Desarrollo de Aplicaciones Web

JavaScript. Introducción.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Pablo García Tahoces

JavaScript

- Es un lenguaje de programación, originalmente creado por *Brendan Eich* de la empresa *Netscape* para añadir interactividad a las páginas web abiertas con su navegador *Netscape Navigator 2.0*.

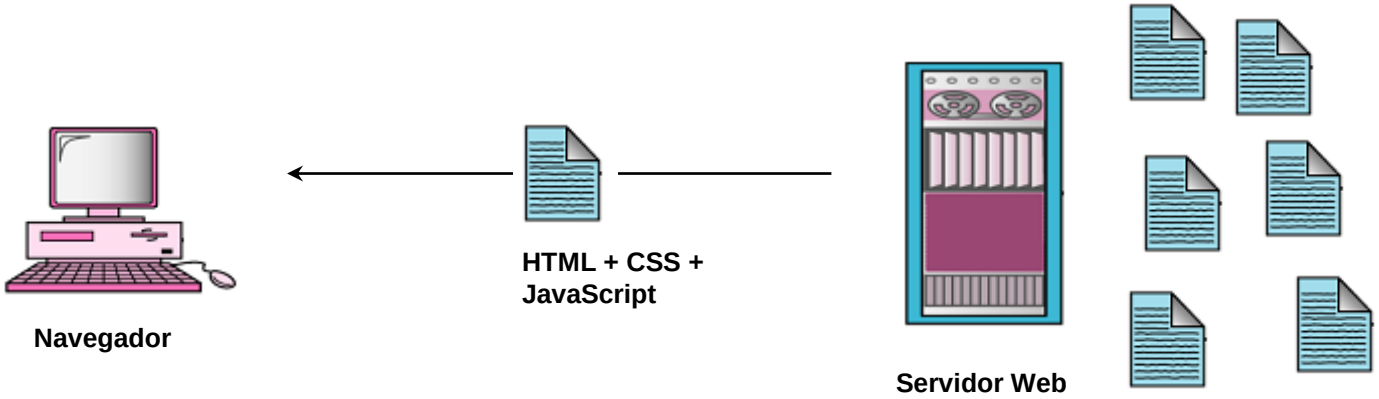
- El nombre original fue *Mocha*, siendo rebautizado posteriormente como *JavaScript*, mediante acuerdo con *Sun Microsystems*. No es *Java*.

- En 1997 comenzó el proceso de estandarización, primero en ECMA (*European Computer Manufacturers Association)* con el nombre *ECMAScript* y posteriormente también estándar ISO (*International Organization for Standardization*).

- Con su aparición se introduce el concepto de HTML dinámico, DHTML.

JavaScript

Javascript aparece embebido en el código HTML, siendo interpretado (no compilado) por el navegador (lado cliente) al descargar la página.



JavaScript

Microsoft desarrolló dos versiones propias para el IE:

- *JScript* → versión *ECMAScript* similar al *JavaScript* de *Netscape*, pero con algunas diferencias en el modelo de objetos.

- *VBScript* → incompatible con navegadores diferentes a IE y con sintaxis basada en el *Visual Basic*.

La diferencia en el modelo de objetos entre *JScript* y *JavaScript*, produjo incompatibilidades → W3C definió el *HTML DOM*, con el fin de aunar criterios.

JavaScript

Motores JavaScript:

Navegador	Script Engine
Apple Safari	Nitro
Google Chrome	V8
Microsoft IE	Chakra
Mozilla Firefox	JägerMonkey
Opera	Carakan

Integración con HTML

Normalmente se incluye en la cabecera, aunque no es obligatorio.

La dos únicas reglas son:

- Que debe ser incluido antes de ser usado.
- Que su contenido debe ir dentro de etiquetas <script>… </script>

Existe la posibilidad de incluirlo como página externa (fichero.js).

```
<script src="codigo.js"></script>
```

JavaScript. Sintáxis básica.

- Es sensible al uso de mayúsculas/minúsculas
- Es conveniente que todas las líneas de código finalizan en “;”.
- Los comentarios se hacen de dos formas:
  - “//” Para comentar líneas individuales.
  - “/* ….*/” Para comentar grupos de líneas.

JavaScript. Declaración de variables.

var mivariable;

El tipo de variable es el que toma al inicializarse.

Ej.- mivariable= “pepe”;
mivariable= 135;
mivariable= 135.0;

JavaScript. Tipos de datos.

- Existen cinco tipos de datos:
  - Cadenas de texto (string).
  - Valores numéricos (number).
  - Booleanos (boolean)
  - Objetos (Object)
  - Nulos (Null)

typeof(*mivariable*).- devuelve el tipo de variable.

JavaScript. Conversión entre tipos.

- Implícita.- la coerción es automática hacia cadenas.
- Explícita.- mediante funciones del tipo:
  - parseFloat()
  - parseInt()
  - toString()

JavaScript. Definición de funciones.

```
function mi_función ( )
{
        _________
        _________

        return( )
}
```


Pablo García Tahoces

JavaScript. Ámbito de las variables.

- Global.- accesible desde cualquier punto de la página.

```
<SCRIPT>
        var variableGlobal
</SCRIPT>
```

- Local.- accesible sólo dentro de la función.

```
<SCRIPT>
        function miFuncion (){
                var variableLocal
        }
</SCRIPT>
```

JavaScript. Operadores.

- Aritméticos.- +, -, *, /, ++, - -, \, %.
- Lógicos.- AND, OR, NOT.
- Relacionales.- ==, !=, >, >=, <, <=, ===, !==.
- Bit a bit.- &, |, ^, ~, <<, >>, >>>.
- Asignación.- =, +=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=, >>>=.
- Otros operadores.- + (concatenación), ?:, .

JavaScript. Operaciones matemáticas.

En general forman parte del objeto Math.

```
Ej.- resultado= Math.cos(37);
```

Utilizando with, podemos predeterminar el objeto.

```
Ej.- with(Math);
     resultado= cos(37);
```

JavaScript. Operaciones matemáticas.

Métodos incluidos en Math:

- log();
- exp();
- sqrt();
- pow();
- abs();
- floor();
- ceil();
- round();
- randon ();
- sin();
- cos();
- tan();
- asin();
- acos();
- atan();
- max();
- min();

JavaScript. Constantes predefinidas.

- E.- cte de Euler.
- LN2.- neperiano de 2.
- LN10.- neperiano de 10.
- LOG2.- $\log_2(e)$.
- LOG10.- $\log_{10}(e)$.
- PI.- número π.
- SQRT2.- raíz cuadrada de 2.
- SQRT1_2.- inversa de raíz cuadrada de 2.

JavaScript. Estructuras de control.

if - else

```
if (condición)
{       _____
        _____
}
else
{       _____
        _____
}
```

JavaScript. Estructuras de control.

for

```
for (valor inicial; condición parada; incremento)
{       _____
        _____
}
```

for-in

```
for (dato in Objeto.valor)
{       _____
        _____
}
```

JavaScript. Estructuras de control.

while

```
while (condición)
{       _____
        _____
}
```

do while

```
do
{       _____
        _____
}
while (condición);
```

JavaScript. Estructuras de control.

Salida de bucles:

- break; salida sin cumplir la condición.
- continue; ejecuta la siguiente vuelta, sin llegar al final del código.

JavaScript. Estructuras de control.

Switch-case

```
switch (expresión)
{
        case caso_1:
                ________
                ________
                break;
        case caso_2:
                ________
                ________
                break;
          .
          .
          .
        case caso_n:
                ________
                ________
                break;
}
```

JavaScript. Objetos.

Se realiza mediante la palabra clave new.

Constructor:

```
function crea_objeto(var_1, var_2, …, var_n);
```

Método:

```
function metodo_1(var_1, var_2, …, var_n);
```

Instanciación:

```
var objeto_1= new crea_objeto(var_1, var_2, …, var_n);
```

Llamada método:

```
objeto_1.metodo_1(var_1, var_2, …, var_n);
```

Acceso propiedades:

```
objeto_1.propiedad_1;
objeto_1[“propiedad_1”];
```

JavaScript. Arrays.

Se crean mediante new. Constituyen un objeto, con métodos propios y propiedades.

Ej.- var proveedor= new Array();
proveedor[1]= “Pepa”;
proveedor[2]= “Antonio”;

Propiedades y métodos:

- .length;
- .join();
- .reverse();
- .sort();
- .concat();
- .slice();

JavaScript. Manejo de cadenas.

Se realiza mediante el objeto string.

Métodos:

- .fromCharCode();
- .charCodeAt();
- .indexOf();
- .lastIndexOf();
- .substr();
- .slice();
- .substring();
- .toLowerCase();
- .toUpperCase();

JavaScript. Programación orientada a eventos.

Manejador de eventos.- función especial que forma parte del objeto que genera el evento y es llamada automáticamente por el navegador cuando la acción ocurre

Nomenclatura.- se añade el prefijo *on* al nombre del evento

Ej.- Click → onClick

```
<!DOCTYPE html>
<html>
    <head>
        <script>
            function miFuncion() {
                alert("La pagina se ha cargado");
            }
        </script>
    </head>
    <body onload="miFuncion()">
        <h1>Hola Mundo</h1>
    </body>
</html>
```

JavaScript. Programación orientada a eventos.

Eventos:

- Click.- pulsar con ratón
- MouseOver.- posicionarse encima con ratón
- MouseOut.- dejar de señalar con ratón
- Load.- cargar página
- Unload.- abandonar página
- Focus.- ventana pasa a activa
- Blur.- ventana deja de ser activa
- KeyPress.- presionar tecla
- Select.- seleccionar
- Change.- modificar
- Submit.- enviar formulario
- Reset.- inicializa formulario
- Error.- se produce un error
- Abort.- se aborta

JavaScript. Programación orientada a eventos.

Objeto	Manejador de evento
window	onload, onUnload, onBlur, onFocus
link	onClick, onMouseOut, onMouseOver
area	onMouseOut, onMouseOver
image	onAbort, onError, onLoad
form	onReset, onSubmit
test, textarea, password	onBlur, onChange, onFocus, onSelect
button, reset, submit, radio checkbox	onClick
select	onBlur, onChange, onFocus
Controles selección archivos	onBlur, onFocus, onSelect

JavaScript. Fechas y Horas.

Se realiza mediante el objeto date.

Dos constructores:

- new Date("mes día, año horas:minutos:segundos"
- new Date(año, mes, día, [horas, [minutos, [segundos, [milisegundos]]]])

Métodos:

- .getTime();
- .getDay();
- .getDate();
- .getMonth();
- .getYear();
- .getFullYear();
- .getHours();
- .getMinutes();
- .getSeconds();
- .getMiliseconds();
- .setTime();
- .setDay();
- .setDate();
- .setMonth();
- .setYear();
- .setFullYear();
- .setHours();
- .setMinutes();
- .setSeconds();
- .setMiliseconds();

JavaScript. Fechas y Horas.

Hora Universal Coordinada (UTC).- es la denomida hora GMT (Greenwich Meridiam Time). Utiliza métodos análogos a los anteriores +UTC.

Métodos:

- .getUTCTime();
- .getUTCDay();
- .getUTCDate();
- .getUTCMonth();
- .getUTCYear();

- .getUTCFullYear();
- .getUTCHours();
- .getUTCMinutes();
- .getUTCSeconds();
- .getUTCMiliseconds();

- .setUTCTime();
- .setUTCDay();
- .setUTCDate();
- .setUTCMonth();
- .setUTCYear();

- .setUTCFullYear();
- .setUTCHours();
- .setUTCMinutes();
- .setUTCSeconds();
- .setUTCMiliseconds();

Conversiones:

- .toUTCString
- .toGMTString