chúng ta sẽ dùng hàm hitTest để kiểm tra xem vị trí của con trỏ chuột có nằm trong button không. Dưới đây là đoạn code xử lý việc này. Chúng ta cùng xem nhé!

CODE

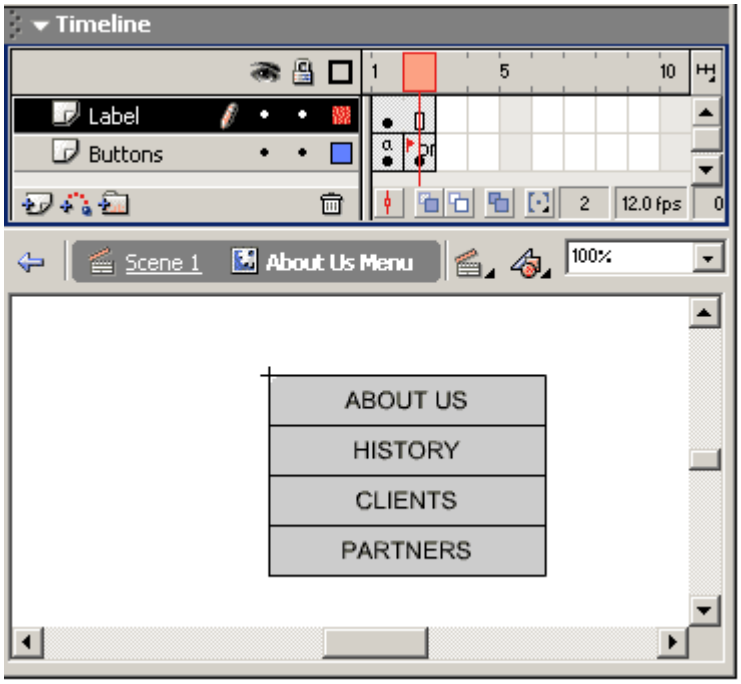
```
onClipEvent(load) {
  previouslyOver = false;
}

onClipEvent(enterFrame) {
  // kiểm tra vị trí con trỏ
  currentlyOver = this.hitTest(_root._xmouse,_root._ymouse,true);

  // kiểm tra sự thay đổi
  if (!previouslyOver and currentlyOver) {
     previouslyOver = true;
     this.gotoAndStop("on");
  }  else if (previouslyOver and !currentlyOver) {
     previouslyOver = false;
     this.gotoAndStop("off");
  }
}
```

Hãy chú ý, nếu bạn sử dụng AS để quản lý những thành phần khác đực đặt chung trong frame, hãy chắc rằng bạn phải đưa menu lên trên cùng bằng cách sử dụng hàm swapDepths

Luyện tập: Làm một menu

Nào, chúng ta bắt tay vào làm thử một menu đơn giản nhé! Menu chính của chúng ta sẽ có 3 phần: About Us, Products và Store, mỗi menu lại chứa nhiều menu con.
- Việc đầu tiên là tạo một movie mới trong Flash
- Tạo một button đơn giản thôi, button này không nên có chữ, và nhớ chừa chỗ trống để chúng ta đưa chữ vào sau
- Tạo movie clip mới, đặt tên là About Us Menu. Tạo hai layer, một là Label và một là Buttons
- Trong layer Buttons, kéo button vừa tạo vào. Đặt dòng chữ About Us lên trên
- Layer Label sẽ trải ra trên 2 frame. Nhưng 2 frame trong layer Buttons sẽ khác nhau, frame đầu đặt tên là off, frame 2 đặt tên là on.
- Trong frame thứ hai của layer Buttons, kéo thêm 3 button nữa vào và tạo nội dung cho chúng là History, Clients, và Partners. Nhớ đặt câu lệnh stop() vào frame đầu tiên.

Movie clip của chúng ta bây giờ sẽ giống như hình bên dưới



Quay trở lại level root, kéo movie clip About Us Menu từ Library vào, đặt tên là aboutUsMenu, và chèn đoạn code sau:

CODE

```
onClipEvent(load) {
  previouslyOver = FALSE;
}

onClipEvent(enterFrame) {
  currentlyOver = this.hitTest(_root._xmouse,_root._ymouse,true);

  if (!previouslyOver and currentlyOver) {
     previouslyOver = true;
     this.gotoAndStop("on");
  }  else if (previouslyOver and !currentlyOver) {
     previouslyOver = false;
     this.gotoAndStop("off");
  }
}
```

- Bây giờ hãy chạy thử movie của bạn xem nào. Đưa con trỏ vào button About Us xem điều gì xảy ra!
- Phần việc còn lại là của bạn đó. Làm tương tự cho các menu còn lại!

Tạo một menu xổ xuống khi chúng ta click vào một nút

Có nhiều cách để làm menu xổ xuống, và cũng có nhiều cách menu hoạt động. Chúng ta đã biết một cách trong ví dụ phần trước, khi người dùng đưa con trỏ ngang qua một button thì một loạt button khác sẽ xuất hiện dọc bên dưới tạo thành một menu, đơn giản chỉ bằng 2 frame.
Menu xổ xuống sẽ hoạt động theo một cách khác: khi người dùng click vào một

button, một loạt menu sẽ xuất hiện nhưng người dùng phải giữa chuột và kéo con trỏ để chọn các menu con, muốn chọn menu nào thì thả chuột tại menu đó. Chúng ta hãy nghiên cứu mổ xẻ menu xổ xuống này nhé!
Chúng ta cũng sẽ tạo 2 frame như bài trước, frame đầu chứa button là tiêu đề của menu, frame hai chứa các button xếp dọc xuống thành một hệ thống menu khi tiêu đề của menu được click. Tuy nhiên cách viết code sẽ khác đi!
Đây là code cho button làm tiêu đề cho menu

CODE
```
on (press) {
  expandMenu();
}

on (release, releaseOutside) {
  collapseMenu();
}
```

Khi người dùng click vào button thì nó sẽ gọi hàm expandMenu(), khi người dùng thả chuột ra thì nó sẽ gọi hàm collapseMenu()
Ngoài ra thì chúng ta còn sử dụng các event on(dragOver) và on(dragOut), hai event này cũng giống với on(rollOver) và on(rollOut) nhưng mà phải giữ chuột trong khi di chuyển

CODE
```
on (dragOver) {
  rollOverMenu();
}

on (dragOut) {
  rollOutMenu();
}
```

Button tiêu đề đã gọi 4 hàm expandMenu(), collapseMenu(), rollOverMenu(), rollOutMenu(), bây giờ chúng ta sẽ viết các hàm này, đặt chúng trên frame nhé! Hàm expandMenu() sẽ đặt giá trị cho biến expanded là true và nhảy sang frame thứ hai

CODE
```
function expandMenu() {
  expanded = true;
  gotoAndStop("on");
}
```

Hàm collapseMenu() sẽ làm ngược lại

CODE
```
function collapseMenu() {
  expanded = false;
```

```
  gotoAndStop("off");
}
```

Hàm rollOverMenu sẽ kiểm tra biến expanded và sẽ di chuyển đến frame thích hợp nếu expanded = true. Có nghĩa là khi người dùng click chuột vào button tiêu đề thì menu sẽ xổ xuống và người dùng phải giữ chuột trong lúc di chuyển để chọn, nếu thả chuột ra thì menu sẽ thu lại.

CODE
```
function rollOverMenu() {
  if (expanded) {
     gotoAndStop("on");
  }
}

function rollOutMenu() {
  if (expanded) {
     gotoAndStop("off");
  }
}
```

Chúng ta sẽ viết code tiếp cho các menu xổ xuống. Chúng đều là các button, và chúng ta sẽ viết event on(release) cho chúng để bắt sự kiện khi người người thả chuột trên button đó, có nghĩa là người dùng chọn menu đó. Khi đó, nó gọi hàm collapseMenu() rồi thực hiện công việc của mình, ở đây đơn giản chỉ gọi hàm trace. Chúng ta cũng viết event on(dragOver) và on(dragOut) cho các button này để giữ menu lại khi người dùng giữ chuột và kéo qua các button cũng như sẽ thu menu lại khi người dùng thả chuột ra hoặc kéo ra ngoài.

CODE
```
on (release) {
  collapseMenu();
  trace("History Button Pressed");
}

on (dragOut) {
  rollOutMenu();
}
```

Điều cuối cùng cần phải làm là phải thay đổi thuộc tính cho các button. Trong phần khung properties của button, chuyển Track as Button thành Track as Menu Item. Điều này sẽ làm cho button nhận được sự kiện release thay vì sẽ nhận press trước.
Còn có rất nhiều cách để làm menu, nó phụ thuộc vào mục đích sử dụng của bạn và khả năng sử dụng AS của mỗi người 

Button động

Một cách khác cũng tương tự để làm menu xổ xuống là sử dụng button động. Chúng ta có thể làm một menu xổ xuống mà không cần phải làm cách button trước, chúng ta sẽ được tự sinh ra bằng AS , thú vị nhỉ.
Điều đầu tiên cần phải làm là tạo một button mẫu. Tiếp theo, đặt button vào trong một movie clip, movie clip này sẽ có hai thành phần, một là button và hai là dynamic text ở trên button, dynamic text sẽ được liên kết với biến buttonLabel. Trong cửa sổ Library, click chuột phải lên tên movie clip và chọn Linkage. Nhớ chọn mục Export for Actionscript và đặt tên cho nó là buttonMovieClip. Được rồi, bây giờ chúng ta đã có một button mẫu, tiếp theo chúng ta sẽ sử dụng AS để sử dụng button này. Việc này cũng rất đơn giản, chúng ta sử dụng lệnh attachMovie để tạo một instance của movie clip và đặt lại giá trị cho dynamic text trong movie clip, và đặt lại vị trí của nó bằng cách đặt cách thuộc tính _x, _y.

CODE
```
function createButton(buttonLabel, x, y) {
  this.attachMovie("buttonMovieClip","button"+buttonLevels,buttonLevels);
  bmc = this["button"+buttonLevels];
  bmc.buttonLabel = buttonLabel;
  bmc._x = x;
  bmc._y = y;
  buttonLevels++;
  return(bmc);
}
```

Được rồi, hãy thử movie của bạn xem nào
Bạn có thể tạo ra hàng loạt button động bằng cách gọi một loạt hàm createButton, hoặc chúng ta sẽ lưu các tên button vào một mảng rồi dùng vòng lặp for để gọi hàm createButton.
Nhưng có một vấn đề cần giải quyết là làm thế nào để xử lý riêng cho từng button. Nếu viết code ngay trong button thì các button sẽ như nhau. Vậy làm cách nào để làm cho các button có thể xử lý những công việc khác nhau? Button sẽ gọi những hàm từ ngoài root, như vậy thì mỗi button có thể gọi một hàm khác nhau, điều này cũng có nghĩa là chúng sẽ thực hiện những việc khác nhau

Luyện tập: Sử dụng button động để tạo menu

- Tạo một movie mới trong Flash
- Tạo button mẫu như trong phần trước, đặt đoạn code sau vào button

CODE
```
on (rollOver) {
  _parent.buttonRolloverAction(thisAction,buttonLabel);
}

on (release) {
  _parent.buttonClickAction(thisAction,buttonLabel);
}
```

Điều này có nghĩa là khi button sẽ gọi hàm buttonRollOverAction khi đưa chuột qua, và gọi hàm buttonClickAction khi click chuột. Hai đối số của nó sẽ giúp báo button nào được click
- Dưới đây là hàm createButton để tạo button động, nhưng lần này chúng ta sẽ tạo một loạt button từ một mảng lưu sẵn

CODE
```
function createButton(buttonLabel, x, y, buttonAction) {

  this.attachMovie("buttonMovieClip","button"+buttonLevels,buttonLevels);
  bmc = this["button"+buttonLevels];
  bmc.buttonLabel = buttonLabel;
  bmc._x = x;
  bmc._y = y;
  bmc.thisAction = buttonAction;
  buttonLevels++;
  return(bmc);
}

// Tạo một loạt button từ mảng
function createButtonList(buttonList, x, y, direction) {
  for (var i=0;i<buttonList.length;i++) {
     ret = createButton(buttonList[i].label,x,y, buttonList[i].action);
     buttons[i].mc = ret;
     if (direction == "down") {
        y += 20;
     }  else if (direction == "across") {
        x += 100;
     }
  }
}
```

- Còn đây là cách tạo mảng để tạo button

CODE
```
mainButtonList = new Array();
mainButtonList.push({label:"About Us", action:"aboutUsButtonList"});
mainButtonList.push({label:"Products", action:"productsButtonList"});
mainButtonList.push({label:"Store",action:"storeButtonList"});
```

- Công việc tiếp theo là gọi hàm createButtonList để tạo button

CODE
```
buttonLevels = 1;
createButtonList(mainButtonList,100,100,"across");
```

- Nếu bạn thử chạy movie lúc nào thì chúng ta sẽ thấy 3 button được tạo nhưng mà sẽ chưa làm gì khi đưa chuột ngang qua hay click vào. Bây giờ chúng ta sẽ viết hàm buttonRollOverAction để xử lý

CODE
```
function buttonRolloverAction(thisAction,thisLabel) {
  if (thisAction == "aboutUsButtonList") {
     deleteAllButtonLists();
     createButtonList(aboutUsButtonList,100,120,"down");
  } else if (thisAction == "productsButtonList") {
     deleteAllButtonLists();
     createButtonList(productsButtonList,200,120,"down");
  } else if (thisAction == "storeButtonList") {
     deleteAllButtonLists();
     createButtonList(storeButtonList,300,120,"down");
  }
}
```

- Hàm buttonRollOverAction gọi hàm createButtonLists với các đối số khác nhau là một trong 3 mảng được định nghĩa dưới đây

CODE
```
aboutUsButtonList = new Array();
aboutUsButtonList.push({label:"History", action:"goto"});
aboutUsButtonList.push({label:"Clients", action:"goto"});
aboutUsButtonList.push({label:"Partners", action:"goto"});

productsButtonList = new Array();
productsButtonList.push({label:"Widgets", action:"goto"});
productsButtonList.push({label:"Toys", action:"goto"});
productsButtonList.push({label:"Power Tools", action:"goto"});

storeButtonList = new Array();
storeButtonList.push({label:"Order Online", action:"goto"});
storeButtonList.push({label:"Find a Store", action:"goto"});
storeButtonList.push({label:"Request Catalog", action:"goto"});
storeButtonList.push({label:"Track Shipment", action:"goto"});
storeButtonList.push({label:"Return Item", action:"goto"});
```

- Hàm deleteAllButtonLists sẽ làm biến mất các button đã được tạo, có nghĩa là tất cả các menu trong 3 mảng vừa tạo sẽ biến mất và sẽ chỉ xuất hiện một mảng tại một thời điểm mà thôi. Hãy tưởng tượng cái menu của chúng ta trong Flash, khi đưa con trỏ đến menu File thì menu File xổ xuống, nhưng khi đưa sang Edit thì menu File sẽ thu lại và menu Edit xổ xuống…
Trước đó, chúng ta phải có đoạn code sau để chỉ từng menu đến các mảng menu con

CODE

```
allButtonLists = new Array();
allButtonLists = [aboutUsButtonList,productsButtonList,storeButtonList];
```

Tiếp theo chúng ta sẽ viết hàm deleteButtonList và deleteAllButtonLists

CODE
```
function deleteButtonList(buttons) {
  for (var i=0;i<buttons.length;i++) {
    buttons[i].mc.removeMovieClip();
  }
}

function deleteAllButtonLists() {
  for(var i=0;i<allButtonLists.length;i++) {
    deleteButtonList(allButtonLists[i]);
  }
}
```

Bây giờ hãy chạy thử movie của chúng ta nhé. Bạn thấy sao? Tuyệt vời phải không nào 

Giờ thứ 17: Liên kết và liên lạc với trình duyệt, Browser Navigation and Communication

Khi thiết kế Flash, bạn có 2 sử lựa chọn, có thể nhúng vào một trang web hoặc làm một application có thể tự chạy riêng. Nếu bạn nhúng vào một trang web thì movie của bạn có thể liên lạc với trình duyệt để báo cho trình duyệt cần phải làm gì.

Trong giờ thứ 17, các bạn sẽ học được:
- Cách load một trang web
- Tìm hiểu cách liên lạc với JavaScript
- Mở một cửa sổ trình duyệt mới từ movie
- Sử dụng JavaScript gởi thông điệp đến movie
- Lưu thông tin người dùng vào JavaScript cookies
- Tạo movie sử dụng JavaScript
- Những câu lệnh đặc biệt cho những application tự chạy

Load một trang web

Ngày nay thì Flash được sử dụng rất nhiều trong các website. Nó được sử dụng để làm trang chủ hoặc là để tạo những thanh liên kết (navigation bar)… Cũng có lúc, chúng ta cần load một trang web mới từ movie Flash.

Cách đơn giản
Bạn có thể load một trang web mới bằng cách sử dụng câu lệnh getURL. Nó hoạt động giống như thẻ <a href…> của HTML. Dưới đây là một ví dụ khi nhấn một button thì sẽ load một trang web mới thay thế cho trang hiện tại:

CODE
```
on (release) {
  getURL("anotherpage.html");
}
```

Ở ví dụ trên thì trang anotherpage.html sẽ được load. Bạn có thể thay bằng một URL hoàn chỉnh (như là http://www.yahoo.com) để liên kết đến một website khác hoặc đường dẫn tương đối để liên kết đến những trang trong cùng một website.
Không cần sử dụng AS, chúng ta cũng có thể tạo được liên kết như vậy bằng cách đặt thuộc tính hypertext links của TextField, cái này thật ra giống hệt như là thẻ <a href…> của HTML

Cách nâng cao
Cách này bạn cũng sử dụng hàm getURL với một cách khác để xác định nơi sẽ load trang web lên là trong frame nào hoặc là trong window nào. Như chúng ta đã biết thì mỗi frame và mỗi window đều có tên, chúng ta sẽ truyền tên này vào đối số thứ 2 của hàm getURL.
Trong ví dụ dưới đây, trang web của bạn sẽ có nhiều frame, trong đó có frame tên là Main, trang web mới sẽ được load trong frame Main này

CODE
```
on (release) {
  getURL("summary.html","Main");
}
```

Bạn cũng có thể sử dụng đoạn code trên để load trang web vào window tên Main.
Ngoài ra, bạn còn có thể sử dụng những đối số đặc biệt để truyền vào thay cho tên:
- _blank : mở một window mới và load trang web vào window đó
- _parent : load trang web vào frame cha của frame hiện tại
- _top : load trang web vào window cũ, không kể đang ở frame nào mà sẽ thay thế tất cả các frame trong window

Nếu bạn muốn thay đổi những thiết lập của window như kích thước… thì bạn phải sử dụng JavaScript. Chúng ta sẽ nói về vấn đề này sau 

Luyện tập: Làm thanh liên kết (navigation bar)

Bây giờ thì bạn đã đủ khả năng làm một thanh liên kết bằng Flash sử dụng AS, nhưng bạn cần phải có thêm những kiến thức khác về HTML.
Thanh liên kết của chúng ta sẽ đặt ở frame bên trái của trình duyệt, frame bên phải sẽ chứa nội dung.
Movie làm bằng Flash sẽ chứa một số button để liên kết sang các trang web khác.
- Đầu tiên chúng ta sẽ tạo một trang HTML chứa 2 frame, trang này tên là

navigation.html

CODE
```
<HTML><HEAD>
<TITLE>Flash Navigation Example</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">

<FRAMESET cols="120,*">
<FRAME name="navigation" src="navbar.html" scrolling="no">
<FRAME name="content" src="content1.html" scrolling="auto">
</FRAMESET>

</BODY>
</HTML>
```

- Trang HTML trên chỉ tạo ra 2 frame, trong mỗi frame sẽ chứa một trang HTML khác. Bây giờ chúng ta sẽ tạo 2 trang HTML đó. Chúng ta chưa cần phải làm trang HTML cho frame bên trái, vì nó sẽ được tạo ra khi chúng ta publish movie thành HTML. Còn trang HTML trong frame bên phải sẽ có nhiều thay đổi.
Bây giờ hãy tạo 3 trang HTML đơn giản tên content1.html, content2.html, content3.html chứa 3 dòng chữ đơn giản.

CODE
```
<HTML><HEAD>
<TITLE>Content 1</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
Content 1
</BODY>
</HTML>
```

- Bây giờ là công việc làm với Flash. Tạo một movie mới rộng 100px, cao 400px. Tạo 3 button và viết code cho các nút để liên kết đến 3 trang content1.html, content2.html, content3.html tương tự như sau:

CODE
```
on (release) {
  getURL("content1.html","content");
}
```

- Lưu movie lại với tên navbar.fla
- Publish movie ra thành file html
- Tiếp theo là đưa tất cả các file vào một thư mục. Các file đó sẽ là: navigation.html, content1.html, content2.html, content3.html, navbar.html, and navbar.swf.
- Hãy mở trang navigation.html để thử xem nhé 

ActionScript và JavaScript

Nếu bạn đã quen sử dụng JavaScript rồi thì bạn sẽ rất vui khi biết rằng JavaScript có thể liên lạc được với ActionScript. Tuy nhiên, cách này không hoạt động tốt đối với tất cả các loại trình duyệt.
Việc liên lạc này được xây dựng trên 2 công nghệ. Một là công nghệ LiveConnect được xây dựng trong những phiên bản trình duyệt Netscape trước phiên bản 6.0. Công nghệ thứ hai là ActiveX dùng để liên lạc giữa Flash và Internet Explorer.
Nhưng nếu bạn đang thiết kế cho người dùng sử dụng trình duyệt của Windows thì cách này sẽ rất tốt.

Gởi thông điệp đến JavaScript

Gởi thông điệp từ ActionScript đến JavaScript thì chỉ cần viết code trong ActionScript nhưng nó cũng sẽ thay đổi nội dung trang HTML của bạn .
Nếu bạn tạo file Flash bằng cách chọn FSCommand trong Publish settings, bạn sẽ tạo được một file HTML đã được sửa chữa đầy đủ để nhận thông điệp. Công việc của bạn chỉ là thay thế những chỗ được đánh dấu Your code here bằng phần JavaScript của bạn.
Để hiểu rõ hơn cách nó làm việc như thế nào thì chúng ta hãy cùng nhau mổ xẻ file html mà Flash đã tạo ra nhé
Đầu tiên, hãy chú ý đến thẻ OBJECT/EMBED, trong đó sẽ có một số phần để chấp nhận sự liên lạc. Tham số ID trong thẻ OBJECT sẽ giống với tham số NAME trong thẻ EMBED. Hai tham số này sẽ đặt tên cho movie của chúng ta trong trang web để JavaScript có thể gọi nó. Ngoài ra còn có một tham số khác trong thẻ EMBED, đó là tham số swLiveConnect để cho phép những phiên bạn Netscape trước phiên bản 6.0 có thể liên lạc với Flash bằng công nghệ LiveConnect.

CODE

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/
[ic:cc]flash/swflash.cab#version=5,0,0,0"
ID=flashmovie WIDTH=120 HEIGHT=120>
<PARAM NAME=movie VALUE="17astojs.swf"> <PARAM NAME=quality
VALUE=high> <PARAM
NAME=bgcolor VALUE=#FFFFFF>     <EMBED src="17astojs.swf" quality=high
bgcolor=#FFFFFF
WIDTH=120 HEIGHT=120
swLiveConnect=true NAME=flashmovie TYPE="application/x-shockwave-flash"
PLUGINSPAGE="http:
//www.macromedia.com/shockwave/download/
index.cgi?P1_Prod_Version=ShockwaveFlash"></
EMBED>
</OBJECT>
```

Trước đoạn code của thẻ OBJECT/EMBED sẽ có một đoạn script. Phần đầu tiên là một hàm JavaScript với tên của movie ID trong thẻ OBJECT và nối tiếp với _DoFSCommand. Như ví dụ ở trên thì tên hàm của chúng ta sẽ là

flashmovie_DoFSCommand.
Trong hàm này thì chúng ta sẽ truyền dữ liệu lại cho movie. Không may là Netscape và Internet Explorer nhìn nhận movie của chúng ta khác nhau  . Internet Explorer nhìn movie với tên truyền vào tham số ID (flashmovie), còn Netscape nhìn movie với tên là document.flashmovie. Hãy xem đoạn code dưới đây xem nhé:

CODE
```
<script LANGUAGE=JavaScript>
function flashmovie_DoFSCommand(command, args) {
  if (navigator.appName.indexOf("Microsoft") != -1) {
     var flashmovieObj = flashmovie;
  }  else {
     var flashmovieObj = document.flashmovie;
  }

  alert(command);
  alert(args);
}
```

Trong Internet Explorer, Flash không chỉ có thể liên lạc với JavaScript mà còn có thể liện lạc với VBScript. Đoạn code dưới đây được viết bằng JavaScript thay cho đoạn JavaScript trên:

CODE
```
if (navigator.appName && navigator.appName.indexOf("Microsoft") != -1 &&
  navigator.userAgent.indexOf("Windows") != -1 &&
  navigator.userAgent.indexOf("Windows 3.1") == -1) {
     document.write('<script LANGUAGE=VBScript\> \n');
     document.write('on error resume next \n');
     document.write('Sub flashmovie_FSCommand(ByVal command, ByVal args)\n');
     document.write('call flashmovie_DoFSCommand(command, args)\n');
     document.write('end sub\n');
     document.write('</SCRIPT\> \n');
}
</SCRIPT>
```

Vậy còn phần ActionScript phải làm sao đây. Xin trả lời là chỉ cần một dòng duy nhất mà thôi 

CODE
```
fscommand ("alert", "This is alert 1.");
```

Nhận thông điệp từ JavaScript

Nhận thông điệp từ JavaScript thì có vẻ dễ hơn. Nhưng bạn phải nhớ rằng đã đặt tham số ID trong thẻ OBJECT và tham số NAME trong thẻ EMBED, nhớ đặt giống tên nhé. Và cũng nhớ đặt swLiveConnect=true trong thẻ EMBED.
Bây giờ thì đã sẵn sàng để truyền thông điệp từ JavaScript cho movie rồi. Hãy xem ví dụ này nhé, trong ví dụ này thì chúng ta sẽ sử dụng hàm gotoFrame để di chuyển movie sang frame thứ hai.

CODE
```
<FORM NAME="flashControlForm">
<INPUT NAME="gotoFrame1" TYPE=Button VALUE="Frame 1"
onClick="window.document.flashmovie.GotoFrame(1);">
</FORM>
```

Chắc chắc đến đây các bạn đang tự hỏi tại sao di chuyển sang frame thứ hai mà lại sử dụng gotoFrame(1), đúng không nào? Bởi vì hệ đếm của chúng ta bắt đầu từ số 0 (zero-based). Vì vậy, frame 1 sẽ là 0, frame 2 sẽ là 1, frame 3 sẽ là 2…
Có hơn 24 câu lệnh trong Flash movie. Tuy nhiên chúng ta không cần phải dành nhiều thời gian để tìm hiểu bởi vì như đã nói ở trên thì cách liên lạc này không phải hoạt động tốt đối với tất cả các loại trình duyệt.
Bạn có thể sử dụng hàm GetVariable và SetVariable để điều khiển những biến trong timeline của movie. Câu lệnh Zoom dùng để kéo dãn movie. Hai hàm isPlaying và percentLoaded dùng để kiểm tra movie nào đang hoạt động. Play dùng để play một movie khi nó đang ngừng.

Luyện tập: Mở window mới

Có lẽ đây là một yêu cầu rất thường gặp đối với những người sử dụng Flash. Bạn có thể làm được điều này bằng cách sử dụng hàm getURL hoặc JavaScript.
Tuy nhiên, sử dụng JavaScript sẽ có rất nhiều đặc điểm mới và hay hơn, bạn có thể đặt lại những thuộc tính của window. Nào, cùng làm thử nhé:

- Tạo một movie mới.
- Đặt vào movie một button.
- Chèn đoạn code sau cho button vừa tạo

CODE
```
on (release) {
 fscommand ("newwindow", "content.html");
}
```

- Trong phần Publish Settings, chọn để publish ra file HTML cùng với Flash movie.

Trong phần HTML, nhớ chọn vào mục Flash with FSCommand
- Publish movie của bạn.
- Mở file HTML mà Flash vừa tạo bằng một trình soạn thảo nào cũng được, như NotePad chẳng hạn. Hãy tìm phần để chèn JavaScript, và chèn đoạn code sau vào:

CODE
```
if (command == "newwindow") {
   window.open(args,"","width=320,height=240,location=no,toolbar=no,
menubar=no");
}
```

- Tạo thêm một file HTML đơn giản nữa và đặt tên là content.html
- Mở file HTML của bạn ra trong một trình duyệt và nhớ là trình duyệt đó phải hỗ trợ JavaScript đó nha. Khi nhấn vào button thì một cửa sổ mới sẽ xuất hiện, không có toolbar, kích thước 320x240.
- Vậy còn đối với những trình duyệt không hỗ trợ JavaScript thì sao? Đơn giản thôi, sử dụng getURL  . Điều quan trọng là cần phải cho Flash biết lúc nào sử dụng Flash, lúc nào sử dụng JavaScript.
Quay trở lại file HTML mà Flash đã tạo, chèn đoạn code JavaScript sau vào cuối đoạn code ta đã chèn vào lúc trước

CODE
```
function initComm() {
   window.document.newWindowMovie.SetVariable("jsCommOK","OK");
}
```

Đoạn code trên sẽ thử đặt giá trị cho biến jsCommOK là OK. Nếu trình duyệt có hỗ trợ JavaScript thì jsCommOK sẽ mang giá trị OK còn nếu không thì biến jsCommOK vẫn là undefined.
- Để chạy hàm initComm đầu tiên thì bạn sửa lại phần BODY của trang HTML như sau:

CODE
```
<BODY bgcolor="#FFFFFF" onLoad="initComm();">
```

Điều này có nghĩa là hàm initComm sẽ được gọi ngay sau khi trang này load xong.
- Bây giờ hãy quay trở lại file Flash của bạn, sửa đoạn script của button thành:

CODE

```
on (release) {
  if (jsCommOK == "OK") {
     fscommand ("newwindow", "content.html");
  }  else {
     getURL ("content.html", "_blank");
  }
}
```

Luyện tập: Tạo một SlideShow được điều khiển bởi JavaScript

Trong phần này, chúng ta sẽ cùng nhau làm một file Flash không có ActionScript mà sẽ được điều khiển bằng JavaScript.
- Tạo một movie Flash mới với 3 frame hoặc nhiều hơn. Nội dung của mỗi frame thì không quan trọng, bạn muốn để gì trong đó cũng đựoc nhưng một lời khuyên là mỗi frame nên khác nhau để chúng ta theo dõi sự thay đổi
- Đặt câu lệnh stop() vào frame đầu tiên
- Publish ra file HTML
- Mở file HTML đó ra trong trình soạn thảo văn bản
- Truyền tham số ID trong thẻ OBJECT và NAME trong thẻ EMBED, cả hai sẽ mang giá trị là slideshow
- Nhớ đặt swLiveConnect=true trong thẻ EMBED
- Tạo 2 button trong file HTML bằng đoạn code sau:

CODE

```
<FORM NAME="flashControlForm">
<INPUT NAME="next" TYPE=Button VALUE="Next" onClick="nextFrame();">
<INPUT NAME="prev" TYPE=Button VALUE="Previous"
onClick="prevFrame();">
</FORM>
```

- Mỗi button sẽ gọi một hàm. Cả hai hàm sẽ sử dụng TcurrentFame(“/”)biết đang ở frame thứ mấy, rồi sử dụng gotoFrame để di chuyển tới hoặc lui.

CODE

```
<script LANGUAGE="JavaScript">
function nextFrame() {
 var frameNum = window.document.slideshow.TCurrentFrame("/");
 window.document.slideshow.GotoFrame(frameNum+1);
}
function prevFrame() {
 var frameNum = window.document.slideshow.TCurrentFrame("/");
 window.document.slideshow.GotoFrame(frameNum-1);
}
```

</SCRIPT>

Công việc tiếp theo bạn còn nhớ không: CHẠY THỬ đi nào

Giờ thứ 18: Gởi thông tin cho máy chủ, Sending Information to the Server

Cho đến tận bây giờ, những movie chúng ta làm được hầu hết là chỉ chạy một mình (stand alone). Có nghĩa là nó chỉ chạy ở một máy khách (client-side) không có sự liên lạc với máy chủ (Server). Nên nhớ rằng Flash cũng có thể gởi trả thông tin cho Server giống như form của HTML. Vậy thì trong giờ thứ 18 này, chúng ta sẽ tìm hiểu về vấn đề này.

Trong giờ thứ 18, chúng ta sẽ học:
- Tìm hiểu về đối tượng LoadVars
- Tạo một chương trình server-side đơn giản
- Sử dụng Flash để gởi dữ liệu về cho Server

Đối tượng LoadVars

Rất may mắn là chúng ta đã có Flash MX, bởi vì trong những phiên bản Flash trước đây, muốn làm điều này thì chúng ta phải làm những movie clip rất khó khăn. Đối với Flash MX, chúng ta có thể sử dụng đối tượng LoadVars.
Đối tượng LoadVars bao gồm một tập hợp các câu lệnh và những biến đặc biệt để gởi dữ liệu cho Server giống như post form trong HTML. Chúng ta có thể tạo đối tượng giống như tạo những đối tượng khác. Hãy xem giờ thứ 12 nhé!

Lấy dữ liệu

Dưới đây là một ví dụ. Thay vì sử dụng new Object() bằng new LoadVars. Sau đó thì đối tượng LoadVars mới sẽ được tạo.

CODE
```
myVars = new LoadVars();
```

Với đối tượng LoadVars, chúng ta có thể làm được 2 việc, đó là gởi và lấy dữ liệu. Để lấy dữ liệu, chúng ta sử dụng câu lệnh load. Cái chúng ta cần ở đây là một đường dẫn đến nơi chứa dữ liệu:

CODE
```
myVars.load("myURL.txt");
```

Ví dụ trên hoạt động giống như câu lệnh LoadVariables trong giờ thứ 10. Tuy nhiên, câu lệnh LoadVariables chỉ lấy và thay thế dữ liệu trong cùng một level với câu lệnh này mà thôi. Hãy tưởng tượng, chúng ta có một file chứa dữ liệu như sau:

CODE
name=George&ID=47

Gởi dữ liệu

Với đối tượng LoadVars, chúng ta cũng có thể gởi dữ liệu đi lên Server. Trước hết, chúng ta đưa dữ liệu vào đối tượng LoadVars, rồi sau đó sử dụng câu lệnh send để gởi dữ liệu đi.

CODE
```
myVars = new LoadVars();
myVars.name = "George";
myVars.ID = 47;
myVars.send("serverprogram.cgi", "_self");
```

Đoạn code trên sẽ tạo một đối tượng LoadVars mới tên là myVars, rồi đưa hai thuộc tính vào đối tượng myVars. Sau đó, gởi dữ liệu đi lên Server, chính xác là đến chương trình CGI tên là echo.cgi.
Câu lệnh send cũng hoạt động tương tự như trong form của HTML. _self là Target, chúng ta có thể thay thế bằng các giá trị khác như đã được biết.
Nhưng còn một câu lệnh nữa là sendAndLoad. Câu lệnh này được ghép câu lệnh send và load. Có nghĩa là đối tượng LoadVars sẽ gởi dữ liệu lên Server, sau đó sẽ lấy giá trị trả về.

CODE
```
mySendVars = new LoadVars();
myLoadVars = new LoadVars();
mySendVars.name = "George";
mySendVars.ID = 47;
mySendVars.sendAndLoad("serverprogram.cgi", myLoadVars);
```

Trạng thái lấy dữ liệu

Hãy nhớ là câu lệnh send và load sẽ không lấy dữ liệu ngay tức thì. Có thể phải đợi trong một thời gian ngắn hoặc dài. Vì vậy, chúng ta sẽ có nhu cầu theo dõi trạng thái

lấy dữ liệu. Chúng ta không thể sử dụng dữ liệu ngay lập tức sau câu lệnh load hoặc sendAndLoad, mà cần phải kiểm tra xem đã lấy dữ liệu xong chưa.
Cách đơn giản nhất là dùng một movie clip lặp đi lặp lại để kiểm tra. Cũng có thể sử dụng getBytesLoaded và getBytesTotal với những dữ liệu lớn. Dưới đây là ví dụ để kiểm tra việc lấy dữ liệu:

CODE
```
myLoadVars.onLoad = function(success) {
if (success) {
 gotoAndStop("load done");
}  else {
 gotoAndStop("load problem");
}
}
```

Như trong đoạn code trên thì đối số success sẽ nhận một trong hai giá trị true hoặc false để cho biết việc nhận dữ liệu đã xong hay chưa

Giờ thứ 19: Dùng XML với Flash

Càng ngày XML càng trở nên thông dụng trong các ứng dụng tin học, và ngay cả trong Flash, vậy trong giờ này mình sẽ học các điều sau:

- XML căn bản
- Đối tượng XML trong Flash
- Phân tích và xử lý XML theo phương pháo đệ quy (recursive)

XML Căn bản

XML chỉ đơn giản là một văn bản để chứa dữ liệu. XML tương tự như HTML, cũng dùng các thẻ. Tuy nhiên, XML khác ở HTML là các thẻ HTML đã được ấn định trước cho các chức năng khác nhau, còn XML thì không.

Với XML, bạn có thể tự tạo cho mình các thẻ theo ý bạn để phù hợp cho mục đích riêng của bạn.

Có thể tạo một XML file với 1 trình biên tập văn bản (như Notepad, Textpad ...) đơn giản hay là các software chuyên để viết về XML (XMLSpy, Epic ...)

Với Flash MX, bạn có thể dễ dàng truy cập được dữ liệu trữ ở trong XML, và đối tượng XML của Flash sẽ tự động phân tích văn bản XML này.

Trong VNFX có rất nhiều bài viết về XML (ví dụ như bài giới thiệu XML của Flash-Lee) các bạn có thể tham khảo thêm

Đối tượng XML

Đối tượng XML trong Flash gồm có nhiều hàm vầ đặc tính dùng để giúp bạn lấy và phân tích dữ kiện trong XML file một cách dễ dàng. Bước đầu tiên khi dùng đối tượng XML là tạo một phiên bản XML trước:

CODE
```
myXML = new XML()
```

Phân tích và sử lý văn bản thành XML

Sau khi tạo phiên bản XML trên, hiện giờ phiên bản này vẫn chưa có gì cả, tuy nhiên bạn có thể nhanh chóng tạo ra một tài liệu XML bằng cách dùng lệnh **parseXML**. Lệnh này sẽ nhập một chuỗi văn bản, và phân tích và xử lý nó thành 1 tài liệu XML.

CODE
```
myXML = new XML();
myXML.parseXML("<user><name>Gary</name><ID>47</ID></user>");
```

Hay bạn có thể viết trực tiếp như sau

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
```

Nếu mà chuỗi được nhập vào không thể tạo thành XML hoàn chỉnh thì bạn có thể dùng đặc tính **status** để kiểm tra.

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</user>");
trace(myXML.status);
```

Output window sẽ cho ra **-9** khi chạy đoạn code trên. **-9** có nghĩa là thiếu thẻ đóng (end tag), vì mình thiếu thẻ **</ID>**. Nếu output window cho ra **-10** thì bạn thiếu thẻ mở (start tag), và **0** có nghĩa là mọi việc hoàn chỉnh

Lấy dữ liệu từ đối tượng XML

Có nhiều hàm để làm việc này. Ví dụ, bạn có thể dùng **firstChild** để lấy nút (node) đầu tiên của đối tượng XML:

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.firstChild);
```

Output window sẽ cho ra **<user><name>Gary</name><ID>47</ID></user>**. Vì đây chính là nút đầu tiên của đối tượng XML. Nếu chúng ta đi xâu vào 1 lần nữa với **firstChild**

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.firstChild.firstChild);
```

Kỳ này output window sẽ cho ra **<name>Gary</name>**, vì phần tử **name** là nút đầu tiên của **user**
Ngoài cách trên ra, còn 1 cách nữa là dùng **childNodes**, là ma trận của của các nút. Cùng một ví dụ trên nhưng có thể dùng như sau với **chidlNodes**

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.childNodes[0].childNodes[0]);
```

Dưới nút **name**, chúng ta còn một nút nữa dưới nút **name**, và có thể dùng cách trên để lấy nút đó:

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.childNodes[0].childNodes[0].childNodes[0]);
```

và output window sẽ cho ra chữ **Gary**. Đây là nút văn bản (text node) của nút **name**. Nhìn thì tưởng đây là tận cùng rồi, không thể xuống sâu hơn được nữa, vì đây là nút cuối cùng, nhưng nếu bạn muốn lấy giá trị của nó như là chuỗi văn bản thì bạn có thể đi thêm 1 bước nữa như sau:

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.childNodes[0].childNodes[0].childNodes[0].nodeValue);
```

Nếu bạn muốn lấy cái **ID** ở trong nút thứ 2 thì thay đổi như sau:

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
trace(myXML.childNodes[0].childNodes[1].childNodes[0].nodeValue);
```

**** Để giải thích thêm về đoạn code trên, nếu bạn chỉ dùng childNodes thì kết quả sẻ là 1 đối tượng, còn nếu bạn dùng nodeValue thì sẽ ra chuỗi. Bạn có thể thử bằng đoạn code mình viết dưới đây***

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</ID></user>");
myVar1 = myXML.childNodes[0].childNodes[0].childNodes[0];
myVar2 = myXML.childNodes[0].childNodes[0].childNodes[0].nodeValue;
trace ("myVar1: " +typeof(myVar1));
trace ("myVar2: " +typeof(myVar2));
```

và output window sẽ cho bạn thấy rõ, cái nào law đối tượng và cái nào là chuỗi.

Tự tạo XML từ "tay trắng"

Không biết dịch câu này sao cho đúng nghĩa với "Creating XML from Scratch". đành dịch như trên vậy.

Các cách tạo XML ở trên đều từ 1 chuỗi văn bản, nhưng nếu như bạn không có 1 chuỗi văn bản nào trước thì sao? Bạn có thể dùng **createElement** để tạo ra các nút mới và dùng **createTextNode** để tạo ra các text node. Tuy nhiên 2 lệnh trên chỉ tạo ra nút chứ không thêm vào trong XML, vậy bạn phải thêm nó vào bằng lệnh **appendChild**.

Để minh hoạ, chúng ta sẽ tạo một XML tương tự như trên nhưng với các lệnh vừa đề cập.

CODE
```
myXML = new XML();
newElement = myXML.createElement("user");
myXML.appendChild(newElement);
newElement = myXML.createElement("name");
myXML.childNodes[0].appendChild(newElement);
newText = myXML.createTextNode("Gary");
myXML.childNodes[0].childNodes[0].appendChild(newText);
newElement = myXML.createElement("ID");
myXML.childNodes[0].appendChild(newElement);
newText = myXML.createTextNode("47");
myXML.childNodes[0].childNodes[1].appendChild(newText);
```

Nếu bạn muốn thay đổi giá trị của text node, thì bạn chỉ cần dùng **nodeValue**

CODE
```
myXML.childNodes[0].childNodes[1].childNodes[0].nodeValue = 53;
```

Thuộc tính

Các thành phần của XML có thể có thuộc tính (**attribute**). ***Trong đây mình gặp rắc rối về cách dịch attribute và property, cả 2 chữ này tiếng việt đều dịch là thuộc tính, đặc tính ... mà tiếng anh thì khác thật rắc rối vậy mình xin được dùng tiếng Anh nhe.*** Attribute gồm có từ khoá và giá trị của từ khoá đó, và dùng để định rõ một thành phần hơn. Ví dụ, đoạn XML dưới dây với attribute **type** làm rõ nghĩa của thành phần **name** hơn, ("alias" là bí danh)

CODE
```
<user>
  <name type="alias">Gary</name>
  <ID>47</ID>
</user>
```

Nếu bạn muốn đưa đoạn code trên vào trong XML object thì phải đổi dấu " (ngoặc kép) thành dấu ' (ngoặc đơn)

CODE
```
trace(myXML.childNodes[0].childNodes[0].attributes.type);
```

Còn một cách nữa cũng cho ra kết quả như trên là dùng **[]**

CODE
```
trace(myXML.childNodes[0].childNodes[0].attributes["type"]);
```

Và bạn có thể thay đổi giá trị của attribute, hay thêm attribute mới như sau:

CODE
```
myXML.childNodes[0].childNodes[0].attributes["type"] = "real";
```

Với câu trên, nếu attribute **type** chưa có thì nó sẽ được tạo

(trong sách các đoạn code trên tác giả viết lộn giữa type và alias, nếu bạn so sánh giữa sách và bài này thì sẽ thấy sự khác biệt, và có thể các source file cũng sẽ không chính xác, vậy các bạn cần kiểm tra lại)

Khác với node, attribute không thể truy cập bằng ma trận (array) nên bạn không thể dùng các lệnh như **length** hay dùng **[]** với index number. Nhưng bạn có thể dùng vòng lặp **for ... in** để truy cập từng attribute của node.

CODE
```
myXML = new XML("<user><name type='alias' validity='verified'>Gary</name>
<ID>47</user>");
for(attr in myXML.childNodes[0].childNodes[0].attributes) {
  trace(attr+": "+myXML.childNodes[0].childNodes[0].attributes[attr]);
}
```

output window sẽ cho ra: **alias** và **verified**

Thêm vài AS về XML

Bạn cũng cần biết thêm một số điều về đối tượng XML. Nhất là điểm sau, bất cứ một phần nào của đối tượng XML cũng có thể là một đối tượng XML riêng biệt. Ví dụ bạn có thể lấy cái **node** đầu tiên của đối tượng XML và quy thành một biến

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</user>");
thisUser = myXML.childNodes[0];
thisUserName = thisUser.childNodes[0];
thisUserNameText = thisUserName.childNodes[0].nodeValue;
thisUserID = thisUser.childNodes[1];
thisUserIDText = thisUserID.childNodes[0].nodeValue;
```

Bạn có thể biết được số **node** ở trong ở trong một **node** khác bằng thuộc tính **length** của **childNodes**. Ví dụ, **user** node có 2 node ở trong, vậy bạn có thể biết được bằng với đoạn code sau:

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</user>");
trace(myXML.childNodes[0].childNodes.length);
```

Ngoài việc có tểh tìm được giá trị của một node văn bản bằng **nodeValue**, bạn có thể biết được tên của thành phần (hay thẻ) với **nodeName**. Ví dụ bạn có thể lấy được tên của thành phần (thẻ) đầu tiên của **user** là **name** như sau:

CODE
```
myXML = new XML("<user><name>Gary</name><ID>47</user>");
trace(myXML.childNodes[0].childNodes[0].nodeName);
```

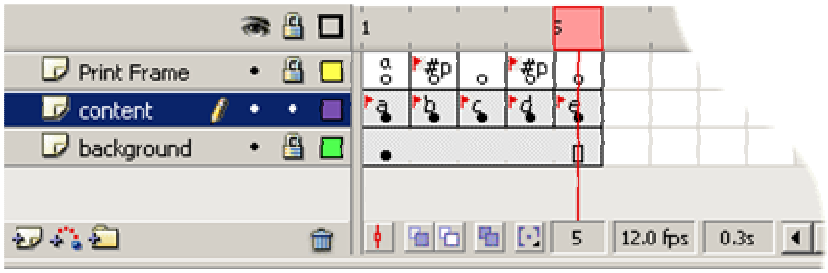
Bạn có thể biết được một node là thành phần của XML hay là một node văn bản (text node) bằng với thuộc tính **nodeType**. Nếu **nodeType** là 1 thì có là 1 thành phần của XML và có thể có nhiều thành phần ở trong node đó, nếu là 3 thì node đó chính là node văn bản.

Giờ thứ 20 : Printing

Không phức tạp như trình duyệt hoặc Server Cominucation, in ấn là một cách mà FLash truyền dữ liệu ra ngoài.
Khả năng in ấn của FLash là rất hữu ích vì nó cho phép chúng ta xây dựng những văn bản mà người sử dụng có thể in ra nội dung từ file flash của bạn.Đây thường là lựa chọn tốt hơn dựa vào hàm in ấn của trình duyệt .
Trong giờ thứ 20 này chúng ta sẽ :
- Học cách thiết lập movie của bạn cho việc in ấn
- Học cách sử dụng các lệnh in
- Tạo ra một biểu mẫu có thể in được

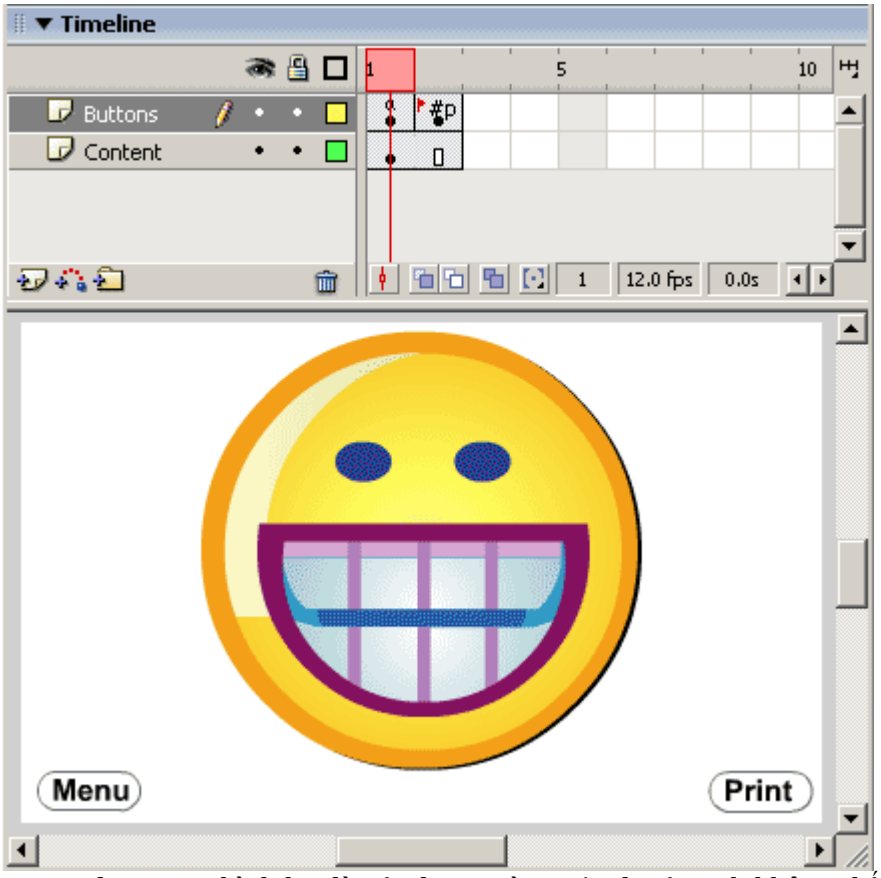
I> Thiết lập cho movie có khả năng in:

1-Bạn cần làm một vài việc với movie trước khi có thể sử dụng được các lệnh in của Action Script. Thật không may,những lệnh in này không được linh hoạt lắm. Nếu sử dụng chúng mà không có sự chuẩn bị , Flash sẽ in ra toàn bộ nội dung của movie, từng frame một.
Và đó thường là điều mà bạn không muốn. Có khi bạn chỉ muốn in duy nhất frame hiện tại, hoặc một frame bất kỳ trong một movie riêng, hoặc một đoạn gồm nhiều frame. Nhưng hiếm khi bạn muốn in hết toàn bộ.
2 - Đặt tên cho frame
Bạn sẽ chỉ định những frame được in bằng ký hiệu "#p" . Nếu ký hiệu đó không được đặt lên frame nào thì Flash sẽ in ra toàn bộ movie của bạn. Ngược lại, nếu bạn đặt từ hai ký hiệu trở lên trong movie của bạn thì flash sẽ in ra tất cả những frame đó.
Hình 1 cho ta thấy timeline với hai frame đã được đặt ký hiệu in. Chú ý rằng, tất cả những frame có chưa ký hiệu in này sẽ được đặt ở một layer riêng. Theo cách đó, bạn có thể đặt được nhiều ký hiệu trên nhiều frame. Và nó cũng thuận tiện hơn khi bạn không muốn sử dụng ký hiệu "#p" như một label thật cho keyframe.



*Chú ý: Khi bạn đặt từ hai ký hiệu in trở lên, khi test movie, Flash sẽ xuất hiện lời cảnh báo : "WARNING : Duplicate layer.."
Đó là điều rủi ro có thế xảy ra, nhưng cũng sẽ không ảnh hưởng đến movie của bạn trừ phi bạn sử dụng label "#p" trong câu lệnh "gotoAndStop" hoặc những lệnh tương tự....

3- Lên kế hoạch in
Bạn cần phải suy nghĩ và lên kế hoạch cho movie của bạn khi muốn movie có khả năng in. Vì không thể chỉ in một frame hiện tại nên bạn
cần tạo ra những frame có khả năng in. Chú ý rằng mọi thứ tồn tại trên movie của bạn sẽ được in ra. Nó bao gồm cả nút Print (để khi nhấn vào sẽ thực hiện lệnh in), nếu trong movie
có tồn tại.
Vì vậy chúng ta cần phải có một frame chứa nút Print, và một frame khác tương tự nhưng không có nút Print cũng như các yếu tố khôgn
cần thiết khác. Frame thứ hai này sẽ được dặt nhãn là "#p"
Bạn có thể bố trí bằng cách sử dụng timeline chính. Ví dụ bạn có một menu và một nút Prin trên hai layer riêng biệt, mà cần có
trong nội dung của biểu mẫu in. Những layer này nếu bạn không muốn in ra thì không cần kéo chũng sang frame có nhãn "#p"
hai hình sau thể hiện điều này. Hình thứ nhất,là frame mà người dùng sẽ thấy xuất hiện trên trang web. NÓ chứa nội dung. nút Print và menu



Ngược lại, trong hình hai là nội dung mà người duyệt web không thấy được, ít nhất là trên màn hình. Frame này được đặt tên là "#p"
bạn có thể thấy trên timeline.Layer Buttons không sử dụng keyframe giống như ở frame 1. Ngoài việc chặn nút Print không đươc in ra,
có thể thêm vào frame này một số thông tin sẽ được in ra như địa chỉ, ...

Bây giờ bạn đã biết làm thế nào để chuẩn bị cho một biểu mẫu in, bây giờ cùng học AS cần thiết

II> Các câu lệnh

Có hai lệnh in chính được sử dụng. Chúng có một khác biệt rất nhỏ, nhưng về căn bản là hoạt động giống nhau.
1- Print
Lệnh thứ nhất là

QUOTE
Print

Lệnh này bao gồm hai tham số. Tham số thứ nhất là đối tượng để in. Thương là timeline chính, hoặc "_root". Bạn cũng có thể sử dụng
"this". Tuy nhiên, nếu bạn không muốn in những frame trong movie, bạn có thể sử dụng tham chiếu tới movie đó.
Tham số thứ hai là một trong ba tuỳ chọn sau :

QUOTE
bframe

QUOTE
bmovie

QUOTE
bmax

Các tham số này giúp FLash có thể co dãn văn bản in theo ý của người sử dụng. Flash có thể co dãn văn bản in tới kích thước
của trang giấy mà không bóp méo văn bản. Ví dụ nếu frame được in có kích thước 550x400, thì chiều ngang sẽ được phóng to tới kích thước 550px
chiều dọc có thể sẽ được scale theo tỷ lệ.
Khi sử dụng tuỳ chọn "bframe", từng frame sẽ tự scale để lấp đầy kích thước của trang đó. Nếu frame thứ nhất có nội dung với kích thước là 550x400
nhưng frame thứ hai chỉ chứa nội dung có kích thước 275x200, khi đó frame thứ hai sẽ tự động scale gấp đôi kích thước ban đầu.
khi sử dụng tuỳ chọn "bmax", Flash sẽ kiểm tra toàn bộ các frame được in để xác định xem frame nào có kích thước lớn nhất. Các frame
còn lại sẽ scale dựa trên kích thước của frame lớn nhất, tính theo tỷ lệ. Điều này taọ nên các frame có kích thước tỷ lệ với nhau
Ví dụ, frame lớn nhất có kích thước 550x400, và nó lấp đầy trang in. Frame khác chỉ có kích thước 275x200, nó chỉ chiếm một nửa trang in.
Tuỳ chọn cuối cùng là "bmovie", trường hợp này bạn cần phải làm thêmm một việc nhỏ nữa, đó là tạo ra frame mới có chưa một khung.
Khung này sẽ xác định kích thước lớn nhất được in ra đối với văn bản (bằng kích thứoc của khung). Bạn phải đặt tên frame này với ký hiệu
"#b". Và flash sẽ sử dụng kích thước của khung để scale toàn bộ các frame còn lại.
Nếu có một phần văn bản này nằmg ngoài khung in
chúng sẽ không được in ra.
Sau đây là ví dụ về lệnh "Print" :

QUOTE
```
on(release) {
  print(this,"bframe");
}
```

Như bạn đã thấy, tuỳ chọn được coi như một chuỗi và được đặt trong dấu ""

2-PrintAsBitmap
Câu lệnh này làm việc tương tự như lệnh Print với hai tham số tương tự.
Điều khác biệt là lệnh Print sẽ gửi các đối tượng đồ hoạ và font chữ tới máy in. Sau đó, máy in sẽ xây dựng lại các vector và nội dung để in ra.
Ngược lại, PrintAsBitmap sẽ chuyển toàn bộ nội dung thành một ảnh bitmap lớn và gửi tới máy in.
Điểm thuận lới nhất của PrintAsBimap là độ trong suốt của văn bản cũng được in ra.
Nếu bạn có một đối tượng đồ hoạ bán trong suốt,
chắc có ưu điểm chắn bạn sẽ cần đến chức năng in này, đồng thời nó cũng hoạt động tốt với nhiều loại máy in hơn.
Lệnh in chính là văn bản được in ra có tính thẩm mỹ cao, đường cong smooth hơn và chữ rõ ràng hơn khi được in ra. Nó cũng nhanh hơn
khi in qua mạng.

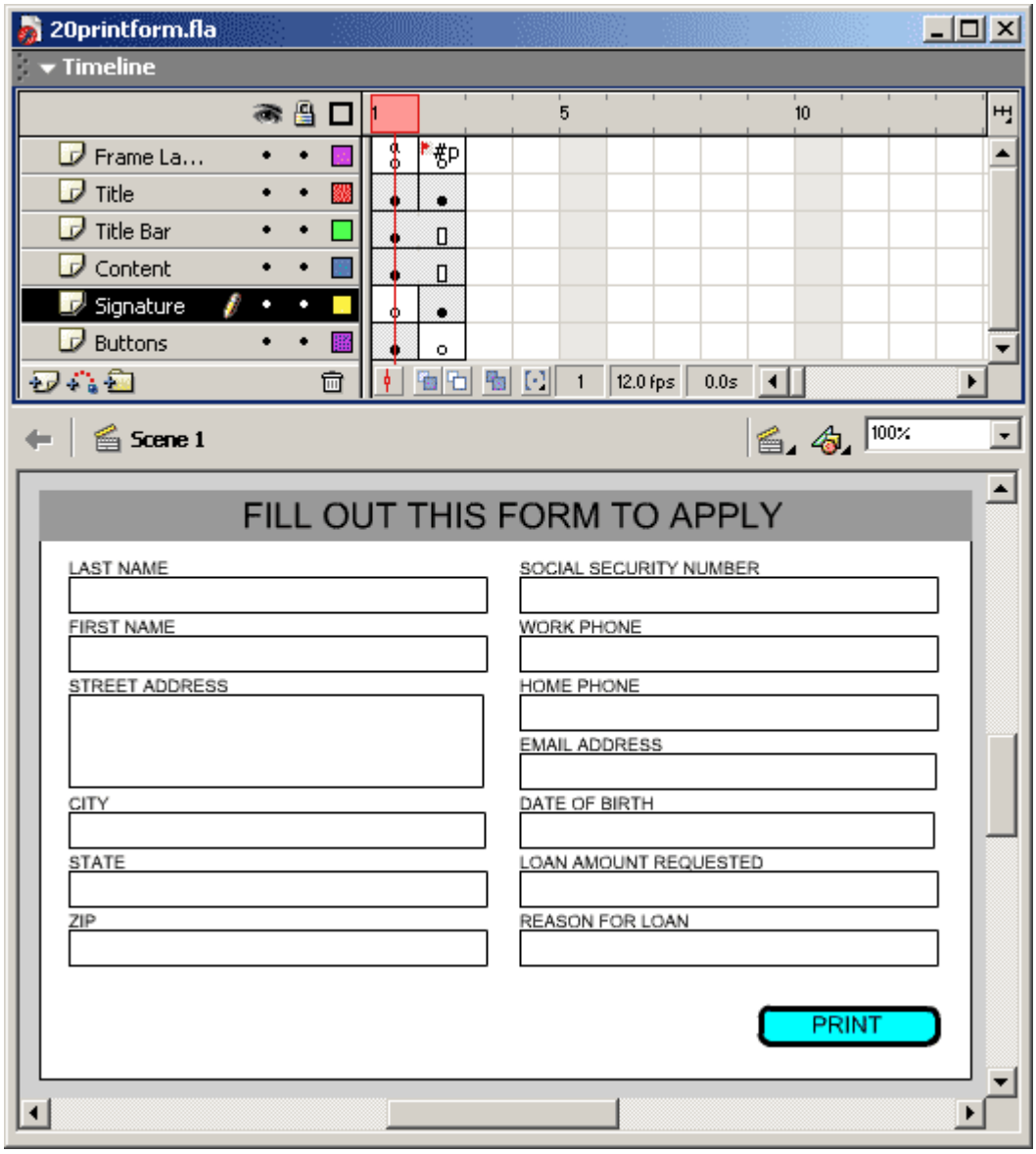
Nguyên tác chung nhất là sử dụng lệnh PrintAsBitmap khi bạn thật sự chắc chắn kết quả sẽ giống như nội dung hiển thị trên màn hình.
Sử dung Print khi văn bản không cần độ chính xác cao hoặc khi bạn xây dựng movie để điều khiển môi trường như mạng nội bộ.

Chú ý: Nếu bạn sưe dụng movie qua internet, lệnh in chỉ làm việc khi tất cả các frame trong movie được load xuống.

Thao tác : Tạo biểu mẫu in

Một điều mà người dùng ghét nhất trên một website là một biểu mẫu mà chúng ta phải in ra, điền vào, rồi gửi đi . Chúng ta đang sử dụng máy tính, đột nhiên lại phải đi kiếm cái bút trên bàn để viết ?
Vì vậy tại sao chúng ta không tạo ra biểu mẫu cho phép điền thông tin vào, sau đó được in ra vời đầy đủ nội dung đã được nhập vào.
Chúng ta sẽ làm ví dụ đơn giản này, và sẽ thấy chúng thuận tiện hơn rất nhiều so với một biêt mẫu html yêu cầu người sử dụng in ra toàn bộ nội dung của site.
Bạn hoàn toàn có thể điề khiển những gì sẽ được in ra, vì vậy những thứ linh tinh trên trang web có thể được bỏ qua và những yếu tố mới như email address có thể được thêm vào.

1- Tạo một movie mới
2- Tạo ra các trường nhập dữ liệu (ví dụ tên, tuổi, địa chỉ, giới tính, email...) (xem hình) sau đó đặt tên layer này là CONTENT

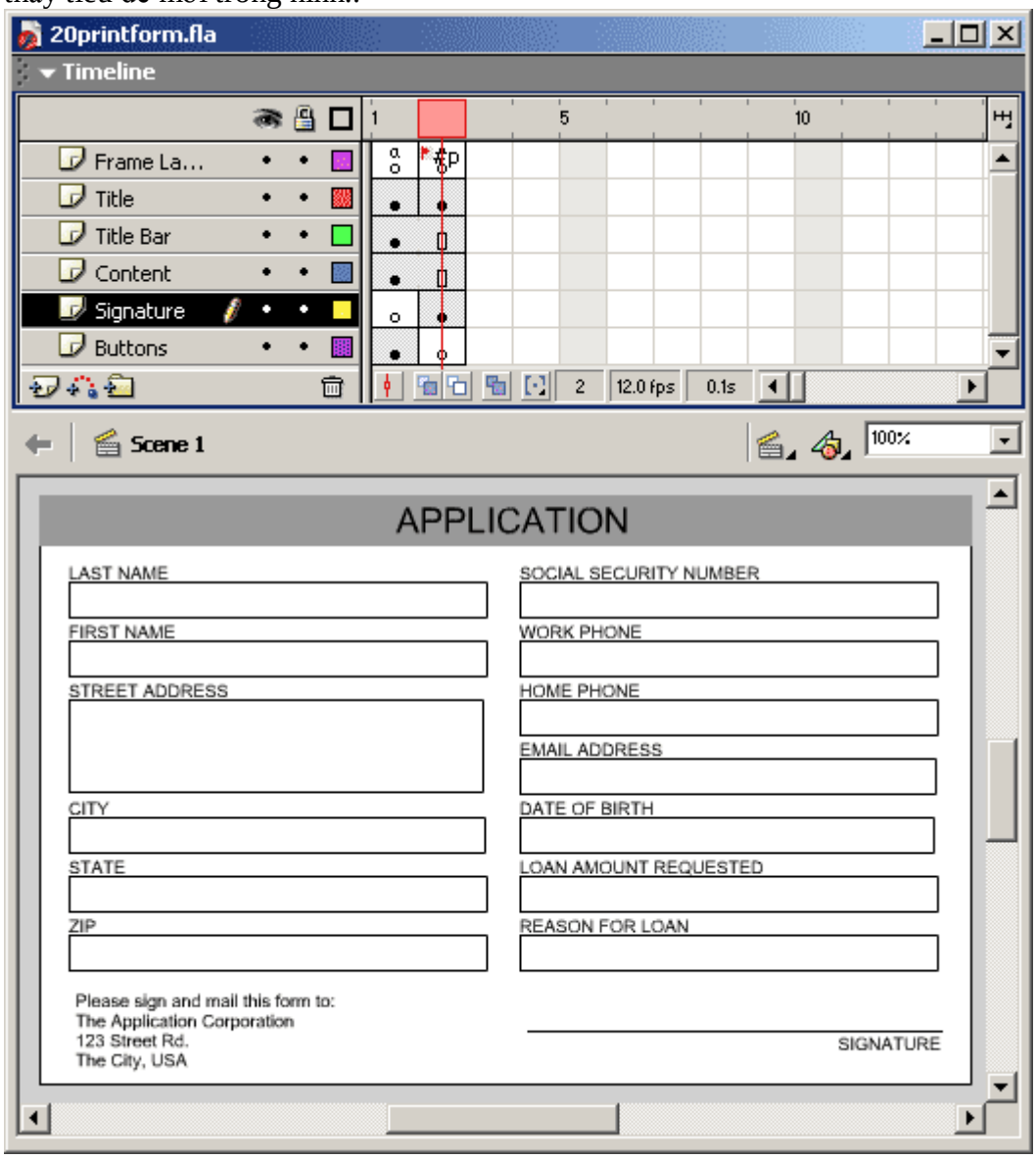


3-Đặt tiêu đề cho trường lên phía trên ô nhập liệu
4-Tạo thanh tiêu đề cho Form. Chú ý,hai loại tiêu đề này sẽ được đặt ở hai layer riêng biệt .
5-Tạo nút Print và đặt trong layer tên là Buttons
6-CHèn đoạn mã sau cho button :

QUOTE
on (release){
print (this, "bmax");
}

7- Thêm Frame thứ hai cho movie, kéo Title Bar và Content sang frame 2
Tạo blank keyframe cho layer Buttons và Titles, vì chúng sẽ có nội dung khác trong frame thứ hai
8- Trong frame thứ hai của layer Title, đặt một tiêu đề khác. Ví dụ, nếu frame thứ nhất là "Fill out that application" thì ở frame thứ hai chỉ cần là Appilcation

9- Sau đó, biểu mẫu đã sẵn sàng để điền vào. Tiêu đề cũ sẽ biến mất, bạn có thể nhìn thấy tiêu đề mới trong hình..



10- Thêm một layer mới tên là Signature. Đặt một keyframe ở frame thứ hai của layer này. Thêm đường ngăn cách và địa chỉ trong frame này. Nếu làm đúng thì frame 1 của layer
này sẽ không có gì, nội dung chỉ tồn tại ở frame 2
11- Thêm layers tên Frame Labels . Đặt trong đó hai keyframe. Keyframe thứ 2, chúng ta đặt ký hiệu "#p" . Keyframe thứ nhất chèn code

QUOTE
stop();

Test movie của bạn. Bạn hãy điền vào biểu mẫu đó, và ấn nút Print. Frame thứ hai sẽ được in ra với tiêu đề "Application", chữ ký, và địa chỉ...

Tóm tắt

Để in được các frame trong một movie bạn cần phải đặt label cho chúng là "#p". Bạn có thể đặt tên cho một hoặc nhiều frame trong cùng một movie.
Mọi thứ trong frame "#p" sẽ được in ra. Bạn có thể trang trí tuỳ thích đối với biểu mẫu được in ra, trừ những yếu tố như menu.
Bạn có thể sự dụng lệnh "Print" để in các frame sử dụng vector shape để gửi tới máy in. Tuy nhiên, nếu cần in những đối tượng đồ hoạ bán trong suốt hoặc chắc chắn rằng tất cả những biểu mẫu in ra giống nhau, bạn có thể sử dụng lệnh PrintAsBitmap

Giờ thứ 21: Sử dụng component, Using component

ActionScript cho các Component đi kèm Flash

Có 7 component gắn liền kèm theo chương trình Flash: ***CheckBox, ComboBox, ListBox, PushButton, RadioButton, ScrollBar, ScrollPane***.

Để thêm một component vào movie của bạn, bạn có thể click đúp lên component trong bảng Component, hoặc nhấn và kéo một component vào stage.

PushButton

Click và kéo component PushButton vào stage sẽ tạo một instance (thể hiện) mới của component PushButton trên màn stage. Khi đó bạn đã thêm một số phần tử Library vào trong movie. Rất may là các phần tử này được cất trong các folder rất gọn trong Library (thư viện), vì thế chúng cũng không ảnh hưởng mấy đến công việc của bạn.

Component PushButton trên stage trông rất đơn giản: chỉ là một khung với từ "**PushButton**" ở giữa.

Bạn nhớ bật tính năng Live Preview (xem trước) của Flash bằng cách chọn Control--->Enable Live Preview.

Sau khi PushButton đã ở trên stage, bạn có thể click chọn nó và mở bảng Properties để đổi tên. Hai thông số có thể thiết đặt cho component PushButton là **Label** (nhãn): ta đổi thành **Press Me !** và Click Handler (quản lý sự kiện nhấn nút): đây là tên của hàm được gọi khi nút được click. Hàm này phải nằm trong cùng Timeline với nút; vì thế nếu nút đặt ở level gốc (root level), hàm phải ở trên Timeline chính. Ta thiết đặt Click Handler thành buttonPressed (nút đã được nhấn).

Ngoài ra bạn đặt cho component một instance name (tên minh họa) là testButton (kiểm tra nút).

Bây giờ tất cả công việc phải làm là viết hàm buttonPressed. Dưới đây là một ví dụ đơn giản. Hàm này chỉ gửi một vài dòng text ra cửa sổ Output:

ActionScript

```
function buttonPressed(buttonInstance) {
  if (buttonInstance == testButton) {
     trace("Test Button Pushed.");
  }  else {
```

```
    trace(buttonInstance._name);
  }
}
```

Mỗi hàm quản lý nút sẽ chuyển một tham số: một tham chiếu đến nút gọi hàm. Vì thế bạn có thể kiểm tra instance này có phải có tên là testButton hay không. Hàm ví dụ ở trên sẽ chuyển thông báo "Test Button Pushed" nếu nút được nhấn là testButton, và in ra tên minh họa của nút nếu nút bị nhấn không phải là testButton.

Bạn có thể xem ví dụ mẫu này trong movie 21pushbutton.fla.

CheckBoxes
Component CheckBox (hộp kiểm) tương tự như cái mà chúng ta đã tạo ở giờ 15 (anh em dịch sau nhé).

Để tạo một CheckBox component, click đúp vào nó trong bảng Components, hoặc click và kéo nó lên trên stage. Để tạo một instance thứ hai, mở folder Flash UI Components trong thư viện Library và kéo component CheckBox lên trên stage.

Trong ví dụ movie 21checkboxes.fla, tôi đã tạo 3 hộp kiểm (CheckBoxes). Nếu bạn chọn một component trong chúng và mở bảng Properties, bạn sẽ thấy nó có nhiều thông số hơn so với component PushButton.

Bổ sung vào các thông số **Label** và **Change Handler**, giờ đây bạn có cả **Initial Value** (giá trị ban đầu) và **Label Placement** (sắp xếp nhãn). Thông số Initial Value là true hoặc false, tùy thuộc vào việc bạn muốn hộp kiểm ban đầu được đánh dấu hay không. Thông số Label Placement cho phép bạn sắp xếp các nhãn ở bên phải hoặc trái (right hoặc left) so với ô kiểm. Right là sắp xếp mặc định. Nếu bạn chuyển thành left, dòng chữ ghi nhãn của hộp kiểm sẽ xuất hiện ở bên trái của ô.

Trong movie mẫu, tôi đặt tên cho 3 instance CheckBox là option1, option2, và option3. Tôi cũng đặt nhãn (label: phần text bên cạnh ô kiểm) là Option One, Option Two, và Option Three. Thông số Change Handler của mỗi instance được đổi thành changeOptions.

Tôi đặt hàm changeOptions trong timeline chính. Hàm này sẽ được thực thi khi nào một trong các hộp kiểm CheckBox được click vào. Nó sẽ gửi tên và trạng thái mới của CheckBox ra cửa sổ Output.

ActionScript

```
function changeOptions(checkBoxInstance) {
  trace(checkBoxInstance._name+": "+checkBoxInstance.getValue());
}
```

Trong 21checkboxes.fla, tôi cũng thêm vào một component PushButton. Nút này được đặt tên là doneButton và sẽ gọi hàm buttonPressed. Hàm này sẽ lặp đi lặp lại với tất cả các CheckBox và gửi trạng thái của chúng (được đánh dấu hay chưa) ra cửa sổ

Output.

ActionScript

```
function buttonPressed(buttonInstance) {
  if (buttonInstance == doneButton) {
     trace("Option One: "+option1.getValue());
     trace("Option Two: "+option2.getValue());
     trace("Option Three: "+option3.getValue());
  }
}
```

Thay vì gửi các kết quả ra cửa sổ Output, hầu hết bạn sẽ muốn sử dụng chúng trong các dạng khác. Ví dụ, bạn có thể đặt chúng trong một đối tượng LoadVars để chúng có thể gửi tới một server.

RadioButtons

RadioButtons giống như CheckBoxes, ngoại trừ việc chúng được sắp xếp thành các nhóm. Tại một thời điểm, bạn chỉ có thể chọn một nút RadioButton trong một nhóm.

File mẫu 21radiobuttons.fla có ba component RadioButtons. Nếu bạn chọn một trong những nút đó và mở bảng các thuộc tính (Properties panel) cho nút, bạn sẽ nhìn thấy component này có nhiều thông số hơn so với các component CheckBox hay PushButton.

Thêm vào các thông số bạn đã thấy trong component CheckBox là hai thông số **Group Name** (tên nhóm) và **Data** (dữ liệu). Thông số Group Name xác định xem RadioButton thuộc về nhóm nào. Trong movie mẫu, cả ba RadioButtons đều thiết đặt thông số này là firstGroup. Nếu đã có một nhóm nút thứ hai với một tên khác, thì hai nhóm này được xem là độc lập với nhau khi quyết định RadioButton nào được bật.

Thông số Data là tùy chọn và bạn có thể sử dụng trong các đoạn mã của mình. Bạn có thể truy cập (access) nó bởi hàm getData(). Bạn có thể lưu trữ các lệnh mà đoạn mã của bạn thực thi khi nút radio được chọn.

Trong movie mẫu, ba RadioButtons được đặt tên là choice1, choice2, và choice3. Nhãn của ba nút này là Choice One, Choice Two, và Choice Three.

Việc xác định xem người dùng đã chọn nút radio nào sẽ được thực hiện khi PushButton trong movie được click. Sau đó PushButton sẽ chạy đoạn script này để quyết định xem lựa chọn nào đã được thiết lập. Đoạn script sẽ lặp đi lặp lại với cả ba nút để tìm kiếm một nút trả về true từ hàm getState(). Điều này nghĩa là nút RadioButton đó đã được bật.

ActionScript

```
function buttonPressed(buttonInstance) {
  if (buttonInstance == doneButton) {
     var choice = "none";
```

```
    for(i=1;i<=3;i++) {
      if (this["choice"+i].getState()) {
        choice = this["choice"+i]._name;
      }
    }
    trace("Choice: "+choice);
  }
}
```

ListBox

Một ListBox (hộp danh sách) là một phương pháp đơn giản cho phép người dùng lựa chọn một hoặc nhiều tùy chọn. Một ListBox có thể đặt như một thiết lập của CheckBoxes hoặc RadioButtons. Nó đặc biệt hữu ích khi bạn có nhiều lựa chọn nhưng khoảng trống trên màn hình có hạn.

Một khung danh sách trong như một trường text cuộn--trên thực tế là như vậy. Mỗi dòng tương ứng với một lựa chọn riêng biệt của người sử dụng. Nếu có nhiều lựa chọn hơn vùng mà khung danh sách có thể hiển thị thì người dùng có thể cuộn lên và cuộn xuống để xem hết các mục trong danh sách.

Khi bạn tạo một instance mới của component ListBox, bạn phải thiết đặt thông số Select Multiple (lựa chọn nhiều dòng) của nó. Nếu tham số này là true, người dùng có thể dùng các phím Shift, Command, hoặc Ctrl để lựa chọn nhiều hơn một dòng. Nếu là false, mỗi lần bạn chỉ có thể chọn được một dòng.

Thêm vào đó, bạn phải thiết đặt thông số Labels (nhãn). Tuy nhiên, đây không phải là một giá trị đơn mà là một mảng các giá trị. Flash có một giao diện đặc biệt (special interface) cho việc nhập các giá trị này. Khi bạn click trên tham số Labels trong bảng Properties, bạn sẽ bắt gặp một hộp thoại cho phép bạn nhập vào một mảng các mục (item) cho các thông số khác.

Bạn cũng có một tham số Data (dữ liệu) để tạo một mảng dữ liệu. Thông số Data này, giống như thông số data đã dùng với các nút radio, cho phép đoạn mã của bạn lấy thông tin bổ sung về các lựa chọn mà người dùng đã chọn. Tuy nhiên, thông số này không bắt buộc phải có.

Trong movie mẫu 21listbox.fla, tôi đặt một component ListBox với ba lựa chọn trên màn hình. Chúng được thiết đặt là có thể chọn nhiều dòng. Khi người dùng click lên trên một dòng, hàm listBoxChange sẽ được gọi. Điều này được xác định bởi thông số Change Handler của nó. Hàm này cho bạn biết dòng nào (lựa chọn nào) vừa được chọn:

ActionScript

```
function listBoxChange(listBoxInstance) {
  trace(listBoxInstance.getValue());
}
```

Trong movie mẫu này cũng có một component PushButton. Khi nó được click, nó sẽ thực thi hàm này. Nó sử dụng hàm getSelectedItems() để lấy một mảng các lựa chọn (choices) đã chọn trong list box. Mỗi mục chọn trong mảng là một đối tượng với một thuộc tính label và data. Vì chúng ta đã không sử dụng các thuộc tính data của hộp danh sách (list box), nên thay vào đó chúng ta sẽ lấy các nhãn (label).

ActionScript

```
function buttonPressed(buttonInstance) {
  if (buttonInstance == doneButton) {
     items = myListBox.getSelectedItems();
     for(var i=0;i<items.length;i++) {
        trace(items[i].label);
     }
  }
}
```

Bạn cũng có thể thêm hoặc bớt các dòng từ list box bằng cách sử dụng ActionScript. Ví dụ, addItem sẽ thêm một lựa chọn bổ sung vào list box.

ActionScript

```
myListBox.addItem("Choice Four");
```

Bạn có thể dùng addItemAt, removeItemAt, và replaceItemAt để thay đổi list box bởi ActionScript.

ComboBox
Một combo box giống như một menu kéo xuống (pull-downl menu) ở đó người dùng cũng có thể gõ vào một giá trị.

May thay, bạn cũng có thể tắt tùy chọn hiệu chỉnh giá trị. Khi đó combo box như một menu pull-down bình thường. Thông số để làm điều này là Editable (có thể hiệu chỉnh) trong hộp thoại Properties. Bổ sung cho tham số đó, bạn cũng có thể cung cấp các mảng chứa nhãn (Labels) và dữ liệu (Data).

Một thông số khác của combo box là **Row Count** (số dòng). Các combo box có thể nhỏ như các list box. Khi người dùng click vào, chúng sẽ trải rộng thành một danh sách các lựa chọn. Nếu số lựa chọn vượt quá giá trị Row Count, một thanh cuộn sẽ xuất hiện ở bên phải cho phép người dùng cuộn lên xuống để lựa chọn.

Cơ bản một combo box có thể có 3 trạng thái. Khi không hoạt động nó thu nhỏ thành một dòng. Khi click vào nó, combo box sẽ trải rộng thành danh sách tuỳ chọn, nếu số lựa chọn lớn hơn số dòng có thể hiển thị thì xuất hiện thêm thanh cuộn.

Khi một người dùng chọn một lựa chọn mới trong combo box, bộ quản lý Click Handler sẽ được gọi. Dưới đây là một hàm đơn giản cho bạn biết nhãn của combo box

đã chọn:

ActionScript

```
function comboBoxChange(comboBoxInstance) {
  trace(comboBoxInstance.getValue());
}
```

Bạn cũng có thể dùng getSelectedIndex() để lấy chỉ mục (tính từ 0) của mục đã chọn.

Movie mẫu 21combobox.fla là một ví dụ về component ComboBox.

ScrollPane
Hai component tiếp theo khác hẳn so với 5 component ở trên. Chúng không dùng để cho phép người dùng lựa chọn, nhưng dùng để hiển thị lượng thông tin lớn trong các khoảng nhỏ (cuộn mà lị).

Component ScrollPane (ô cuộn) gồm có một thanh cuộn dọc, cuộn ngang và một vùng hiển thị hình chữ nhật. Thông số chính của component này là Scroll Content (cuộn nội dung). Đây là tên liên kết (Linkage name) cho một movie clip. Khi bạn chạy movie, movie clip được copy từ Library và đặt vào vùng hiển thị của ô cuộn. Sau đó các thanh cuộn sẽ cho phép người dùng nhìn thấy các phần khác nhau của movie clip.

Bạn có thể xem ví dụ trong file 21scrollpane.fla.

Nếu bạn thiết đặt thông số **Drag Content** là true, người dùng cũng có thể click vào trong vùng hiển thị và kéo hình trong đó chạy. Các thanh cuộn cũng thay đổi khi bạn kéo nội dung trong ô cuộn (chính là cái movie clip hiển thị trong ô đó).

Mặc dù component ScrollPane không đòi hỏi bất kỳ ActionScript nào để làm việc, nhưng có rất nhiều hàm mà bạn có thể dùng để xác định xem phần nào của movie clip đang được xem hoặc để thay đổi chiều rộng (width) và chiều cao (height) của ô.

Bạn cũng có thể dùng bảng Properties để thay đổi chiều rộng và chiều cao của ô cuộn. Khi bạn làm việc đó, ô cuộn trông bị méo mó trong Flash, nhưng no star where, nó sẽ ngon lành ngay khi bạn chạy movie.

Một lệnh ActionScript rất hữu ích là loadScrollContent . Lệnh này sử dụng một địa chỉ URL và hiển thị một movie clip ở ngoài vào trong ô cuộn.

ActionScript

```
myScrollPane.loadScrollContent("myMovieClipFile.swf");
```

Ô cuộn có thể được dùng như một trình duyệt hình ảnh.

ScrollBar

Component cuối cùng là ScrollBar. Component này thêm các thanh cuộn vào trường text. Bạn có thể dùng component này mà không cần dùng bất kỳ code ActionScript nào. Chỉ việc kéo và thả một component ScrollBar vào một trường text (text field), tự nó sẽ thêm vào trường text.

Component ScrollBar có ít thuộc tính Actionscript có thể sử dụng được. Ví dụ, bạn có thể sử dụng **getScrollPosition()** để lấy vị trí cuộn hiện tại và **setScrollPosition()** để thay đổi nó.

Thực hành với Components
Bây giờ chúng ta sẽ phối hợp 5 component khác nhau để tạo một form nhập dữ liệu: CheckBox, RadioButton, ComboBox, ListBox, và PushButton components.

Tạo một Flash movie mới.

Tạo ba component CheckBox. Đặt tên cho chúng là checkbox1, checkbox2, và checkbox3. Nhãn của chúng là: Macintosh, Windows, Linux.

Tạo ba component RadioButton. Đặt tên cho chúng là radiobutton1, radiobutton2, và radiobutton3. Nhãn của chúng lần lượt là ... như trên

Tạo một ComboBox component. Đặt tên cho nó là combobox. Thêm một vài nhãn (label) vào đó để người dùng có thể chọn lựa.

Tạo một ListBox component đặt tên là listbox. Thêm bao nhiêu nhãn vào tuỳ bạn. Đừng lo lắng về thứ tự của chúng vì chúng ta sẽ sắp xếp lại sau. Thiết đặt thông số ListBox Multiple Selections thành false. Dùng bảng Properties để tạo cho khung danh sách (list box) rộng 200 và cao 200 pixel.

Thêm một PushButton component. Đặt cho nó nhãn là Done và thông số điều khiển Click Handler là buttonPressed.

Thêm dòng sau vào trong frame script. Nó sẽ sắp xếp lại nhãn của các mục trong component ListBox.

ActionScript

```
listbox.sortItemsBy("label","Asc");
```

Lệnh sortItemsBy làm việc rất tốt với component ComboBox. Bạn có thể dùng "label" hoặc "data" cho thông số đầu tiên. Điều đó tùy thuộc vào nhãn (label) hay các trường dữ liệu (data) sẽ được sử dụng để sắp xếp. Thông số thứ hai có thể là "Asc" (sắp xếp theo thứ tự tăng dần) hoặc "Desc" (giảm dần).

Component PushButton sẽ gọi hàm buttonPressed. Chúng ta sẽ tạo hàm này theo từng đoạn nhỏ để xử lý từng phần của form.

Hàm bắt đầu bằng việc tạo một mảng mới. Sau đó nó kiểm tra từng check box xem hàm getValue() của nó có phải là true không. Nếu là true, nhãn của check box đó sẽ

được thêm vào mảng đó. Khi vòng lặp kết thúc, mảng own chứa bất kỳ lựa chọn nào mà người dùng đã làm với các check box.

ActionScript

```
function buttonPressed(buttonInstance) {
  if (buttonInstance == doneButton) {

    // sưu tập mảng của các CheckBoxes đã chọn
    own = new Array();
    for(var i=1;i<=3;i++) {
      if (this["checkbox"+i].getValue()) {
        own.push(this["checkbox"+i].getLabel());
      }
    }
    trace("Computers Owned: "+own);
```

Đoạn mã tiếp theo kiểm tra tất cả các component RadioButton và ghi nhớ xem component nào đã được bật (đã được chọn):

ActionScript

```
// xác định RadioButton nào đã lựa chọn
favorite = "none";
for(var i=1;i<=3;i++) {
  if (this["radiobutton"+i].getState()) {
    favorite = this["radiobutton"+i].getLabel();
  }
}
trace("Favorite: "+favorite);
```

Component đơn giản nhất là combo box. Đoạn mã này chỉ đơn giản trả về giá trị của nó:

ActionScript

```
// lấy giá trị của ComboBox ComboBox
nextPurchase = comboBox.getValue();
trace("Next Purchase: "+nextPurchase);
```

Để check (kiểm) nhiều lựa chọn của list box, bạn cần lặp từ đầu đến cuối mảng trả về bởi getSelectedItems(). Sau đó bạn cần xem xét thuộc tính label của mỗi mục chọn (item).

Đoạn mã sau sẽ làm điều đó và xây dựng một mảng chứa các nhãn đã lựa chọn:

ActionScript

```
    // sưu tập các lựa chọn của ListBox
    uses = new Array();
    items = listbox.getSelectedItems();
    for(var i=0;i<items.length;i++) {
       uses.push(items[i].label);
    }
    trace("Uses: "+uses);
  }
}
```

Nếu thực sự hiện tại bạn đang làm điều này, nên sử dụng các đối tượng có tính cấu trúc hơn là sử dụng lệnh trace. Cho ví dụ, bạn có thể tạo một đối tượng LoadVars để sau đó gửi thông tin đến máy chủ.

Thay đổi kiểu dáng (Style) của một Component
Các component của Flash vốn trông đã rất đẹp. Nhưng nếu tất cả các nhà phát triển Flash đều bắt đầu sử dụng chúng, thì tất cả các Flash movie của chúng ta trông sẽ na ná nhau.

May thay, bạn có thể tuỳ chỉnh các component theo nhiều cách khác nhau. Bạn có thể dễ dàng tạo các giao diện riêng (custom skin) cho chúng. Dưới đây là ba cách tùy chỉnh component bằng cách sử dụng ActionScript.

Global Customization (tuỳ biến chung)
Sử dụng đối tượng globalStyleFormat, bạn có thể tùy chỉnh giao diện cho tất cả các component trong một lần. Đây là một ví dụ thay đổi màu chữ của tất cả các text trong tất cả các component thành màu xanh da trời (blue):

ActionScript

```
globalStyleFormat.textColor = 0x0000FF;
globalStyleFormat.applyChanges();
```

Lệnh applyChanges chấp nhận sự thay đổi. Ngoài ra, bạn có thể thiết đặt nhiều thuộc tính khác. Dưới đây là một số thay đổi chi tiết:

ActionScript

```
globalStyleFormat.textColor = 0x0000FF;
globalStyleFormat.textFont = "Arial";
globalStyleFormat.textSize = 18;
globalStyleFormat.textBold = true;
globalStyleFormat.applyChanges();
```

Bạn có thể thay đổi thành nhiều giá trị như font chẳng hạn. Số mục thay đổi trong kiểu dáng của các component quá dài để có thể liệt kê hết ở đây. Bạn có thể thay đổi

màu sắc và kiểu dáng của các dấu kiểm trong CheckBoxes, các hình tròn trong những RadioButton, những mũi tên trong các thanh cuộn ScrollBars, màu nền, màu sắc của vùng tô sáng (highlight colors), màu sắc các mục chọn, v.v... Hãy xem trong chương trình Flash của bạn để biết đầy đủ thông tin về danh sách này.

Grouped Customization (tùy chỉnh theo nhóm)
Mặc dù đối tượng globalStyleFormat đã được sử dụng bởi tất cả các component trên stage, bạn vẫn có thể tạo cho các đối tượng của bạn kiểu dáng (style) riêng biệt chỉ dùng trong một vài component mà bạn chỉ định.

Bạn thực hiện điều đó bằng cách tạo một đối tượng FStyleFormat. Khi bạn làm điều này, đối tượng mới của bạn sẽ có các thuộc tính giống như đối tượng globalStyleFormat.

Ví dụ, bạn có thể tạo một đối tượng kiểu dáng (style object) và thiết đặt màu sắc của nó thành đỏ tươi như dưới đây:

ActionScript

```
myStyle = new FStyleFormat();
myStyle.textColor = 0xFF00FF;
```

Các yếu tố khác của style không được thiết đặt chính xác trong đối tượng style này. Vì thế bạn có thể áp dụng style này cho một component, mà diện mạo của style không thay đổi.

Để áp dụng kiểu dáng style này vào một component, hãy sử dụng lệnh addListener:

```
myStyle.addListener(radiobutton1);
```

Bạn nghĩ có vẻ kỳ quặc khi sử dụng với addListener, đúng là như vậy. Hãy hiểu rằng: bạn báo cho component để tiếp nhận (listen) đối tượng style.

Single Component Customization (tùy chỉnh component riêng lẻ)
Bạn cũng có thể thiết đặt một cách chính xác một trong những thuộc tính của style. Tuy nhiên, bạn không thể làm điều đó bằng cách sử dụng cú pháp chấm (dot) gọn và đẹp như bạn mong muốn. Thay vì thế, bạn cần phải dùng lệnh setStyleProperty. Lệnh này sẽ đặt thuộc tính style dưới dạng một chuỗi ở tham số đầu tiên và giá trị mà bạn muốn thiết đặt cho nó ở tham số thứ hai.

ActionScript

```
checkbox1.setStyleProperty("textColor",0xFF0000);
```

Bằng cách sử dụng ba phương pháp thiết đặt style cho các component, bạn có thể tùy chỉnh các component theo như mong muốn.

Giờ thứ 22: Điều khiển âm thanh với ActionScript

Có hai cách để bạn có thể cho âm thanh vào đoạn phim Flash của mình. Cách thứ nhất là đặt nó vào ngay timeline. Việc này không cần phải sử dụng mã. Cách thứ hai là sử dụng ActionScript để chơi những đoạn nhạc được lưu trữ trong Library.

Tìm hiểu cách truy xuất âm thanh với ActionScript:

Việc kết nối và chơi nhạc:

Điều đầu tiên mà bạn cần trước khi chơi một đoạn nhạc với ActionScript là kết nối đến nó. Bạn hãy "Import" file nhạc vào Library, bấm phải chuột chọn "Linkage", rồi đánh dấu vào các phần export. Bạn cũng có thể đặt một cái tên "Identifier" khác cho đoạn nhạc thay vì để tên file nhạc.
Ví dụ, nếu bạn "import" file nhạc "mysound.wav", nó sẽ xuất hiện trong Library với cái tên mysound.wav. Khi bạn chọn Linkage, và đánh dấu export, nó sẽ được đặt tên "Identifier" ngay là "mysound.wav", nhưng bạn có thể sửa thành bất kì tên gi bạn muốn(vd: nhac1). Đó chính là cái tên mà bạn sẽ sử dụng trong ActionScript.

Để chơi một đoạn nhạc bằng ActionScript, bạn phải làm ít nhất ba bước:
Đầu tiên là tạo một đối tượng "Sound" :

ActionScript
```
mySound = new Sound();
```

Thứ hai, bạn cần gắn đoạn nhạc trong thư viện vào đối tượng "Sound" này:
mySound.attachSound("mySound.wav") //mySound.wav là tên bạn đã đặt trong phần "Identifier"
Cuối cùng, bạn hãy ra lệnh cho đối tượng "Sound" của bạn chơi đoạn nhạc:

ActionScript
```
mySound.start();
```

Và đây là một đoạn đơn giản đặt vào trong một Button để chơi đoạn nhạc từ Library:

ActionScript
```
on (release) {
  mySound = new Sound();
  mySound.attachSound("poof.wav");
  mySound.start();
}
```

Bạn có thể tham khảo một ví dụ đơn giản trong file "22playsound.fla" kèm theo quyển sách này.
Bạn cũng có thể xem qua một phương pháp hơi phức tạp một tí trong file "22playsoundfunction.fla". Trong file Flash này, có một function tên là "playsound" được đặt ở timeline chính. "function" này bao gồm tất cả các mã mà bạn cần để chơi một đoạn nhạc đơn giản.

ActionScript

```
function playSound(soundName,balance) {
  var mySound = new Sound();
  mySound.attachSound(soundName);
  mySound.start();
}
```

Với việc sử dụng function này, bạn đã đơn giản hóa đoạn ActionScript nên bạn sẽ chỉ cần sử dụng đúng một dòng lệnh để chơi nhạc. Đây là đoạn mã đặt trong một Button để sử dụng function này:

ActionScript

```
on (release) {
  playSound("poof.wav");
}
```

Câu lệnh "start"

Câu lệnh "start" trong ví dụ trên có thể được sử dụng bằng nhiều cách. Bạn có thể thêm hai tham số nữa vào nó để thay đổi cách mà đoạn nhạc sẽ được chơi.

Tham số thứ nhất bạn có thể thêm được gọi là "offset". Nó giúp bạn có thể chơi bản nhạc từ bất kì vị trí nào bạn muốn chứ không phải chơi lại từ đầu. Ví dụ, dòng lệnh này sẽ bắt đầu chơi từ vị trí thứ 1000 miligiây của đoạn nhạc(1 giây sau khi bật):

ActionScript

```
mySound.start(1000);
```

Tham số thứ hai của câu lệnh "start" là số lần lặp lại của đoạn nhạc. Ví dụ, nếu bạn muốn bắt đầu bản nhạc tại vị trí sau 1 giây và lặp ba lần, câu lệnh sẽ như sau:

ActionScript

```
mySound.start(1000,3);
```

Bắt đầu bản nhạc tại vị trí từ đầu và lặp ba lần:

ActionScript

```
mySound.start(0,3);
```

Câu lệnh đi đôi với "start" là "stop". Khi bạn đang chơi một đoạn nhạc, bạn có thể đưa ra câu lệnh "stop" bất cứ khi này để ngừng hẳn đoạn nhạc. Bạn phải thêm vào tham số là tên (Identifier) của đoạn nhạc. Nếu không, lệnh "stop" sẽ ngừng tất cả các đoạn nhạc đang chơi.

ActionScript
```
mySound.stop("poof.wav");
```

Điều chỉnh âm lượng

Bạn có thể sửa đổi đoạn trước và khi đang chơi bằng một số câu lệnh. Các câu lệnh này còn có thể điều chỉnh âm lượng của đoạn nhạc, trong tất cả loa hay chỉ từng cái.

Lệnh "setVolume" là cách điều chỉnh đơn giản nhất, bạn chỉ cần cho một tham số từ 0 đến 100 là bạn có thể vặn to, nhỏ một đoạn nhạc:

ActionScript
```
mySound.setVolume(50);
```

Trong file "22playsoundvolume.fla" bao gồm một nút Play và bốn nút khác để điều chỉnh âm lượng lần từ là 0, 10, 50 và cuối cùng là 100. Nút Play sẽ chơi một đoạn nhạc 100 lần nên bạn có thể thử điều chỉnh âm lượng trong lúc đang chơi nhạc.

Chú ý là việc bạn điều chỉnh âm thanh của một đoạn nhạc sẽ không liên quan đến các đoạn nhạc khác. Vì vậy bạn có thể điều chỉnh các âm thanh khác nhau như nhạc nền và các tiếng động.

Đối tượng "Sound"

Đối tượng "Sound" có hai thuộc tính mà bạn nên biết đến. Thứ nhất là "duration" là độ dài của đoạn nhạc tính bằng miligiây. Thứ hai là "position" là vị trí mà đoạn nhạc đang chơi, cũng tính bằng miligiây.

Ví dụ, nếu một đoạn nhạc có thuộc tính "duration" bằng 3000, có nghĩa là nó dài 3 giây. Nếu thuộc tính "position" bằng 1500 thì đoạn nhạc đang chơi ở chính giữa.

Trong file "22tracksound.fla" thể hiện cách dùng "position" và "duration" để thể hiện của đoạn nhạc. Sau khi bắt đầu, một vạch đen sẽ chạy dần dần theo vị trí của đoạn nhạc.

ActionScript
```
onClipEvent(enterFrame) {
  // tính xem đã chơi được bao nhiêu của đoạn nhạc (giá trị từ 0.0 đến 1.0)
  percentPlayed = thisSound.position/thisSound.duration;

  // lấy độ dài của thanh
  barWidth = _root.bar._width;

  // đặt vị trí của dấu vạch
  _root.mark._x = _root.bar._x + barWidth*percentPlayed;
}
```

Khi đoạn nhạc kết thúc:

Bạn có thể sử dụng thuộc tính "position" và "duration" để kiểm tra một đoạn nhạc hết hay chưa, khi đó hai giá trị này sẽ bằng nhau. Tuy nhiên nếu đoạn nhạc được lặp nhiều lần thì hai giá trị này sẽ bằng nhau mỗi khi cuối đoạn nhạc.

Một cách tốt hơn để kiểm tra khi nào đoạn nhạc kết thúc là dùng "onSoundComplete". Đây là một function sẽ được thực thi khi đoạn nhạc kết thúc hẳn.

ActionScript

```
mySound = new Sound();
mySound.attachSound("mySound.aif");
mySound.onSoundComplete = function () {
  trace("sound done!");
}
mySound.start();
```

Điều chỉnh độ cân bằng:

Bạn có thể điều chỉnh âm lượng của loa này to hơn loa khác thông qua lệnh "setPan". Điều này giống như giàn máy stereo. Bạn có thể đặt giá trị từ -100 đến 100. Nếu bạn để là -100, tất cả âm thanh sẽ được phát ra bên loa trái, còn 100 sẽ là loa phải.

ActionScript

```
mySound.setPan(-100);
```

Bạn có thể tham khảo file "22monopan.fla". Khi bạn bấm nào nút "Play" bên trái, loa trái sẽ phát ra, bấm nút "Play" bên phải, loa phải sẽ phát ra.

Giờ thứ 23: Quản lý streaming cho movie, Managing Movie Streaming

Trong giờ 23 chúng ta sẽ tiến hành những bước sau:

- **Tìm hiểu, giám sát movie khi load**
- **Làm 1 loader đơn giản**
- **Làm 1 loader phức tạp hơn với loading bar**
- **Load các media movie ở bên ngoài vào flash (sound, image)**

Chúng ta bắt đâu với bước 1 nhé :

Các Movie Flash đều có tính chất stream. Điều này có nghĩa là frame đầu tiên của movie sẽ bắt đầu ngay khi nó được nạp xuống, ko phụ thuộc vào việc cái frame cuối cùng đã được load xong hay chưa.
Bạn có thể ko muốn việc này xảy ra. Ví dụ như movie của bạn là 1 đoạn animation

ngắn, bạn có thể ko muốn nó bắt đầu cho đến khi movie được load hoàn toàn từ web. Có vài cách để bắt movie đợi cho đến khi loading xong. Cách thông dụng nhất là ta tạo 1 loader frame. Đó là frame đầu tiên của movie. Nó sẽ quan sát tính chất của movie và xác định xem khi nao thì movie kết thúc việc loading.

Để biết được có bao nhiêu frame đã được load xuống bạn sử dụng tính chất _framesLoaded, còn để biết tổng số Frame của movie bạn sử dụng tính chất: _framesTotal.

Bạn có thể sử dụng điều này trong 1 số trường hợp đơn giản. Ví dụ bạn đặt ở frame đầu tiên lệnh stop();
Ở đó ta tạo 1 button cho phép người sử dụng tiếp túc. Khi người sử dụng click vào btn bạn có thể dùng 1 đoạn script tương tự như sau để xác định xem phải làm gì tiếp theo:

CODE
```
on (release) {
  if (_root._framesLoaded == _root._totalFrames) {
    play();
  } else {
    textDisplay = "Wait a few seconds and press again.";
  }
}
```

Nếu như movie chưa được load hết thì text field mà ta đã liên kết qua var textDisplay sẽ thông báo cho user.
Bạn cũng có thể sử dụng kĩ thuật này trong 1 phần của 1 movie dài. Ví dụ btn có thể nằm ở Frame 50 và chỉ để người sử dụng tiếp tục khi mà 50 frames tiếp theo đã sẵn sàng. Ta dùng đoạn code sau để thực hiện:

CODE
```
onClipEvent(load) {
  _root.textDisplay = "Waiting for the next sequence to load.";
  _root.stop();
}

onClipEvent(enterFrame) {
  if (_root._framesLoaded >= 100) {
    _root.play();
  }
}
```

Đây là phần đầu cơ bản của 1 loader script. Tuy nhiên có các cách khác chính xác hơn để giám sát việc loading hơn là tính số frames. Bạn có thể sử dụng getBytesLoaded and getBytesTotal để tính tổng số file và số file đã load.
Đây là đoạn script đặt vào trong 1 mc ở frame đầu tiên của movie. Ở frame đầu tiên bạn chú ý đặt thêm lệnh stop();

CODE
```
onClipEvent(enterFrame) {
  if (_root.getBytesLoaded() == _root.getBytesTotal()) {
    _root.play();
  }
}
```

Ở mỗi lần enterFrame điều kiện sẽ được kiểm tra và khi thỏa mãn tức movie được load hoàn toàn thì movie sẽ được play tiếp tục

Chúng ta tiếp tục tiến hành làm **1 loader đơn giản** để cho toàn bộ movie được load trước khi nó vượt qua frame 1 :

1.Mở 1 file mới.
2.Ở frame đầu ta tạo 1 keyframe
3.Tạo thêm 1 kf mới ở frame 2
Để có thể test cái loader 1 cách rõ ràng thì frame 2 nên chứa 1 movie tối thiểu là 100 K. Cách tốt nhất để tăng dung lượng là ta import 1 video.

4.Trở lại frame 1, chúng ta muốn có 1 movie chờ cho đến khi toàn bộ movie đã được load trước khi tiếp tục sang frame 2 ---> cho 1 lệnh **stop();** vào frame 1 này.
Ta tạo 1 shape đơn giản và convert nó sang mc và tống cổ nó sang 1 góc màn hình mà người xem ko nhìn thấy (chuối rừng). Ta cho đoạn script sau vào:

CODE
```
onClipEvent(enterFrame) {
  bytesLoaded = _root.getBytesLoaded();
  bytesTotal = _root.getBytesTotal()
  percentLoaded = Math.round(100*bytesLoaded/bytesTotal);
  _root.displayText = "Loading: "+percentLoaded+"%";
  if (bytesLoaded == bytesTotal) {
    _root.play();
  }
}
```

đoạn script sẽ kiểm tra getBytesLoaded để xem liệu movie đã kết thúc việc loading chưa. Ở đây ta tính số phần trăm đã load (percentLoaded) và cho hiển thị con số này qua biến displayText ở root level
Bạn đừng quên tạo 1 dynamic text field và đặt var cho nó là **displayText** nhe.

Sẽ rất khó kiểm tra khi bạn test cái movie này với Flash player trên máy vì movie của bạn chạy nhanh quá, ko kịp nhìn cái loader. Bởi vậy nên publish nó lên 1 trang web rùi test . Hoặc bạn có thể giả thiết lập 1 cai modem 56 K
Nào bây giờ goto website và test bạn sẽ thấy quá trình loading được hiện ở số phần trăm ở text field. Khi đạt đến 100% movie sẽ tiếp tục play.
Bạn có thể kiểm tra movie của bạn với file: **23simpleloader.fla**

3.Để cho cái loader của chúng ta thêm đẹp, chúng ta tiến hành bước 3 làm **1 loader**

phức tạp hơn có thêm 1 cái progess bar nữa nhé :
các bước làm như sau:

1.Bắt đầu 1 movie mới

2.Vẽ 1 hình chữ nhật rộng với border

3.Chọn toàn bộ cái hình chữ nhật này và đặt chúng vào 1 mc bằng cách chọn insert và convert to mc.

4.Click 2 cái vào mc mới này để edit nó

5.Tách riêng phần fill và border của cái hình chữ nhật ra làm 2 layer

6.Copy cái fill của hình chữ nhật và tạo 1 layer mới để paste nó vào. Layer này nên nằm ở trước và layer có chứa hình chữ nhật cũ thì nằm sau nó.

7.Chọn hình chữ nhật mới này(cai fill) và chọn cho nó màu tối hơn. đăt vị trí khớp với cái border, phía trên hình chữ nhât cũ

8.Bây giờ ta convert nó sang mc và đặt instance name là **barFill**
Double click vào mc mới này và ta chỉnh reg. point cho nó là góc tận cùng bên trái.

Trở lại movie chính, ta tạo thêm 1 layer mới. Và đặt 1 dynamic text ở đó. Đặt var cho nó là displaytext, bạn nhớ chọn font chữ đẹp dẹp và màu cũng đẹp đẹp 1 chút nhé
Ở movie của timeline chính ta đặt đoạn code sau vào mc chính của chúng ta:

CODE
```
onClipEvent(load) {
  // initialize variables
  bytesLoaded = 0;
  bytesTotal = _root.getBytesTotal();
}
```

Đoạn code enterFrame phải làm nhiều việc nhất. Nó có nhiệm vụ theo dõi số bytesLoaded và bytesTotal liên tục mỗi Frame. Biến percentLoaded có giá trị từ 0 đến 100. Và nó được sử dụng chính để thay đổi _xscale của thanh Bar. Bạn còn nhớ là ta đã mặc định điểm reg. point của thanh Bar này ở góc bên trái ko? Chính vì vậy mà thanh bar sẽ dài ra theo giá trị của biến percentLoaded sang phía bên phải
Khi mà số bytesLoaded = số bytesTotal thì display text sẽ hiện thông báo: "Loading Complete" và chuyển movie sang frame tiếp theo.

CODE
```
onClipEvent(enterFrame) {
  // if there is more to load
```

```
  if (bytesLoaded < bytesTotal) {

    // get current amount loaded
    bytesLoaded = _root.getBytesLoaded();

    // calculate percentage
    percentLoaded = Math.round(100*bytesLoaded/bytesTotal);
    // if there is still more
    if (bytesLoaded < bytesTotal) {

      // display text
      displayText = "Loading: "+percentLoaded+"%";

      // set scale of bar
      barFill._xscale = percentLoaded;

    // no more left
    }  else {

      // display complete
      displayText = "Loading Complete.";

      // fill out bar
      barFill._xscale = 100;

      // go to next frame
      _root.nextFrame();
    }
  }
}
```

Ko biết hướng dẫn như trên các bạn đã hình dung ra cách làm chưa nhỉ. Nếu các bạn chưa hiểu thì cbt sẽ post thêm hình minh họa vào

Timeline của chúng ta tôt nhất nên phân ra như sau nha:
-Timeline chính sẽ gồm có 3 layer. Layer thứ nhất sẽ chứa cái loader bar mc mà chúng ta đã tạo. Nó sẽ kéo dài trong 2 frames.

-Layer thứ 2 sẽ gồm có 2 keyframes ở fr1 và fr2. Ở kf 1 là lệnh stop();
Layer thứ 2 ta sẽ cho 1 button. Người sử dụng sẽ click vào button để xem tiếp phần còn lại của movie.
Trong btn cho đoạn code sau:

CODE
```
on (release) {
  play();
}
```

-Layer thứ 3 bắt đầu từ frame thứ 3 sẽ chứa cái movie cần load của bạn

Các bạn có thể xem thêm ở file : **23complexLoader.fla**

Có những lúc bạn cần tạo các movie lớn có các media thì ta ko cần phải tạo 1 movie đầy những file media trong đó mà có thể load các file media có sẵn này từ bên ngoài. Nhờ vậy bạn có thể xây dựng 1 trình diễn lớn bằng cách sử dụng những file bên ngoài.
Để làm được điều này chúng ta sẽ tìm hiểu các bước sau:

Thế chỗ movie hiện thời

Cách đơn giản để làm điều này là bạn chia cắt movie này ra thành các phần riêng. Khi 1 movie kết thúc ta có thể chuyển đến 1 movie khác. Tất cả những gì bạn cần là dùng lệnh **loadMovie** 
Ví dụ, bạn có 1 frame ở cuối 1 movie dài. Khi người xem đến đó, họ có thể click vào 1 button và xem 1 movie khác. Đoạn code đơn giản như sau:

CODE
```
on (release) {
  loadMovie("animation2.swf");
}
```

Hoặc là bạn có thể cho người sử dụng lựa chọn animation mà họ muốn xem tiếp. Ở cuối movie sẽ có 2 buttons chứa các movie khác nhau. Việc tạo liên kết giữa các movie là rất quan trọng sao cho khi user có thể trở lại movie cũ ban đầu.
Bạn có thể xem ví dụ ở file 23movie1.fla để biết thêm làm thế nào để trình bày tác phẩm của bạn với các files có sẵn.

Loading a Movie Clip

Với lệnh loadMovie bạn có thể thay thế chỗ của 1 mc bằng 1 mc khác. Ví dụ để thay thế chỗ của myMovieClip bằng file otherMovie.swf bạn chỉ cần làm:

CODE
```
myMovieClip.loadMovie("otherMovie.swf);
```

Khi sử dụng loadMovie bạn có thể dùng **getBytesTotal and getBytesLoaded** functions để thông báo cho người dùng bằng text hoặc bằng progess bar mà chúng ta đã làm ở trên quá trình load.

Nếu muốn preload một movie clip để sẵn sàng lúc cần hiển thị, chúng ta sẽ tạo một movie trống, không có gì ở trong ngoại trừ một câu lệnh **stop()**. Sau đó chúng ta load mc vào mc trống này (ko hiển thị trên stage). Khi load hoàn toàn thì movie của chúng ta đã sẵn sàng ở frame đầu tiên.

Movie file này sẽ nằm sẵn sàng trong browser cache của user. Bây giờ khi đến đoạn có sử dụng mc này thì lệnh loadMovie sẽ làm việc. Lúc đó sẽ nhanh hơn vì file đã được download hoàn toàn về rồi. Sau đó sử dụng lệnh gotoAndPlay(2) để qua frame 1.

Loading a JPEG

myMovieClip.loadMovie("picture.jpg");

Flash MX cũng cho phép ta khả năng load 1 file JPEG ở ngoài vào.
Cách làm tương tự như cách chúng ta load movie ở trên, chỉ cần thay đổi địa chỉ movie bằng địa chỉ của file JPEG là ok:

CODE
```
myMovieClip.loadMovie("picture.jpg");
```

mc myMovieClip bây giờ được thay thế chỗ bằng 1 mc có chứa bitmap image này.
Bạn có thể kiểm tra và xem ví dụ ở file 23loadipeg.fla

Loading a Sound

Có 2 cách để play 1 sound từ 1 file bên ngoài. Cả 2 đều sử dụng sound object và lệnh loadSound. Các sound file này cần ở dạng phổ biến là mp3.
Sau đây là 1 ví dụ cho cách thứ 1, để chơi 1 event sound. Ở đây toàn bộ sound sẽ được load vào bộ nhớ trước và sau đó được chơi nếu như có lệnh **start();**

CODE
```
on (release) {
  mySound = new sound();
  mySound.loadSound("mysound.mp3",false);
  mySound.start();
}
```

Flash sẽ ghi nhớ là lệnh start đã được đưa ra thậm chí khi sound mới chỉ bắt đầu download. Khi sound được load xong thì nó sẽ được play ngay lập tức

CODE
```
on (release) {
  mySound = new sound();
  mySound.loadSound("mysound.mp3",true);
}
```

Cách thứ 2 là ta sử dụng **true** ở param thứ 2. Giá trị true này sẽ bảo với flash đó là 1 stream sound. Ngay khi sound được load phần nào thì sẽ bắt đầu chơi ngay trong lúc phần còn lại vẫn tiếp tục được load. Nếu người dùng có kết nối mạng tốt thì sẽ nghe được toàn bộ sound khi load.

Ghi chú là bạn ko cần phải sử dụng lệnh start với 1 streaming sound. Tuy nhiên bạn

cần phải chú ý khi sử lí file MP3. Ví dụ nếu bạn sử dụng file nhạc mp3 với 128 Kbps hay 160 Kbps thường dùng để nghe thì nó sẽ là 1 file quá lớn để có thể stream với internet, đặc biệt nếu người sư dụng dùng modem. 32 Kpsb hay ít hơn sẽ thích hợp hơn khi ta sử dụng stream.

Chúng ta tổng kết lại nhe:

Streaming là 1 cách rất tốt giúp chúng ta sử dụng dễ dàng hơn với 1 movie lớn. Ta có thể sư dụng AS để kiểm tra 1 quá trình loading. Bạn có thể giữ movie ở frame thứ nhất và chỉ để nó tiếp tục khi toàn bộ movie đã được load.

Bạn cũng có thể thông báo cho người dùng quá trình loading bằng text thông báo số % hoặc 1 bằng 1 progess bar chẳng hạn.

Bạn có thể chia movie ra thành cách file riêng và sử dụng loadMovie để nhảy từ file này tới file khác như người sử dụng muốn.

Các file bên ngoài có thể được load vào bằng các cách khác nhau. Ngoài movie bạn có thể load 1 file ảnh, file nhạc nữa.

Một số câu hỏi và trả lời sau có thể giúp bạn nắm rõ bài hơn:

Câu 1: ta có thể load 1 mc, vậy có thể unload chúng ko??

-Trả lời: có, bằng lệnh unloadMovie

Câu 2: bình thường thì flash cần bao lâu để load trước khi nó start??

-Trả lời: ngay frame đầu tiên. Bởi vậy bạn cần sử dụng lệnh stop nếu như muốn nó đợi trước khi tiếp tục.

Câu 3: 2 cách để xác định khi 1 movie được load hoàn toàn??

-Trả lời: cách 1 dùng getBytesLoaded == getBytesTotal function
cách 2 dùng _frameLoaded property và _totalFrames property.

Ngoài ra bạn có thể xem thêm các bài viết sau:
Cách tạo 1 preloader đơn giản
http://www.vnfx.com/ipb/index.php?showtopic=2325

Cách tổng quát để tạo 1 loader đẹp:
http://www.vnfx.com/ipb/index.php?showtopic=2921

Ngoài ra về loadMovie, loadSound cũng có rất nhiều bài. Bạn chịu khó search ha

Giờ thứ 24: Vẽ với AS, Hour 24. Drawing with ActionScript

Đây là h cuối cùng, cũng là giờ rất thú vị, hy vọng các bạn cảm thấy dzui dzẻ khi vẽ bằng AS  . CBT ko có kinh nghiệm dịch bài, với lại nhìn vào mấy bài text dày đặc

chữ là tối mắt tối mũi nên nhiều chỗ dịch lung tung, theo ý thích bởi vậy có chỗ nào diễn đạt tối nghĩa, ko đúng thì các bạn cứ thẳng thắn góp ý, đừng thương tiếc .

Trong giờ này chúng ta sẽ học cách:

- **Vẽ đường thẳng và đường cong**
- **Tô màu 1 vùng**
- **Để người sử dụng vẽ với chuột**
- **Đặt các hình đã vẽ trong 1 movie clip mới**
- **Tạo các text fields**

1a.Drawing lines

-Để vẽ 1 đường thẳng, việc đầu tiên cần làm là định nghĩa các giá trị của lineStyle, nó dày như thế nào nè, rùi có màu gì và độ alpha.

CODE
```
lineStyle(thickness, color, alpha);
```

Màu (color): giá trị được đưa ra dưới dạng số thập lục phân hexa, ví dụ: 0x000000 là màu đen, 0xffffff: trắng , chúng ta có thể nhìn vào bảng color mixer để biết thêm
Độ trong suốt(alpha): min = 0; max=100;
Độ dày (thickness): độ dày nhỏ nhất là 0, còn lớn nhất là bao nhiu thì cbt ko biết .
Với độ dày 1 cái line có độ dày là 1 pixel, còn nếu ta cho giá trị là 0 (hairline) thì nó vẫn có độ dày là 1 pixel . Tuy nhiên chúng khác nhau ở chỗ: với hairline nếu như chúng ta thay đổi scale của nó thì độ dày của nó vẫn ko thay đổi. Các bạn thử đoạn code sau, và thay đổi giá trị độ dày nhé:

CODE
```
lineStyle(0,0x000000,100);
moveTo(20,50);
lineTo(200,200);
_xscale=300;
_yscale=300;
```

Nhìn đoạn code trên ta thấy xuất hiện lệnh:

CODE
```
moveTo(20,50)
```

Để vẽ 1 đường thẳng ta cần xác định 2 điểm: đầu và cuối. Mỗi điểm lại được xác đinh với 2 giá trị x và y
---> moveTo là điểm đầu, điểm đặt bút. Như vậy ở câu trên ta đặt bút tại x=20; y=50.

----> lineTo là điểm tiếp đến. Ta có x= 200; y=200. Như vậy chúng ta đã vẽ được đường thẳng nối từ điểm (20,50) đến (200,200).

Chú ý:

1. nếu như các bạn ko nêu điểm moveTo thì nó tự mặc định là điểm (0,0)
2. nếu như các bạn vẽ thêm các line tiếp theo với lineTo thì điểm đặt bút được mặc định là điểm cuối cùng của line trước.

Ví dụ để vẽ 500 đường lung tung trên màn hình ta có đoạn code đơn giản sau:

CODE

```
// set line style
lineStyle(2,0x000000,100);

// draw 500 lines
for(var i=0;i<500;i++) {

  // pick random start point
  x1 = Math.random()*550;
  y1 = Math.random()*400;

  // pick random end point
  x2 = Math.random()*550;
  y2 = Math.random()*400;

  // move to start point
  moveTo(x1,y1);

  // draw to end point
  lineTo(x2,y2);
}
```

Bây giờ các bạn thử play trò này xem sao, thử thay đổi độ alpha, color or thickness của nó 

Mọi người thử copy and paste đoạn code sau xem hiện ra gì nào  :

CODE

```
// set line style
lineStyle(2,0x999999,100);

for(var x=-400;x<550;x+=10) {
  // draw diagonal strip from left to right
  moveTo(x,0);
  lineTo(x+400,400);

  // draw opposite strip
  moveTo(550-x,0);
  lineTo(550-x-400,400);
}
```

1b.Drawing Curves

Tương tự khi bạn đã biết cách vẽ line rồi thì vẽ curveTo ko có gì là khó. Nó chỉ có thêm 1 chút giá trị thôi. Ta có đoạn code:

CODE
```
lineStyle(3,0x000000,100);
moveTo(150,200);
curveTo(275,275,400,200);
```

Như đã biết moveTo là điểm bắt đầu, anker1 ha.
curveTo(control_x, control_y, anker2_x, anker2_y);
Control_point là tiếp tuyến của 2 điểm anker với đường cong
anker2 là điểm cuối cùng của curve.
Các bạn có thể đọc thêm 1 số bài viết trong diễn đàn về đường cong bezier để hiểu rõ thêm  .

Chúng ta play tiếp với đoạn code sau:

CODE
```
lineStyle( 1, 0x0000FF, 100 );
moveTo(200,200);
curveTo(250,200,300,200);
curveTo(300,250,300,300);
curveTo(250,300,200,300);
curveTo(200,250,200,200);
```

Olala, tại sao chúng ko hiện ra các curve mà lại là 1 hình vuông. Nguyên nhân là do chúng ta cho các giá trị control point ở giữa 2 điểm kia 
Hãy thử thay đổi 1 chút xem chúng ta có gì nào:

CODE
```
var bend = 42;
moveTo(200,200);
curveTo(250,200-bend,300,200);
curveTo(300+bend,250,300,300);
curveTo(250,300+bend,200,300);
curveTo(200-bend,250,200,200);
```

sử dụng biến bend để thay đổi control point ta được 1 hình trong hơi méo rùi  . Về phần này plz đọc thêm các topic trong box AS

2. Drawing Filled Areas

Trước hết để có thể tô màu cho 1 vùng ta cần phải vẽ 1 hình khép kín ha.
Sau đó dùng lệnh beginFill để vẽ:

CODE
beginFill(color of fill, alpha of fill)

Ví dụ đoạn code sau:

CODE
```
lineStyle( 3, 0x000000, 100 );
beginFill( 0xFF0000 );
moveTo(175,100);
lineTo(375,100);
lineTo(375,300);
lineTo(175,300);
lineTo(175,100);
endFill();
```

Nếu như 1 vùng được cắt bởi line lần thứ 2 thì ở chỗ đó nó sẽ ko có màu nữa. Dựa vào đó ta có thể tạo nên 1 số hình thú vị  . Bạn thử copy đoạn code sau xem nó hiện ra cái gì nào và chỗ nào được tô màu, chỗ nào ko  :

CODE
```
lineStyle(3,0x000000,100 );
beginFill(0xFF0000);
moveTo(250,50);
lineTo(308,230);
lineTo(155,120);
lineTo(345,120);
lineTo(192,230);
lineTo(250,50);
endFill();
```

Có cách khác tốt hơn để vẽ 1 star, nó cho phép chúng ta qui định các giá trị ban đầu. Bạn thử xem file fla 24betterstar.fla xem sao 

Pause 

3.Vẽ với chuột

Phần tiếp theo trong sách này có đoạn code để user vẽ với chuột, thành thật mà nói, nếu bạn đã đọc đoạn code của DS post bên phần Flash hack thì bạn sẽ thấy nó khác nhau 1 trời 1 vực như thế nào http://www.vnfx.com/ipb/index.php?showtopic=3007
Nếu bạn đã đọc đoạn code của DS thì đừng nên đọc tiếp đoạn code hướng dẫn trong

cuốn sách này làm gì. CBt thấy là ko cần thiết phải dịch chỗ này nhưng trót rùi nên tôn trọng quyển sách, cbt dịch cho đầy đủ vậy 

1.Tạo 1 shape đơn giản, convert to mc.
2.Copy đoạn code sau vào mc (cbt đã phân tích kèm)

CODE

```
onClipEvent (load) {
  // cho các giá trị cho kiểu line
  _root.lineStyle(0, 0x000000, 100);
}

//khi chuột được dzí xuống thì bắt đầu vẽ
onClipEvent(mouseDown) {
  // ok to draw
  draw = true;

  // xác định điểm start để vẽ
  startX = _root._xmouse;
  startY = _root._ymouse;
  _root.moveTo(startX,startY);
}

//khi thả chuột ra thì ko vẽ nữa
onClipEvent(mouseUp) {
  // don't draw anymore
  draw = false;
}

//bắt đầu
onClipEvent (enterFrame) {
  if (draw) {

    //lấy vị trí hiện tại của chuột
    newX = _root._xmouse;
    newY = _root._ymouse;

    //nếu như vị trí khác với vị trí ban đầu
    if ((newX != startX) or (newY != startY)) {

      //vẽ 1 line tới vị trí mới
      _root.lineTo(newX,newY);

      // reset location for new time
      startX = newX;
      startY = newY;
    }
  }
}
```

Phần này chỉ có vậy

4.Tạo 1 movie clip riêng cho các hình đã vẽ

Những gì chúng ta đã làm ở trên chỉ là vẽ 1 cách đơn giản trên stage, nó có bất lợi là có thể bị bất kì mc nào che khuất. ko thể di chuyển.
Khi chúng ta đặt những cái này vào trong 1 mc thì thuận tiện hơn rất nhiều. ta có thể thay đổi _x, _y, rotation, alpha, scale ....

Để tạo 1 new mc ta dùng lệnh :
CODE
my_mc.createEmptyMovieClip("tên của new mc", level của nó)

Ví dụ :

CODE
this.createEmptyMovieClip("myMovieClip",1);
myMoveClip.lineStyle(0,0x000000,100);
myMoveClip.moveTo(100,100);
myMovieClip.lineTo(200,200);

Như vậy bạn đã có 1 mc mới tên là myMovieClip, level 1, có chứa các hình vẽ trên.

Nếu bạn muốn viết nhanh hơn thì có thể viết lại như sau:

CODE
this.createEmptyMovieClip("myMovieClip",1);
with(mymovieClip){
lineStyle(0,0x000000,100);
moveTo(100,100);
lineTo(200,200);
}

Lệnh createEmptyMovieClip này còn có rất nhiều tác dụng khác như ta có thể attach thêm movie vào đó hay là sao chép (duplicateMovieClip)

Rùi bi giờ bạn hãy thử tạo 1 mc mới đi nào

Có 1 bài hướng dẫn vẽ bông tuyết khá đẹp trong cuốn sách, tuy nhiên code được chia nhỏ thành các function khá dài dòng và mất công. trong khi trong diễn đàn của chúng ta có nhiều đoạn code về làm tuyết hay hơn rất nhiều: bài thế trời làm tuyết của DS trong box hướng dẫn thực hành. Đặc biệt hợp với chủ đề này là bài dùng 100% bằng AS của Raider (tìm trong mục lục), trong đó có đoạn code rất hay. Bởi vậy cbt sẽ ko dịch về đoạn code trong sách này nữa.

Để biết thêm chi tiết xin xem file fla 24snowflakes.fla 

5.Textfield

Đây là phần cuối cùng, chúng ta sẽ học cách tạo 1 text field và sau đó là làm 1 effect flying words.

Tạo Text
Để tạo 1 text chúng ta cần tạo 1 ô text (text field) với lệnh :

CODE
```
createTextField("text_name", level , vị trí _x, vi trí _y, chiều rộng,chiều dài);
```

Sau đó là text hiện lên trong text field đó:

CODE
```
text_name.text = "Nội dung text";
```

Ví dụ đoạn code sau, mình phân tích trong đó luon cho nhanh:

CODE

```
//tạo 1 ô chữ tên my..., level 0, _x=0, _y=170, width=550, height=60
createTextField("myTextField",0,0,170,550,60);

//text
myTextField.text = "Welcome to VNFX";

// giá trị true là sử dụng font chúng ta add trong thư viện, false là ta sử dụng font mặc định
myTextField.embedFonts = true;

//mặc định các tính chất của text
myTextFormat = new TextFormat();

//loại chữ, cỡ, màu, vị trí
myTextFormat.font = "Arial";
myTextFormat.size = 48;
myTextFormat.color = 0x330000;
myTextFormat.align = "center";

//liên kết những tính chất này với ô text của chúng ta
myTextField.setTextFormat(myTextFormat);
```

Chú ý khi sử dụng embedFont ta cần phải chọn 1 font trong thư viện click vào góc phải trên cùng của thư viện, chọn font mà bạn muốn, sau đó click chuột phải, linkage,

export, IDname

Nếu bạn muốn tạo 1 Input text thì cần phải cho type của textField là input, sau đó có thể set thêm các variable để máy có thể nhận info từ user

Còn rất nhiều các tính chất trong textField và textFormat class, các bạn nên đọc thêm help để trang trí cho textField của mình

Lý thuyết chỉ có vậy, nào bây giờ chúng ta sẽ làm effect flying words
CBT ko dám nhận xét nhiều về đoạn code và cách làm trong cuốn sách này, tuy nhiên nó được chia nhỏ thành các function gây rắc rối, dài dòng cho người đọc, lại còn phải tạo thêm mc trên stage, có chỗ hạn chế nữa... Thôi thì cứ đưa ra vậy

CODE

```
function createText(n,text) {
  // create a new movie clip
  this.createEmptyMovieClip("text"+n,n);
  mc = this["text"+n];

  // set the text format
  myFormat = new TextFormat();
  myFormat.font = "Arial";
  myFormat.color = 0x000000;
  myFormat.size = 24;
  myFormat.align = "center";

  // create a new text field
  mc.createTextField("myTextField",1,-100,-20,200,40);
  mc.myTextField.text = text.toUpperCase();
  mc.myTextField.embedFonts = true;
  mc.myTextField.setTextFormat(myFormat);

  // return reference to this movie clip
  return(mc);
}

function createAllText(textList) {
  // loop through array of text
  for(var i=0;i<textList.length;i++) {

    // create movie clip with this text
    mc = createText(i,textList[i]);

    // set random location
    mc._x = Math.random()*450+50;
    mc._y = Math.random()*350+25;

    // set scale to nothing
```

```
      mc._xscale = 0;
      mc._yscale = 0;

      // set scale variable to negative amount
      mc.scale = 0-i*100;
   }
}

function init() {
   // create array of text
   var words = "Love,Peace,Destiny,Llamas,Fate,History,Cheese,Rainbows,Tiny
Rocks";
   var textList = words.split(",");

   // create all text movie clips
   createAllText(textList);

   // remember how many there are
   numWords = textList.length;
}
function moveText() {
   // loop through words
   for(var i=0;i<numWords;i++) {

      // increase the scale of this movie clip
      mc = this["text"+i];
      mc.scale += 10;

      // hide movie clip when scale is too big
      if (mc.scale > 300) {
         mc._visible = false;

      // set scale of movie clip to scale when it is a positive number
      }  else if (mc.scale > 0) {
         mc._xscale = mc.scale;
         mc._yscale = mc.scale;
      }
   }
}
init();
stop();
```

Sau đó bạn tạo 1 mc trên stage và nhét đoạn code sau vào:

CODE

```
onClipEvent(enterFrame) {
   _parent.moveText();
}
```

Cuối cùng vào thư viện, trong menu của thư viện ta chọn font ruì linkage, rùi export với tên Arial.

effect này ko có gì khác là ta tạo 1 array chứa các ô text, sau đó các text hiện ra từ từ và dần dần phóng to lên, đến 1 kích thước nào đó thì biến mất.
Để các text hiện ra từ từ ta chỉ cần mặc định scale ban đầu của text là ko, sau đó dùng thêm 1 var nữa. Dần dần phóng to lên thì ta thay đổi scale thôi.
Thấy đoạn code trên dài dòng quá nên cbt viết lại cho nó đơn giản bớt như sau

CODE

```
function a(){
wordArr=new Array("vnfx","flash","actionscript");
//ta tạo 3 mc chứa 3 text field
for(var n=0;n<wordArr.length;n++){
 _root.createEmptyMovieClip("text"+n,n)
 mc=_root["text"+n];
 myFormat=new TextFormat();
 myFormat.font="Arial";
 myFormat.color = 0x000000;
    myFormat.size = 24;
    myFormat.align = "center";

mc.createTextField("myTextField",1,-100,-20,200,40);
mc.myTextField.text = wordArr[n];
mc.myTextField.embedFonts = true;
mc.myTextField.setTextFormat(myFormat);

            //mặc định các giá trị ban đầu
mc._x=Math.random()*450+50;
mc._y=Math.random()*350+25;
mc._xscale=mc._yscale=0;
//biến để tạo sự xuất hiên từ từ
            mc.scale=0-n*100;

mc.onEnterFrame=function(){
 this.scale+=10;
 if (this.scale > 300) {
  removeMovieClip(this);
 } else if (this.scale > 0) {
  this._xscale=this._yscale = this.scale;
 }
}
}
}
a();
```

```
//cho lặp lại sau 5 s
setInterval(a, 5000);
```

Kết thúc rồi 
có thắc mắc kiện cáo gì các bạn cứ góp ý ha