

ACTIONSCRIPT (AS)

Proyectos II. Audiovisuales. Dpto. Escultura. UPV.

[sintaxis elemental]

AS

Action script es el lenguaje de programación que lleva incorporado el software de creación multimedia Macromedia Flash.

Es necesario aprender AS, para realizar herramientas interactivas en FLASH.

- Mantiene un gran parecido al lenguaje de programación **JavaScript.**

AS

Action script es el lenguaje de programación que lleva incorporado Flash.

Es necesario aprender AS, para realizar herramientas interactivas en FLASH.

- Mantiene un gran parecido al lenguaje de programación JavaScript.

DIFERENCIAS: **JAVASCRIPT**



AS

- 1- **ActionScript** no admite objetos específicos de navegador como Documento, Ventana y Ancla.
- 2- **ActionScript** no admite completamente todos los objetos incorporados de JavaScript.
- 3- **ActionScript** no admite algunas construcciones sintácticas de JavaScript,
- 4- **ActionScript** es un lenguaje de compilación.

AS

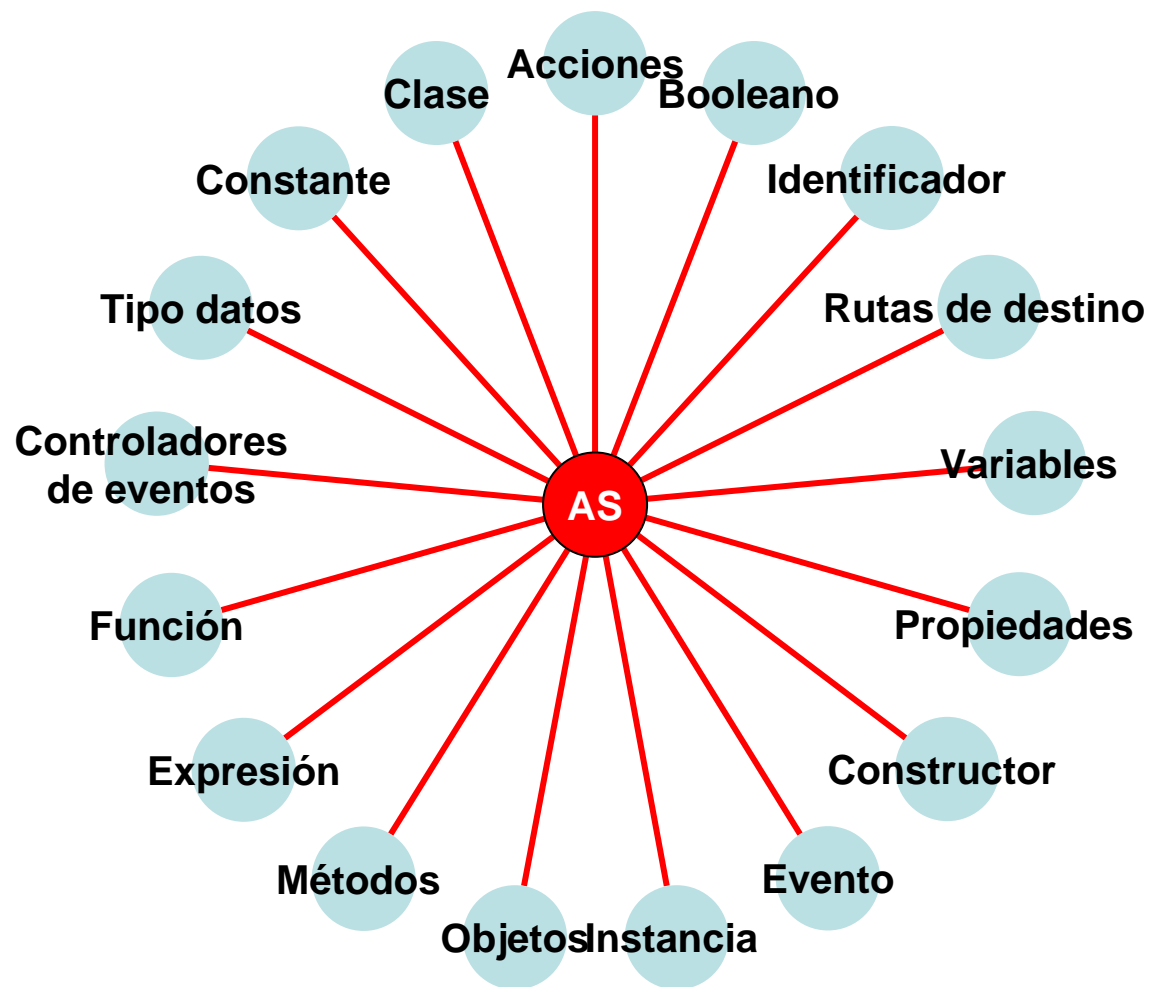
Terminología

Como con cualquier otro lenguaje de **creación de scripts**,
ActionScript utiliza su **propia terminología**

AS

Terminología

Como con cualquier otro lenguaje de **creación de scripts**,
ActionScript utiliza su **propia terminología**



AS

Sintaxis

- - Distinción entre mayúsculas y minúsculas

Myvalor = 100;

myvalor= 100;



Son dos variables
diferentes para AS

Esto es importantísimo para el uso de palabras clave, nombres de clase, variables, nombres de método, etc...

AS

Sintaxis

- - Distinción entre mayúsculas y minúsculas

Myvalor = 100;

myvalor= 100;



Son dos variables
diferentes para AS

Esto es importantísimo para el uso de palabras clave, nombres de clase, variables, nombres de método, etc...

NO UTILIZAR: En la declaración de variables, instancias, funciones, etc...

Ñ



CAÑA = 100;

ACENTOS



acción = 100;

NÚMEROS



112 = 100;

12hola= 100;

■ - LLaves

las definiciones de clase y las funciones de ActionScript se agrupan en bloques mediante llaves ({ }).

```
on(release) {  
    myDia = new Date();  
    esteMes =myDate.getMonth();  
} // fin del evento on(release)
```

■ - Punto y coma

Una sentencia ActionScript se termina con un punto y coma (;),

```
var mivariable = 0;
```

■ - Paréntesis

Al definir una función, tipo de evento, etc., los parámetros deben incluirse entre paréntesis

```
function myFuncion (edad, anyo){  
    // el código se escribe aquí  
}
```

- Comentarios

Es muy aconsejable añadir notas a los scripts utilizando comentarios

Para indicar que una línea o parte de una línea es un comentario, iniciar con dos barras inclinadas (//):

```
on(release) {  
    // crear un objeto Date nuevo ( esto no lo lee la aplicación como código)  
    myDate = new Date();  
    currentMonth = myDate.getMonth();  
    // convertir número de mes en nombre de mes  
    monthName = calcMonth(currentMonth);  
    year = myDate.getFullYear();  
    currentDate = myDate.getDate();  
} // fin de acción
```

Para indicar varias líneas (/ * ----- */):

```
/*  
    Esto es un comentario  
    Multilinea  
    De actionS  
*/
```

AS

Tipos de datos

Un tipo de datos **describe la clase de información que puede contener una variable** o un elemento de ActionScript

En action existen 2 tipos + 1:

- **PRIMITIVOS**
- **REFERENCIALES**
- **ESPECIALES**

En action existen 2 tipos:

PRIMITIVOS · REFERENCIALES · ESPECIALES

PRIMITIVOS

Los tipos de datos primitivos (**String**, **Number** y **Boolean**) tienen un valor constante, por consiguiente, pueden contener el valor real del elemento que representan. Las variables que contienen datos de tipo primitivo se comportan de modo diferente en ciertas situaciones que las que contienen datos de tipo referencia

En action existen 2 tipos:

PRIMITIVOS · REFENCIALES · ESPECIALES

PRIMITIVOS

Los tipos de datos primitivos (**String**, **Number** y **Boolean**) tienen un valor constante, por consiguiente, pueden contener el valor real del elemento que representan. Las variables que contienen datos de tipo primitivo se comportan de modo diferente en ciertas situaciones que las que contienen datos de tipo referencia

REFEREN.

Los tipos de datos de referencia (**MovieClip** y **Object**) tienen valores que pueden cambiar y, por consiguiente, contienen referencias al valor real del elemento

En action existen 2 tipos:

PRIMITIVOS · REFERENCIALES · ESPECIALES

PRIMITIVOS

Los tipos de datos primitivos (**String**, **Number** y **Boolean**) tienen un valor constante, por consiguiente, pueden contener el valor real del elemento que representan. Las variables que contienen datos de tipo primitivo se comportan de modo diferente en ciertas situaciones que las que contienen datos de tipo referencia

REFEREN.

Los tipos de datos de referencia (**MovieClip** y **Object**) tienen valores que pueden cambiar y, por consiguiente, contienen referencias al valor real del elemento

ESPECIALES

Null (NULO NO HAY VALOR)

Undefined (INDEFINIDO , NO SABE QUE VALOR ES)

PRIMITIVOS

- **STRING** Tipo de dato cadena. Una cadena es una secuencia de caracteres tales como **letras, números**. Las cadenas se introducen en una sentencia de ActionScript **entre comillas simples o dobles**.

Las cadenas se tratan como caracteres, no como variables.

Ej: “Curso” “2005” “G2”

- **NUMBER** El tipo de datos numérico. Es un **número entero** (ej:5) ó de **coma flotante**. (ej: 5,3)

- **BOLEANO** Un valor booleano puede ser **true** o **false**. ActionScript también convierte los valores true y false en **1** y **0**

Podríamos decir que son como interruptores.

REFERENCIAL

- **OBJECT** Un objeto es un conjunto de propiedades. Cada propiedad tiene un nombre y un valor. El valor de la propiedad puede ser cualquier tipo de datos de Flash, incluso el tipo de datos de objeto. Esto permite organizar unos objetos dentro de otros o *anidarlos*.

Ej: `empleo.semanaW.horasW`

- **MOVIECLIP** Los clips de película son símbolos que pueden reproducir animaciones en una aplicación Flash.

El tipo de datos MovieClip **permite controlar los símbolos de clips de película** mediante los métodos de la clase MovieClip.

Podemos llamar a los métodos mediante el operador punto (.),

Ej: `mi_bolita.startDrag(true);`

TIPOS DE OPERADORES PRINCIPALES:

- Operadores **numéricos (X)**
- Operadores de **comparación (X)**
- Operadores de **cadena (X)**
- Operadores **lógicos (X)**
- Operadores **en modo bit**
- Operadores **de igualdad (X)**
- Operadores **de asignación (X)**

(X) – Más usados

- Operadores **numéricos**

Los operadores numéricos realizan sumas, restas, multiplicaciones, divisiones y otras operaciones aritméticas.

+ Suma

* Multiplicación

/ División

% Módulo (resto de la división)

- Resta

++ Incremento

-- Decremento

- Operadores de comparación

Los operadores de comparación comparan los valores de las expresiones y devuelven un valor booleano (true o false).

< Menor que

> Mayor que

<= Menor o igual que

>= Mayor o igual que

- Operadores de cadena

El operador + tiene un efecto especial en las cadenas:
concatena los dos operandos de cadena

"¡Hola " + "mundo!"

- Operadores de cadena

El operador `+` tiene un efecto especial en las cadenas:
concatena los dos operandos de cadena

`"¡Hola " + "mundo!"`

- Operadores de lógicos

Los operadores lógicos comparan los valores booleanos (true y false) y devuelven un tercer valor booleano. Por ejemplo, si ambos operandos son true, el operador lógico AND (`&&`) devuelve true.

`&&` AND lógico

`||` OR lógico

`!` NOT lógico

Ej: `if (i > 10 && h < 40){
play();
}`

- Operadores en modo bit

Los operadores en modo bit manipulan internamente los números de coma flotante para cambiarlos por enteros de 32 bits.

& AND en modo bit

| OR en modo bit

^ XOR en modo bit

~ NOT en modo bit

<< Desplazamiento a la izquierda

>> Desplazamiento a la derecha

**>>> Desplazamiento a la derecha con relleno con
ceros**

- Operadores de igualdad

Podemos utilizar el operador de igualdad (==) para determinar si los valores o las identidades de dos operandos son iguales.

Esta comparación devuelve un valor booleano (true o false).

Si los operandos son cadenas, números o valores booleanos, se comparan por su valor. Si los operandos son objetos o matrices, se comparan por referencia

==	Igualdad
===	Igualdad estricta
!=	Desigualdad
!==	Desigualdad estricta

- Operadores de asignación

Puede utilizar el operador de asignación (=) para asignar un valor a una variable, como se muestra en el ejemplo siguiente:

```
var password = "Sk8tEr";
```

También puede utilizar el operador de asignación para asignar valores a diversas variables en la misma expresión.

En la siguiente sentencia, el valor de a se asigna a las variables b, c y d:

```
a = b = c = d;
```

Presentación basada en: “Manual de Action Script ”
Macromedia Flash MX 2004.



Prof. Moisés Mañas Carbonell
Proyectos II - Audiovisuales
Dpto. Escultura. UPV
moimacar@esc.upv.es