

- Scripting
  - Del lado del servidor
  - Del lado del cliente
    - Para modificar dinámicamente el contenido de un documento cuando éste es cargado
    - Para procesar las entradas de un form a medida que son ingresadas; validar los valores ingresados; completar una parte del form de acuerdo al valor de algún campo
    - Para ejecutar algún comportamiento cuando ocurre un evento tal como la carga o descarga del documento, foco de un control, movimiento del mouse, etc.
    - Para producir elementos de la interfaz con el usuario gráficos

- Elemento SCRIPT
  - Puede aparecer tanto en el head como en el body, cualquier cantidad de veces
  - Atributos
    - `scr` = URI  
Indica la ubicación de un script externo
    - `type` = content-type  
Indica el lenguaje de scripting en el que está escrito el contenido. Por ejemplo: "text/javascript"

- Elemento SCRIPT. Ejemplo

```
<HTML>
<HEAD>
<TITLE>Documento con SCRIPT</TITLE>
<SCRIPT type=" text/javascript"
  src="http://someplace.com/progs/vbcalc.js">
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT type="text/javascript">
...código JavaScript...
</SCRIPT>
</BODY>
</HTML>
```

- Elemento SCRIPT. Ejemplo

```
<HTML>
<HEAD>
<TITLE>Primer ejemplo de JavaScript</TITLE>
</HEAD>
<BODY>
<P><B>Esto es texto nornal de un documento HTML</B></P>
<P>
<SCRIPT LANGUAGE="JavaScript">
    document.write("Texto generado desde JavaScript")
</SCRIPT>
</P>
<P><B>Esto es de nuevo HTML</B></P>
</BODY>
</HTML>
```

Esto es texto normal de un documento HTML

Texto generado desde JavaScript

Esto es, de nuevo, HTML

- Valores
  - Numéricos
    - Enteros
      - Decimales: Del 0 al 9.
      - Octales: Comienzan por 0. Del 0 al 7.
      - Hexadecimales: comienzan por "0x" o "0X".  
Del 0 a 9 y a (o A) a f(o F).
    - Flotante
      - Un entero decimal, un punto decimal("."), una parte fraccionaria, un exponente ("e" o "E")
      - Un literal de coma flotante debe tener al menos un dígito y un punto decimal o una "e"(o "E")
      - 3.1415      -6.23E11      .2e10      2E-10
  - Lógicos
    - True y false

- Valores

- Strings

- Secuencia de caracteres encerrados entre " o '
    - Secuencia de escape: \

- \n                      \r              \t              \'              \"              \\\

- Objetos

- Conjunto de cero o más pares nombre:valor
    - Ejemplo:

- ```
avion={marca:"Boeing",modelo:"747",npasajeros:"450"}
```

- Para referirnos desde JavaScript a una propiedad del objeto avión:

- ```
document.write(avion.modelo)
```

- Valores

- Arrays

- Conjunto de 0 o más expresiones encerradas entre corchetes

- ```
coches= [ "BMW" , "Mercedes" , "Audi" , "Volvo" ]
```

- Coches es un array de 4 elementos.

- Se puede dejar elementos del array vacíos:

- ```
ciudades= [ "Madrid" , , "Valladolid" ]
```

- ```
ciudades= [ , , "Pamplona" ]
```

- ```
ciudades= [ "Madrid" , "Pamplona" , , ]
```

- Null

- Undefined

- Variables

- El nombre debe comenzar con un carácter o un \_
- Cuando no se le asigna valor, es undefined

```
var unValor = 50;
```

- El tipo de una variable puede cambiar

```
unValor = "Ahora es un string";
```



- Variables
  - Globales
    - Declaradas fuera de toda función
    - Cuando se le asigna valor el uso de `var` es opcional
  - Locales
    - El uso de `var` es obligatorio

```
function f1() {  
    return y-2;  
}  
f1() // error en tiempo de ejecución
```

```
function f2() {  
    return var y-2;  
}  
f2() // devuelve el valor NaN
```

- Operadores

- Aritméticos

- +    -    \*    /    % (resto)    - (unario)

- Notación prefija y postfija

- variable++ y ++variable

- Lógicos

- !    &&    ||    (short circuit)

- Strings

- "Hoy es un día" + "frío"

- resultado = "Día " + tiempo

- Operadores
  - Asignación

- `var resultado=50`

- `x += y`                      `x = x + y`

- `x -= y`                      `x = x - y`

- `x *= y`                      `x = x * y`

- `x /= y`                      `x = x / y`

- `x %= y`                      `x = x % y`

- Comparación

- > < >= <=

- == Devuelve true si son iguales. Fuerza conversiones de tipo

- != Devuelve true si son distintos. Fuerza conversiones de tipo

- === Devuelve true si son iguales y del mismo tipo

- !== Devuelve true si son distintos o de distinto tipo

- Conditional
  - `condición?acción1:acción2`

- Sentencias condicionales

- if

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

- Ejemplo

```
var i = 5  
if (i > 5) {  
    document.write("i es mayor que 5")  
}  
else{  
    document.write("i es menor o igual que 5")  
}
```

- Sentencias condicionales
  - switch

```
switch (variable) {  
    case valor1:  
        acciones1;  
        break;  
    case valor2:  
        acciones2;  
        break;  
    .....  
    .....  
    case valorN:  
        accionesN;  
        break;  
    default  
        acciones;  
}
```

- Sentencias iterativas

- for

```
for (inicio; final; incremento) {  
    acciones  
}
```

- do-while

```
do {  
    acciones  
} while (condición)
```

- while

```
while (condición) {  
    acciones  
}
```



- Funciones

```
function nombreFunción([parámetros]) {  
    acciones  
}
```

- return

- Parámetros

- arguments.length

- nombreFunción.arguments[i] (i >= 0)

### • Ejemplo

```
<form name="cálculo">
  Número: <input type="text" name="base">
  Potencia: <input type="text" name="elevado">
  Resultado: <input type="text" name="resultado">
  <input type="button" name="poten" Value="Calcular potencia"
    onClick="potencia()">
</form>

function potencia() {
  var i = 0;
  var resul = 1;
  if (document.cálculo.elevado.value == 0)
    document.cálculo.resultado.value=resul;
  else {
    resul = document.cálculo.base.value;
    for(i = 1; i < document.cálculo.elevado.value; i++)
      resul = resul * document.cálculo.base.value;
  }
  document.cálculo.resultado.value=resul;
}
```

- Funciones predefinidas
  - `getDate ( )` - Día del mes
  - `getDay ( )` - Día de la semana
  - `getMonth ( )` - Día del mes
  - `getFullYear ( )` - Año (con dos dígitos)
  - `getTime ( )` - Milisegundos transcurridos desde 1/1/1970
  - `getHours ( )` - Hora entre 0 y 23
  - `getMinutes ( )` - Minutos entre 0 y 59
  - `getSeconds ( )` - Segundos entre 0 y 59

- Funciones matemáticas

- `abs ( )` - Valor absoluto
- `max ( v1 , . . , vn )` - Valor máximo
- `min ( v1 , . . , vn )` - Valor mínimo
- `round ( )` - Redondear
- `exp ( )` - Exponencial
- `log ( )` - Logaritmo
- `pow ( base , exponente )` – Potencia (exp puede ser neg.)
- `sqrt ( )` - Raíz cuadrada
- `sin ( )` - Seno
- `cos ( )` - Coseno
- `tan ( )` - Tangente
- `asin ( )` - Arcoseno
- `acos ( )` - Arcocoseno
- `atan ( )` - Arcotangente

- Ventanas

- Alert

- Muestra el contenido de lo que recibe como parámetros, y un botón de aceptar.
    - `alert("Esto es una prueba");`

- Confirm

- Como "alert", pero muestra dos botones: aceptar y cancelar. Si se evalúa, al pulsar aceptar devuelve true y con cancelar, false.

- Prompt

- Muestra una ventana donde se puede escribir un valor, que es devuelto por la función.

- Objetos

- Entidad con propiedades (variables) que definen su estado y métodos (funciones) que actúan sobre esas propiedades

- Función constructora

```
function casa(localidad, superficie, precio) {  
    this.localidad = localidad  
    this.superficie = superficie  
    this.precio = precio  
}
```

- Alocación, instanciación

```
casal = new casa("Pamplona", 90, 15000000);  
casa2 = new casa("Bilbao", 110, 23000000);
```

- Desalocación

```
delete casal;
```

- Objetos

- Referencia a un atributo

- `casal.año = "1980"`

- Otra forma de crear un objeto

- `nombreobjeto = {propiedad1:valor, propiedad2:valor, ....., propiedadN:valor}`

- Subobjetos

- `casal = {localidad:"Pamplona", precio:15000000, superficie:{interior:90, terraza:10}}`
    - `casal.superficie.terraza = 20`

- Objetos

- Métodos

- Un método es una función asociada a un tipo de objeto

- Ejemplo

```
function VerCasa() {  
    var mostrar = "La casa está en " + this.localidad +  
    ", tiene " + this.superficie + " m2 y cuesta "  
    +this.precio + " ptas.";   
    return(mostrar),  
}
```

- y en la función constructora

```
this.VerCasa = VerCasa
```



- DOM: objetos predefinidos
  - Document
    - Link
    - Image
    - Anchor
    - Form
      - Textarea
      - Text
      - fileUpload
      - Password
      - Hidden
      - Submit
      - Reset
      - Radio
      - Checkbox
      - Button
      - Select
        - » Option

- Eventos

- Acción que el usuario realiza (click, posicionar con el ratón en un lugar determinado, enviar un formulario, seleccionar un cuadro de texto, ...)
- Cada evento podrá tener un “gestor del evento”. Para el evento “Click” el gestor se llama “onClick”

- Ejemplo:

```
<form ....>  
  <input type="button" onClick="función([parámetros])">  
</form>
```

### • Eventos posibles

- DragDrop - Arrastrar un objeto a la ventana del navegador
- Error - Error en la carga de un documento
- Focus - El usuario se posiciona en una ventana o cuadro de texto de un formulario
- KeyDown - Se pulsa una tecla
- KeyPress - Se pulsa o libera una tecla
- KeyUp - Se libera una tecla
- Load - Se carga un documento en el navegador
- MouseDown - Se pulsa un botón del ratón
- MouseMove - Se mueve el cursor
- MouseOver - El puntero del ratón se posiciona sobre un enlace
- MouseOut - El puntero del ratón sale de un enlace
- MouseUp - Se libera un botón del ratón
- Move - Se mueve la ventana. Esta acción también la puede realizar el script
- Reset - Se pulsa sobre el botón reset del formulario
- Resize - Las dimensiones de la ventana cambian
- Select - Se selecciona una de las opciones de un cuadro combo del formulario
- Submit - Se pulsa el botón submit del formulario
- UnLoad - El usuario sale de la página

- Ejemplo

```
<script>
  function contar(objetoSelect) {
    var seleccionadas = 0
    for (var i = 0; i < objetoSelect.options.length; i++) {
      if (objetoSelect.options[i].selected)
        seleccionadas++;
    }
    return seleccionadas;
  }
</script>
```

- Ejemplo

```
<form name="formulario">
  <p><b>Selecione los temas que son de su interés y pulse
  después el botón</b><br>
  <select name="temas" múltiple>
    <option selected>Informática
    <option>Naturaleza
    <option>Música
    <option>Deportes
    <option>Economía
    <option>Coleccionismo
  </select>
  <p>
  <input type="button" value="¿Cuántos hay seleccionados?"
  onclick="alert('Número de opciones seleccionadas: ' +
  contar(document.formulario.temas))">
</form>
```