

LECTURA 3

“CONDICIONALES, FUNCIONES, LISTENER Y EVENTOS” ACTION SCRIPT 3.0 FLASH CS5.5

LAS CONDICIONALES: Nos permiten controlar la lógica del programa, podemos decir que podemos “hablar” con el programa para decirle lo que tiene que hacer en determinadas ocasiones.

ESTRUCTURA: `if (condiciones) { acciones que se realizaran si se cumple la condición }`

Podemos escribir varios if para varias condiciones diferentes, por ejemplo si tenemos un numero podemos decirle al programa si el número es mayor a 4, si es menor esto, si vale 10 esto,etc.

También tenemos:

if (condición) { acciones_si_se_cumple; } else { acciones_si_no se cumple; }

Para poner dos condicionales (dos condiciones) se separan con && (condicional Y)
if (condicion 1 && condicion 2) se tienen que cumplir las dos condiciones.

Si queremos que se cumpla solo una de varias condiciones: Usamos condicional O (||)
if (condicion1 || condicion2 || condicion3) solo tiene que cumplir una de las 3 condiciones para realizarse las acciones de la condicional. Las dos barras rectas salen con Alt + 124.

Por último tenemos una condicional muy usada que es: La condición es si el objeto1 choca con el objeto 2 (colisiona).

if (Objeto1.hitTestObject(Objeto2)) { lo que hace si cumple la condición }

Ejercicio 12: Crea un numero aleatorio entre 0 y 50. Si el resultado es mayor de 25 que de salida se visualice, en un cuadro de texto, “El número es mayor de 25”, y si es menor que aparezca “Es menor de 25”. Si quieres saber el número ya sabes trace.

Ejercicio 13: Dibuja 2 objetos en el escenario y que uno de ellos se arrastre cuando pasa el ratón sobre él. Cuando el que se arrastra toque el segundo objeto este tiene que dejar de verse (alpha) y en un cuadro de texto aparecer la palabra “Colisión”.

LAS FUNCIONES: Son trozos del programa que realizan determinadas cosas. Por ejemplo una función que realiza una multiplicación, que mueve un objeto, etc. Una función se escribe de la siguiente forma:

ESTRUCTURA: `function NombredelaFuncion () :void {las acciones a realizar por la funcion}`

Ejemplo `function sumar ():void { var suma = variableA + variableB; nextFrame(); }`

Para que el programa entre en la función a realizar lo que tiene dentro se hace escribiendo simplemente el nombre de la función y el programa nos llevará a la función (en el caso anterior escribir sumar;). Lo más normal es que a una función se acceda desde la llamada de un listener.

LOS LISTENER: Los listener se utilizan para detectar que el usuario realiza alguna acción, diremos que ha sucedido un evento (pulsar un tecla, pinchar con el ratón, etc). Los listener escuchan el evento y cuando lo detectan envían al programa a una función, en la cual el programa realizará algo. Tienen la siguiente estructura:

```
Nombreinstancia.addListener (NombreDelEvento, FunciónLaQueNosLlevará)
```

Ejemplo: Objeto1.addListener (MouseEvent.MOUSE_DOWN, escriba)
function escriba (e:MouseEvent) :void {lo que hace la función}

En este listener cuando el ratón se pulse sobre el Objeto1, el listener nos lleva a la función escriba, y el programa realizará lo que haya dentro de esta función. Ojo cuando a una función se entra después de un evento (desde un listener) hay que poner dentro del paréntesis de la función el evento desde la que se le llama precedido de e: . En este caso (e:MouseEvent) .

Ejercicio 14: Realiza un programa con un cuadro de texto y que al pulsar el ratón sobre el escenario (stage será quien llama al listener no Objeto1) aparezca "Pulsamos El Ratón".

Los eventos más importantes son los realizados con el ratón MouseEvent y tenemos los siguientes:

MouseEvent.CLICK	= Se produce al hacer clic (pulsar y soltar el botón ratón).
MouseEvent.DOUBLE_CLICK	= Se produce al hacer doble clic.
MouseEvent.MOUSE_DOWN	= Se produce al pulsar el botón del ratón (antes de soltarlo).
MouseEvent.MOUSE_MOVE	= Se produce cuando el ratón se mueve sobre el elemento.
MouseEvent.MOUSE_OUT	= Se produce cuando se saca el cursor de encima del objeto.
MouseEvent.MOUSE_OVER	= Se produce cuando el ratón se coloca sobre el objeto.
MouseEvent.MOUSE_UP	= Se produce cuando se suelta el botón del ratón.
MouseEvent.ROLL_OUT	= Se produce cuando se mueve el ratón fuera del elemento.

Una acción que podríamos poner dentro de una función son los hipervínculos en objetos (por ejemplo al hacer click en un botón). Se escribe de la siguiente forma:

navigateToURL(new URLRequest("url"), "_blank"); Te lleva a la página web definida en mi URL. Blank significa que se abrirá en una nueva página. Si ponemos _self se abrirá en la misma ventana.

Ejemplo:

```
navigateToURL (new URLRequest ("http://www.jugarconjuegos.com"), "_blank");
```

Ejercicio 15: Crea un botón con hipervínculo al hacer click sobre él

Ejercicio 16: Crea un programa para una tienda donde podamos introducir el precio del artículo, pero que también le podamos aplicar un porcentaje de descuento. Al final nos tiene que dar el precio a pagar por el artículo, después del descuento.

Ejercicio 17: Crea un programa que haga una pregunta y se conteste la solución en un cuadro de texto de introducir texto. Dibuja un rectángulo (botón) que al pulsarlo salga en un cuadro de texto dinámico si la solución escrita es Correcta o Incorrecta.

Ejercicio 18: Crea un juego con 3 preguntas (cada pregunta en un fotograma diferente) y que además cada pregunta acertada sume un punto al jugador (usa variables).

Hay una categoría genérica de eventos que no pertenecen a ninguna en especial. Esta categoría se llama Event.

ENTER_FRAME Ejecuta la función a la que se llama, un número de veces igual al que tengamos definido la velocidad de nuestra película. Si es 24fps pues ira a la función que se llama 24 veces por segundo. El listener llamará a la función 24 veces cada segundo.

Se suele usar mucho para definir movimientos de objetos. En nuestro caso podríamos cambiar la posición del objeto cada 1/24 segundos.

Estructura: `stage.addEventListener(Event.ENTER_FRAME, movimiento);`

Ejercicio 19: Crea un objeto que se mueva por el eje x del escenario. Ojo cuando salga del escenario por la derecha que se coloque en la parte izquierda del escenario.

CONTROL DESDE EL TECLADO: Las teclas del teclado tienen cada una asignado un código numérico para identificarlas llamado keyCode. Así cuando queremos hacer algo con una tecla, para especificar la tecla debemos de poner su keyCode. Ejemplos:

Tecla derecha 39, Tecla izquierda 37, Tecla arriba 38, Tecla abajo 40; Tecla enter 13, escape 27, tecla espacio 32.

Lo mismo que los eventos del ratón se llamaban MouseEvent, los eventos para detectar cuando se ha pulsado una tecla del teclado se llaman KeyboardEvent y por ejemplo para detectar que una tecla se pulsa sería el siguiente listener

Estructura: `NombreInstancia.addEventListener(KeyboardEvent.KEY_DOWN, funcion);`

KEY_DOWN sirve para detectar la pulsación de una tecla cualquiera.

Con este listener podemos llamar a una función que dependiendo el KeyCode que se haya pulsado haga una cosa u otra:

Ejemplo

```
Stage.addEventListener(KeyboardEvent.KEY_DOWN, posicion),
function posicion (e:KeyboardEvent):void {
    if (e.keyCode == 37)
    {objeto1.x += 3}
}
```

desplaza 3px en la posición x del objeto1 siempre que pulse la tecla izquierda

Ojo a veces hay que determinar que objeto del escenario esta seleccionado. A esto se le dice quien tiene el foco. Para determinar quién tiene el foco se pone de la siguiente forma:
`stage.stageFocusRect = false; //elimina el cuadro que esta alrededor del objeto`
`stage.focus = Objeto; //para decir que objeto tiene el foco`

Ejercicio 20: Crea un programa que mueva un objeto creado por ti con las teclas de dirección