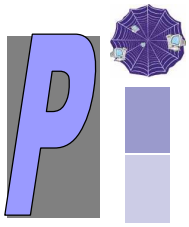


# Tema 3. JavaScript

**Contenido**

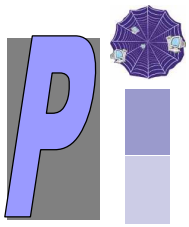
1. Introducción
2. Fundamentos de JavaScript
   - Tipos básicos y variables
   - Operadores
   - Estructuras de Control
3. Funciones y Objetos en JavaScript
   - Funciones
   - Objetos y Métodos
4. Eventos en JavaScript

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA, SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

**Referencias**

Libro: Capítulos 7-14. Internet & WWW: How to Program.
Capítulo 5 . Programación de Aplicaciones Web.

W3 Schools,
http://www.w3schools.com/js/



# Programación en Internet

## Tema 3. JavaScript

**Contenido**

**1. Introducción**
**2. Fundamentos de JavaScript**
- **Tipos básicos y variables**
- **Operadores**
- **Estructuras de Control**

**3. Funciones y Objetos en JavaScript**
- **Funciones**
- **Objetos y Métodos**

**4. Eventos en JavaScript**

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA, SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

## Introducción

- Características de JavaScript
  - Tecnología del lado del servidor que aporta dinamismo a documentos HTML
  - Lenguaje interpretado (script) embebido en el código HTML
  - Lenguaje orientado a eventos. Es posible ejecutar acciones en respuesta a eventos.
  - Lenguaje orientado a objetos. Aunque con un modelo de objetos reducido, JavaScript dispone de las ventajas de los paradigmas orientados a objetos.
- Historia de JavaScript
  - Netscape incluyó JavaScript 1.0 en su navegador Netscape Navigator 2.0 y Posteriormente JavaScript 1.1 en la versión 3.0
  - Microsoft incluyó su versión de JavaScript (Jscript) en Internet Explorer 3.0
  - ECMA (European Computer Manufacturers Association) estandariza JavaScript bajo la denominación ECMAScript-262
  - Desde las versiones de Netscape 4.0 e Internet Explorer 4.0 soportan el estándar ECMAScript-262 o JavaScrip 1.2 (aunque con diferencias)
  - La últimas versiones de los navegadores soportan JavaScript 1.3 (Full ECMAScript-262)



## Introducción

- JavaScript en un documento HMTL
  - <script … > … </script>
  - Atributos
    type : [ text/ecmascript | **text/javascript** | text/jscript | text/vbscript | text/vbs | text/xml ]
    src : URL al fichero (.js) que contiene el código JavaScript
  - Ubicación
    <head> … </head>. El código será ejecutado cuande se invoque.
    <body> … </body>. El código se ejecutará durante la carga de la página
  - Ejemplos

```
… <!-- Ejemplo 1 -->
<head> …
<script type="text/javascript"
        src="codigo.js"> </script>
… </head>
…
```

```
… <! -- Ejemplo 2 -->
<head> … </head>
<body> …
<script type="text/javascript" >
  document.write("<h1>Hola Mundo!</h1>");
</script>
 … </body>
```

## Introducción

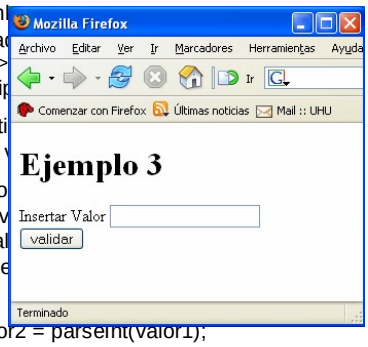
- Otro ejemplo

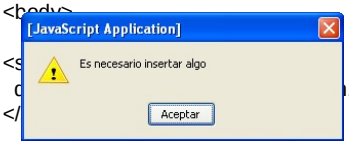
```
<html>
<head>
<title>JavaScript: Ejemplo 3</title>
<script type="text/javascript">

function validarNumero() {
  var valor1, valor2;

  valor1=document.f.n.value;
  if ( valor1=="" ) {
     alert("Es necesario insertar algo");
     return false;
  }

  valor2 = parseInt(valor1);
  if (isNaN(valor2))
       alert("Ha insertado la cadena: "+valor1);
  else alert("Ha insertado el número:
     "+valor1);
}
</script>
</head>

<body>

<script type="text/javascript">
  document.write("<h1>Ejemplo 3</h1>");
</script>

<form name="f" method="" action="" id="f">
Insertar Valor <input type="text" name="n"
      id="n">
<br />
<input type="button" value="validar"
      onclick="validarNumero()">
</form>


</body>
</html>
```

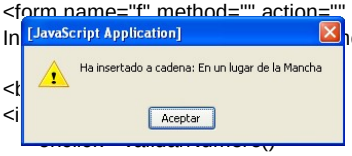


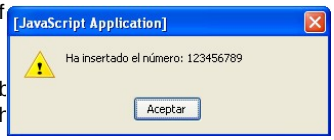
## Introducción

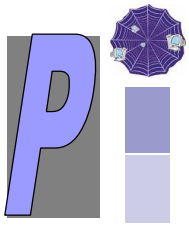
- Otro ejemplo





# Programación en Internet

## Tema 3. JavaScript

**Contenido**





DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA


## Fundamentos de JavaScript

- Tipos de datos y variables
  - Tipos básicos
    Numéricas: Enteros y Reales
    Booleanas.
    Cadenas de Caracteres.
  - Declaración de variables (no es necesario especificar un tipo, se asigna un tipo a la variable en la primera asignación)
    **var** *nomb_variable* **;**
  - Funciones de conversión de tipos
    **int parseInt(***cad***):** Devuelve el entero incluido en *cad* o NAN si no es numérica
    **float parseFloat(***cad***)**: Devuelve el real incluido en *cad* o NAN si no es numérica
  - Concatenación de cadenas y variables numéricas: +
    cadena + numero

    Ejemplo. “El valor de X es “ + x

## Fundamentos de JavaScript

- Ejemplo 4

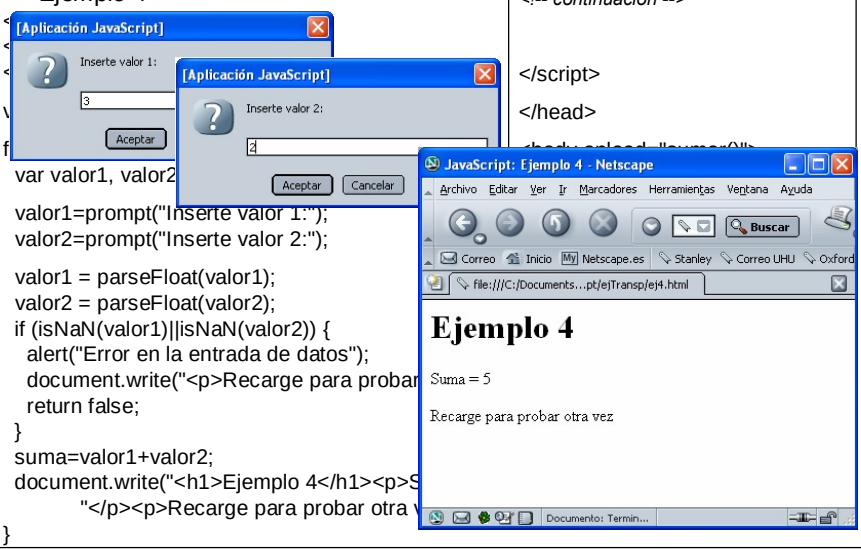
```
<html>  <head>
<title>JavaScript: Ejemplo 4</title>
<script type="text/javascript">

var suma;

function sumar() {
  var valor1, valor2;

  valor1=prompt("Inserte valor 1:");
  valor2=prompt("Inserte valor 2:");

  valor1 = parseFloat(valor1);
  valor2 = parseFloat(valor2);
  if (isNaN(valor1)||isNaN(valor2)) {
    alert("Error en la entrada de datos");
    document.write("<p>Recarge para probar otra vez</p>");
    return false;
  }
  suma=valor1+valor2;
  document.write("<h1>Ejemplo 4</h1><p>Suma="+suma+
          "</p><p>Recarge para probar otra vez</p>");
}
```

```
<!-- continuación -->


</script>

</head>

<body onload="sumar()">

</body>

</html>
```



## Fundamentos de JavaScript

- Ejemplo 4




```
<!-- continuación -->


</script>

</head>
```

## Fundamentos de JavaScript

- Operadores
  - Asignación

| Operador | Descripción |
|---|---|
| = | Asigna el valor del operando de la izquierda al operando de la derecha |
| += | Suma el operando derecho al izquierdo y le asigna el resultado |
| -= | Resta el operando derecho al izquierdo y le asigna el resultado |
| *= | Multiplica el operando derecho por el izquierdo y le asigna el resultado |
| /= | Divide el operando izquierdo entre el derecho, asignando el resultado al operando izquierdo |
| %= | Divide el operando de la izquierda por el de la derecha y asigna el valor del resto de la división al operando de la izquierda |

  - Ejemplo

```
a = 5;
x1 = x2 = x3 = 0;
b += a:
b %= 7;
```



## Fundamentos de JavaScript

- Operadores
  - Aritméticos

| Operador | Descripción |
|---|---|
| binarios | |
| + | Sumar |
| - | Restar |
| * | Multiplicar |
| / | Dividir |
| % | Resto división |
| unarios | |
| ++ | Incremento |
| -- | decremento |

Ejemplos

```
c = a+b;
d = (a + b) / 2;
resto =  a % 2;
area = lado * lado;
valor = 12 / ( 4 * 3 );
a++;
++b;
--c;
d--;
```

## Fundamentos de JavaScript

- Operadores
  - Comparación y Lógicos

| Operador | Descripción |
|---|---|
| == | Igual que |
| != | Distinto de |
| > (>=) | Mayor (igual) que |
| < (<=) | Menor (igual) que |
| && | Y (AND) |
| \|\| | O (OR) |
| ! | No |

| x | y | x&&y | x\|\|y | !x |
|---|---|---|---|---|
| F | F | F | F | T |
| F | T | F | T | T |
| T | F | F | T | F |
| T | T | T | T | F |

T: true , F:false

Ejemplos

```
if ( a == “ok”) …
if ( salir != 100 ) ..
while (x1 <= 10) …
for ( i=1; i<=10; i++) ...
if ( !condicion ) …
while ( x<10 && y<10) …
if (!a || (b>0 && c!=“si”)) …
```



## Fundamentos de JavaScript

- Estructuras de control: Condicionales
  - **if (***cond***) { *sentSi* }** → Si se cumple la condición se ejecuta *sentSi*
  - **if (***cond***) { *sentSi* }** → Si se cumple la condición se ejecuta *sentSi*
    **else {***sentSiNo* **}** en otro caso, se ejecuta *SentSiNo*
  - **( *cond* ) ? *valorSi* : *valorSiNo*** → Si se cumple la condición, devuelve *valorSi*
    en otro caso devuelve *valorSiNo*.
  - **switch (** exp **) {** → Se evalua el valor de la expresión *exp*
    **case** *val1* **:** *sent1***; break;** si toma el valor *val1* se ejecuta *sent1*
    **case** *val2* **:** *sent2***; break;** si toma el valor *val2* se ejecuta *sent2*
    …
    **case** *valN* **:** *sentN***; break;** si toma el valor *valN* se ejecuta *sentN*
    **default : ***sent***;** si toma otro valor se ejecuta *sent*.
    }

## Fundamentos de JavaScript

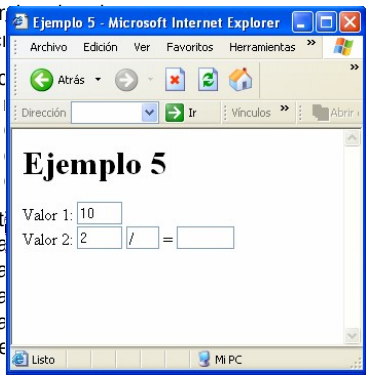
- Estructuras de control: Condicionales

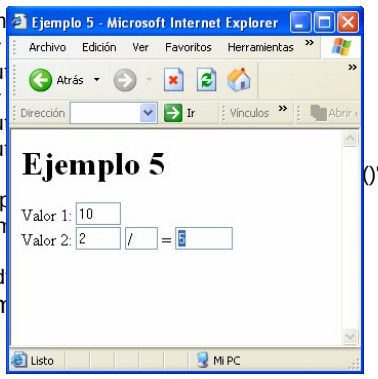
```
<html>  <head>
<script>

function operar() {
var resultado;
var operacion=f.op.value;
var op1=parseFloat(f.v1.value);
var op2=parseFloat(f.v2.value);

switch (operacion) {
  case ("+") :  resultado=op1+op2;break;
  case ("-") :  resultado=op1-op2;break;
  case ("*") :  resultado=op1*op2;break;
  case ("/") :  resultado=op1/op2;break;
  default    :  resultado="Error";
}
f.r.value=resultado;
}

</script>
<title>Ejemplo 5</title>
</head>

<body>

<h1>Ejemplo 5</h1>

<form id="f">
Valor 1:
<input type="text" id="v1" size="3"><br />
Valor 2:
<input type="text" id="v2" size="3">
<input type="text" id="op" size="1"
                                onblur="operar()">
= <input type="text" id="r" size="5">
</form>

</body>
</html>
```



## Fundamentos de JavaScript

- Estructuras de control: Condicionales




I. T. Informática de Gestión  Universidad de Huelva  Programación en Internet

## Fundamentos de JavaScript

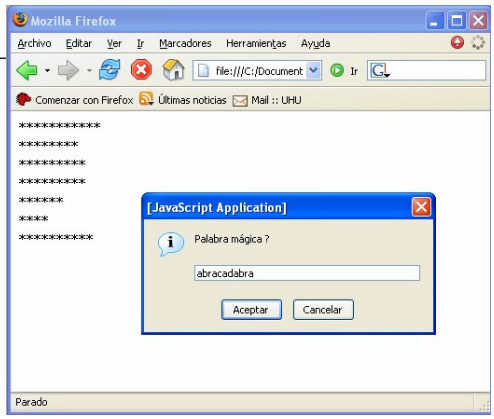
- Estructuras de control: Bucles
  - **for (***exp_inic ; cond ; exp_incr* **) { *sent* }**
    Se ejecuta la expresión de inicialización *exp_inic*. Se comprueba la condición *cond*, si es verdadera se ejecutan las sentencias *sent*. Posteriormente, se ejecuta la expresión de incremento exp_incr, y se vuelve a comprobar la condición, repitiendo el proceso hasta que ésta tome valor falso.
    Ejemplo: *for( i=0; i<100; i++) { document.write("Línea "+i+"<br />"); }*
  - **while (***cond***) { *sent* }**
    Las sentencias *sent* se ejecutaran mientras que se cumpla la condición *cond*.
    Ejemplo: *while ( dato!="Fin" ) { dato = prompt("Inserta dato); … }*
  - **do {***sent* **} while (***cond***);**
    Las sentancias *sent* se ejecutaran mientras la condición *cond* sea verdadera.
    Ejemplo: *do { dato = prompt("Inserta dato); … } while ( dato!="Fin" );*
  - **break** y **continue**
    **break** : finaliza la ejecución de un bucle.
    **continue** : Finaliza un ciclo de un bucle, comenzando uno nuevo.



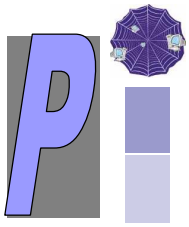
## Fundamentos de JavaScript

- Estructuras de control: Bucles

```
<html>
<head>
<title>Ejemplo 6</title>

<script>

function entrar() {
  var r;
  var i;
  do {
     r=prompt("Palabra mágica ?");
     if(r=="abracadabra") break;
     if(r.length==0) continue;

     for(i=0;i<r.length;i++)
            document.write("*");
     document.write("<br />");

  } while(true);
}
</script>
</head>
```




```
<!-- continuación -->

<body onload="entrar()">
</body>
</html>
```



# Programación en Internet

## Tema 3. JavaScript

**Contenido**





DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA


## Funciones y Objetos en JavaScript

- Funciones
  - Declaración

    **function** *nomb_funcion* **(** *lista_parametros* **) {**
    *sentencias;*
    **}**

  - Llamada: *nomb_funcion* **(** *valor_parametros* **);**

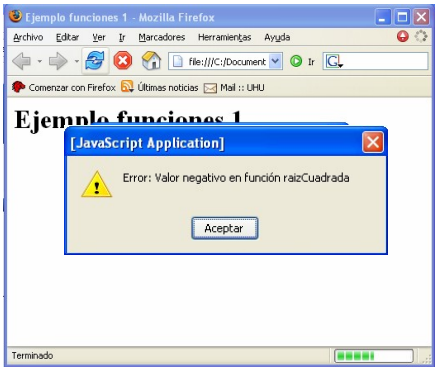
| **Funciones predefinidas** |
|---|
| isNaN(dato) |
| eval(expresion) |
| escape(cadena) |
| unescape(cadena) |

Ejemplo: fichero **func.js**

```
function aviso( error, funcion  ) {
    alert("Error: "+error+" en función"+funcion);
}

function raizCuadrada(x) {
    var dato;
    if ( x<0 ) aviso("Valor Negativo", "raizCuadrada");
    dato = Math.sqrt(x);
    return dato;
}
```

## Funciones y Objetos en JavaScript

- Funciones – Ejemplo 1




```
<html>
<head>
<script src="func.js"></script>
<title>Ejemplo funciones 1</title>
</head>

<body>

<h1>Ejemplo funciones 1</h1>

<script>

  var d=prompt("Dame valor","0");
  document.write("<p>Valor = "+raizCuadrada(d)+"</p>");

</script>

</body>
</html>
```



## Funciones y Objetos en JavaScript

- Funciones – Ejemplo 2


```
<html>  <head>
<script src="func.js"></script>
<script>
    function actuar() {
            var v=parseInt(f.dato.value);
            f.resultado.value=raizCuadrada(v);
    }
</script>
<title>Ejemplo funciones 2</title>  </head>
<body>
<h1>Ejemplo funciones 2</h1>

<form id="f">
Valor <input type"text" id="dato" /><br />
<input type="button" value="Raiz" onclicK="actuar()" />
Resultado <input type"text" id="resultado" />
</form>

</body>
</html>
```

## Funciones y Objetos en JavaScript

- Objetos y métodos
  - Creación

```
function nomb_objeto ( lista_atributos ) {
        this.atributo1 = atributo1;
        this.atributo2 = atributo2;
        ..
        this.metodo1 = funcion1;
        this.metodo2 = funcion2;
        …
}
```

  - Utilización : *objeto* **= new** *nom_objeto* **(** *valores_atributos* **);**

Ejemplo

```
funcion Calculo ( x, y ) {
        this.x = x;
        this.y = y;
        this.sumar = suma
        this.restar = resta
}
function suma { return this.x + this.y ; }
function resta  { return this.x + this.y ; }
```

```
…
o = new Calculo(4,4);
valor1 = o.sumar;
valor2 = o.restar;
…
```



## Funciones y Objetos en JavaScript

- Objetos y métodos: Ejemplo 1

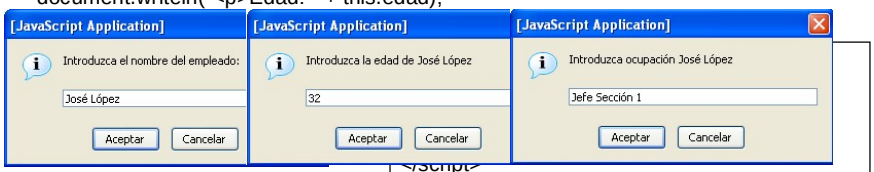
```
<html><head><title>Ejemplo objetos 1</title>
<script type="text/javascript">

function Empleado() {
   this.nombre = prompt("Introduzca el nombre del empleado: ", "Nombre");
   this.edad = prompt("Introduzca la edad de " + this.nombre, "00");
   this.puesto = prompt("Introduzca ocupación " + this.nombre, "Parado");
   this.mostrarPerfil = mostrarPerfil;
}
function mostrarPerfil() {
   document.write("<h1>Perfil del empleado</h1><hr />");
   document.write("<h2>" + this.nombre + "</h2>");
   document.writeln("<p>Edad: " + this.edad);
   document.writeln("<br />Puesto: " + this.puesto+"</p>");
}
</script>

</head>
```

```
<body>
<script type="text/javascript">
nuevoEmpleado = new Empleado()
nuevoEmpleado.mostrarPerfil()
</script>
</body></html>
```

## Funciones y Objetos en JavaScript

- Objetos y métodos: Ejemplo

```
<html><head><title>Ejemplo 7</title>
<script type="text/javascript">

function Empleado() {
    this.nombre = prompt("Introduzca el nombre d
    this.edad = prompt("Introduzca la edad de " +
    this.puesto = prompt("Introduzca ocupación " +
    this.mostrarPerfil = mostrarPerfil;
}
function mostrarPerfil() {
    document.write("<h1>Perfil del empleado</h1>
    document.write("<h2>" + this.nombre + "</h2>
    document.writeln("<p>Edad: " + this.edad);
</script>
</body></html>
```

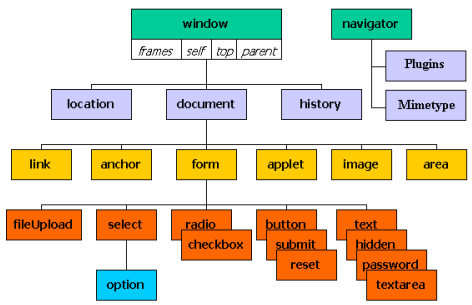



## Funciones y Objetos en JavaScript

- Objetos predefinidos
  - Objetos básicos

    Number
    Boolean
    String
    Array
    Date
    Math
  - Objetos del Navegador

    Windows
    Navigator



## Objetos Predefinidos en JavaScript

- **Objeto String**: Toda variable tipo cadena es un objeto String
  - Propiedades y métodos: Su utilización *cadena.método*

length : longitud de la cadena
anchor(*"ancla"*): Crea un ancla con el identificador especificado.
big() : Devuelve la cadena de caracteres con una fuente grande.
blink() : Devuelve la cadena de texto con un efecto intermitente.
charAt(*indice*): Devuelve el carácter situado en la posición especificada.
fixed() : Devuelve la cadena de caracteres con una fuente proporcional.
fontcolor(*color*) : Devuelve la cadena con el color especificado.
fontsize(*tamaño*) : Devuelve la cadena con el tamaño (1-7) indicado.
indexOf(*cad*, *indice*): Devuelve la posición de cad desde el índice indicado.
italics(): Muestra la cadena en cursiva.
lastIndexOf(*cad*, *indice*): Devuelve la posición de la última ocurrencia de cad.
link(URL): Devuelve la cadena como un enlace a la URL indicada.
small(): Devuelve la cadena con fuente más pequeña
split(*separador*): Devuelve la cadena separada con el separador especificado.
strike(): Devuelve la cadena tachada.
sub() : Devuelve la cadena con formato de subíndice.
substring(*Indice1*,*Indice2*): Devuelve la subcadena entre indice1 e indice2.
sup(): Devuelve la cadena con formato de superíndice.
toLowerCase(): Devuelve la cadena en minúsculas.
toUpperCase() : Devuelve la cadena en minúsculas.



## Objetos Predefinidos en JavaScript

- **Objeto String**: Ejemplo

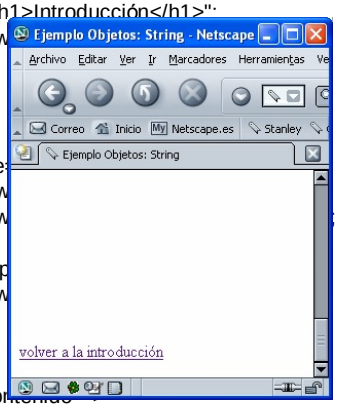
```
<html><head><title>Ejemplo Objeto: String</title>
</head><body>

<script type="text/javascript">
var cad="<h1>Introducción</h1>";
document.write(cad.anchor("intro"));
</script>

<hr />

<script type="text/javascript">
document.write("Formula: ".fontcolor("red"));
document.write("x"+"1".sub()+" = y"+"2".sup());

var c="Desplazate hacía abajo</p>";
document.write("<p>"+c.blink()+"</p>");

</script>

<!-- más contenido -->
```

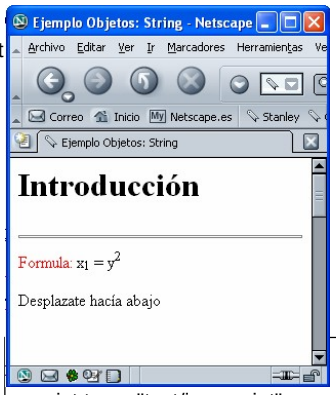
```
<!- - Continuación -->

<script type="text/javascript">
var cad="<p>volver a la introducción</p>";
document.write(cad.link("#intro"));
</script>

</body></html>
```

## Objetos Predefinidos en JavaScript

- **Objeto String**: Ejemplo

```
<html><head><title>Ejemplo Objeto: String</tit
</head><body>

<script type="text/javascript">
var cad="<h1>Introducción</h1>";
document.w
</script>

<hr />

<script type
document.w
document.w

var c="Desp
document.w

</script>

<!-- más co
```




```
<script type="text/javascript">
var cad="<p>volver a la introducción</p>";
document.write(cad.link("#intro"));
</script>

</body></html>
```



## Objetos Predefinidos en JavaScript

- **Objeto Array**: Colección de elementos indexados.

  (Los elementos no tienen porque ser del mismo tipo)

  - Declaración

    *objArray* **= new Array(***n_elementos***);**

    Es posible inicializarlo en su declaración new Array(*lista_elementos);*

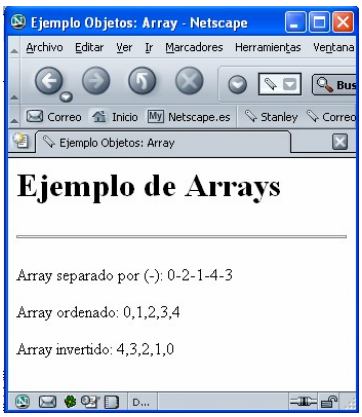
    Ejemplos: c=new Array(10); vocales=new Array(”a”,”e”,”i”,”o”,”u”);
  - Uso: c[1] = 10; c[2]=“Hola”; a=c[1]+4;
  - Propiedades y métodos ( *objArray.método* )

    length: devuelve el número de elementos del array.

    join(*separador*): Devuelve una cadena separada por el separador indicado
    reverse() : Devuelve el array con los elementos en orden invertido.
    sort() : Devuelve el array ordenado siguiendo el orden lexicográfico

## Objetos Predefinidos en JavaScript

- **Objeto Array**: Ejemplo




```
<html><head><title>Ejemplo Objetos: Array</title>
<script type="text/javascript">
function tratarArray(array, metodo) {
  switch (metodo) {
    case "join": return array.join("-");break;
    case "sort": return array.sort();break;
    case "reverse": return array.reverse();
  }
}
</script>
</head><body>
<h1>Ejemplo de Arrays</h1>
<hr />
<script type="text/javascript">
vocales=new Array(0,2,1,4,3);
document.write("<p>Array separado por (-): "+tratarArray(vocales,"join")+"</p>");
document.write("<p>Array ordenado: "+tratarArray(vocales,"sort")+"</p>");
document.write("<p>Array inverso: "+tratarArray(vocales,"reverse")+"</p>");
</script>
</body></html>
```



## Objetos Predefinidos en JavaScript

- **Objeto Date**: Manipulación de fechas
  - Declaración

    *objFecha* **= new Date(**[*año, mes, día, hora, minutos, segundos*]**);**

    donde año: xxxx, mes: 0-11, día: 1-(según mes), hora: 0-23, min y seg: 0-59

    Ejemplo: fecha = new Date();

  - Principales métodos

    getDate() : Devuelve el día del mes actual como un entero entre 1 y 31.
    getDay() : Devuelve el día de la semana actual como un entero entre 0 (D) y 6 (S).
    getHours() : Devuelve la hora del día actual como un entero entre 0 y 23.
    getMinutes() : Devuelve los minutos de la hora actual como un entero entre 0 y 59.
    getMonth() : Devuelve el mes del año actual como un entero entre 0 y 11.
    getSeconds() : Devuelve los segundos del minuto actual, un entero entre 0 y 59.
    getYear() : Devuelve el año actual como un entero, (para Netscape 0 es 1900).
    setDate(día_mes), setDay(día_semana), setHours(horas), setMinutes(minutos),
    setMonth(mes), setSeconds(segundos), setTime(milisegundos), setYear(año):
    Modifican la parte de la fecha indicada en el parámetro.
    toGMTString() : Devuelve una cadena con la especificación de zona horaria GMT.

## Objetos Predefinidos en JavaScript

- **Objeto Date**: Ejemplo

```
<html><head><title>Ejemplo Objetos: Date</title>
<script type="text/javascript">
meses=new Array("Enero", "Febrero", "Marzo", "Abril");
semana = new Array("Domingo","Lunes","Martes","Miércoles","Jueves","Viernes","Sábado");
</script>
</head><body>
<h1>Ejemplo con Fechas</h1> <hr />
<script type="text/javascript">
  f=new Date();
  var base=0;
  if (navigator.appName=="Netscape") base=1900;
  var agno=base+f.getYear();
  document.write(f.toGMTString()+"</p><hr />");
  document.write("<p>En Huelva: ");
  document.write(semana[f.getDay()]+", ");
  document.write(f.getDate()+" de "+meses[f.getMonth()]);
  document.write(" de "+agno);
  document.write(" a las "+f.getHours()+" horas "+f.getMinutes());
  document.write(" minutos "+f.getSeconds()+" segundos.</p>");
</script> </body></html>
```



## Objetos Predefinidos en JavaScript

- **Objeto Date**: Ejemplo

```
<html><head><title>Ejemplo Objetos: Date</title>
<script type="text/javascript">
meses=new Array("Enero", "Febrero", "Marzo", "Abril");
semana = new Array("Domingo","Lunes","Martes","Miércol
</script>
</head><body>
<h1>Ejemplo con Fechas</h1> <hr />
<script type="text/javascript">
  f=new Date();
  var base=0;
  if (navigator.appName=="Netscape") base=1900;
  var agno=base+f.getYear();
  document.write(f.toGMTString()+"</p><hr />");
  document.write("<p>En Huelva: ");
  document.write(semana[f.getDay()]+", ");
  document.write(f.getDate()+" de "+meses[f.getMo
  document.write(" de "+agno);
  document.write(" a las "+f.getHours()+" horas "+f.
  document.write(" minutos "+f.getSeconds()+" seg
</script> </body></html>
```


## Objetos Predefinidos en JavaScript

- **Objeto Math**: Funciones matemáticas
  - Utilización

    **Math.** *Metodo(…)_o_propiedad*;
  - Propiedades

    E : Número 'e', base de los logaritmos naturales (neperianos).
    LN2 : Logaritmo neperiano de 2.
    LN10 : Logaritmo neperiano de 10.
    LOG2E : Logaritmo en base 2 de e.
    LOG10E : Logaritmo en base 10 de e.
    PI : Número PI.
    SQRT1_2 : Raíz cuadrada de 1/2.
    SQRT2 : Raíz cuadrada de 2.

    Ejemplo: Math.E; Math.PI



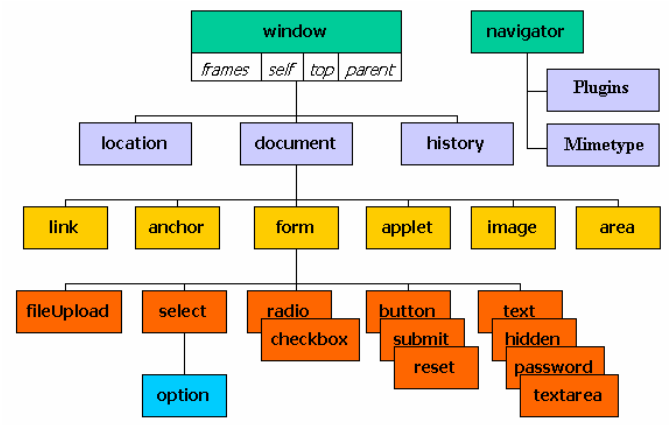
## Objetos Predefinidos en JavaScript

- **Objeto Math**: Funciones matemáticas
  - Métodos

    abs(num): Valor absoluto.
    acos(num): Arcocoseno. 'num' debe estar en el rango [-1,1], si no devuelve NaN.
    asin(num): Arcoseno. 'num' debe estar en el rango [-1,1], si no devuelve NaN.
    atan(num): Arcotangente. Si 'num' no es numérico devuelve NaN.
    atan2(x,y): Devuelve el ángulo formado por el vector (x,y) respecto al eje X.
    ceil(num): Devuelve el entero obtenido de redondear en exceso 'num'.
    cos(num): Coseno de 'num' o NaN si éste no es numérico.
    exp(num): Devuelve $e^{num}$.
    floor(num): Devuelve el entero obtenido de redondear por defecto 'num'.
    log(num): Devuelve el logaritmo neperiano de 'num'.
    max(x,y): Devuelve el máximo de 'x' e 'y'.
    min(x,y): Devuelve el mínimo de 'x' e 'y'.
    pow(base,exp: Devuelve $base^{exp}$.
    random(): Devuelve un número pseudoaleatorio entre 0 y 1.
    round(num): Redondea 'num' al entero más cercano.
    sin(num): Devuelve el seno de 'num' o NaN.
    sqrt(num): Devuelve la raíz cuadrada de número.
    tan(num): Devuelve la tangente de 'num' o NaN.

## Funciones y Objetos en JavaScript

- Objetos del Navegador






## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto Navigator**: Información del navegador

Propiedades
appCodeName. Código del Navegador.
appName: Nombre del Navegador.
appVersion: Versión del Navegador.
language (NS) o systemLanguage (IE): Idioma del Navegador.
mimeTypes: Array que contiene todos los tipos MIME soportados por el Navegador.
platform: Nombre de la plataforma sobre la que se ejecuta el Navegador.
plugins: Array que contiene todos los plug-ins soportados por el Navegador.
userAgent: Información de cabecera enviada en una petición HTTP.

Ejemplo

navigator.appName → Netscape
navigator.appName → Microsoft Internet Explorer

## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto window**: Objeto principal de la jerarquía.
    Información de la ventana donde se muestra el documento HTML.

    Propiedades principales

    closed: Valor booleano que inica si si la ventana está cerrada.
    defaultStatus: Valor por defecto para la barra de estado del navegador.
    frames: Array con los marcos (frames[0], frames[1], ...) que contiene la ventana. Su orden se asigna según se definen en el documento HTML.
    history: Array con las direcciones (URL's) visitadas en la ventana (historial).
    length: Número de marcos (frames) que tiene la ventana.
    location : Dirección (URL) actual mostrada en ventana.
    name : Nombre de la ventana.
    opener : Referencia al objeto window que abrió está ventana.
    parent : Referencia al objeto window que contiene el frameset.
    self y window : Nombres alternativo para la ventana.
    status : Información de la barra de estado.
    top : Nombre de la ventana del nivel superior.



## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto window**: Objeto principal de la jerarquía.
    Información de la ventana donde se muestra el documento HTML.

    Métodos principales (I)

    alert(msj) : Muestra el mensaje 'msj' en un cuadro de diálogo
    confirm(msj) : Muestra un cuadro de diálogo con el mensaje 'msj' y dos botones: Aceptar y Cancelar. Devuelve true si se pulsa aceptar y false si se pulsa cancelar.
    prompt(msj, defecto) : Muestra un cuadro de diálogo con el mensaje "msj" y una caja de texto. El parámetro 'defecto' es opcional, y muestra la respuesta por defecto. El método devuelve la respuesta introducida en el cuadro de texto.
    setTimeout(func, retardo) : Establece un retardo, en milisegundos, antes de evaluar func. Devuelve un identificador.
    setInterval(func, retardo) : Establece un intervalo, en milisegundos, para evaluar "func", periódicamente. Devuelve un identificador.
    clearTimeout(id) : Cancela el retardo "id".
    clearInterval(id) : Cancela el intervalo "id".

## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto window**: Objeto principal de la jerarquía.
    Información de la ventana donde se muestra el documento HTML.

Métodos principales (y II)

open(*url*, *nombre*, *opciones*) : Abre una ventana nueva con la *url* indicada. Donde *opciones* establecen las características de la nueva ventana. Estas son:
directories / location / menubar / resizable / scrollbars / status / toolbar (yes/no/ 1/0)
width / heigth / top / left / outerWidth / outerHeight (pixels)
Ejemplo: ventNueva=open(“”,”Prueba”, “directories=no, toolbar=yes”);

close() : Cierra la ventana.
focus() : Asigna el foco sobre la ventana.
blur() : Elimina el foco de la ventana.
moveBy(x, y) : Mueve la ventana hor. y vert. el número de pixels x e y, respect.
moveTo(x, y) : Mueve la ventana a las coordenadas (x, y).
scrollTo(x, y) : Desplaza el contenido de la ventana a las coordenadas (x, y).
scrollBy(x, y) : Desplaza el contenido x e y pixels, hor. y vert., respectivamente.



## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto window**: Ejemplo.

```
<html>  <head>
<title> Ejemplo Objeto Windows </title>
<script type=“text/javascript">
var v;
function a()  {
  var opciones = "left=100,top=100,width=250,height=150";
  v = window.open("", "", opciones);
  v.document.write("<h1>Nueva Ventana</h1>");
  setTimeout("b()",2000);   setTimeout("c()",4000);   setTimeout("v.close()",6000);
 }
function b() {   v.moveBy(200, 200);
                 v.document.write("<p>Movimiento desde función b().</p>");}
function c() {   v.moveTo(500, 100);
                 v.document.write("<p>Movimiento desde función c().</p>");}
</script>  </head>
<body><h1>Ejemplo con Objeto Window</h1>
<script type=“text/javascript">   a(); </script>
</body> </html>
```

Ejemplo

ej-objetoWindow.html

## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto frame**: Información sobre los marcos que contiene una ventana.

```
<html>  <head>   <title> Frame superior </title>
 <script>
  function escribirEn(Opcion, Texto) {
    var marco;
    if (Opcion[0].checked) marco=top.Izquierdo;  else marco=top.Derecho;
    marco.document.write(Texto.value);  Texto.value="";
  }
 </script>  </head>
 <body>
 Escribe un texto y elige el marco para escribirlo.
 <form>
 Texto : <input type="text"  name="texto" size="20"><br />
 Marco: <input type="radio" name="opcion" value="izq"
     checked="checked">Izquierdo
        <input type="radio" name="opcion" value="der">Derecho<br />
 <input type="button" value="Escribir"
                onClick="escribirEn(this.form.opcion, this.form.texto);">
 </form>
</body> </html>
```

Ejemplo

ej-objetoFrame.html



## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto Location**: Información sobre la URL visualizada.

Propiedades:
hash : Nombre del enlace, dentro de la URL.
host : Nombre de dominio y número del puerto en la URL.
hostname : Nombre de dominio (o dirección IP) en la URL.
href : Cadena URL completa.
pathname : Ruta al recurso solicitado por la URL.
port : Puerto especificado en la URL
protocol : Protocolo utilizado (incluyendo los dos puntos), dentro de la URL.
search : Información pasada en una llamada a un script CGI, dentro de la URL.

Métodos:
reload() : vuelve a cargar la URL especificada en la propiedad href del objeto location.
replace(cadenaURL) : Reemplaza el historial actual mientras carga la URL especificada en cadenaURL.

Ejemplo:  location.replace("http://www.uhu.es");

Ejemplo

ej-objetoLocation.html

## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto history**: Histórico de páginas visitadas.

Propiedades:

current : URL de la entrada actual en el historial.
next : URL de la siguiente entrada en el historial.
previous : URL de la anterior entrada en el historial.
length : Número de entradas en el historial.

Métodos:

back() : cargar la URL anterior en el historial.
forward() : cargar la URL siguiente dentro del historial.
go(pos) : cargar la URL especificado por 'pos' dentro del historial.



## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto document**: Contenido de la página visualizada.

Propiedades:

alinkColor : Color de los enlaces activos
anchors : Array con los anclas (enlaces internos) existentes en el documento
applets : Array con los applets existentes en el documento
bgColor : Color de fondo del documento
cookie : Valores de las cookies del documento actual
domain : Nombre del servidor que ha servido el documento
fgColor : Color del primer plano
forms : Array con los formularios del documento.
images : Array con todas las imágenes del documento.
lastModified : Fecha de la última modificación del documento.
linkColor : Color de los enlaces
links : Array con los enlaces externos
location : URL del documento actual
referrer : URL del documento que llamó al actual, en caso de usar un enlace.
title : Título del documento actual
vlinkColor : Color de los enlaces visitados

## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto document**: Contenido de la página visualizada.

Métodos:

clear() : Borra la ventana del documento.
close() : Cierra la escritura sobre el documento.
open(mime, "replace") : Abre la escritura sobre el documento, donde mime es el tipo de documento y “replace“ (opcional) reusa el documento anterior en el historial.
write() : Escribe texto en el documento.
writeln() : Escribe texto en el documento, y además lo finaliza con un salto de línea.

getElementBy(id) : Devuelve el elemento con identificador id del documento

Objetos principales que contiene:

link
anchor
image
form



## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto document**: Ejemplo document.cookie

Las cookies proporcionan una manera de conservar información entre peticiones del cliente. El cliente enviará esta información al servidor en cada petición.
La información se guarda en pares *nombre*=*valor*. Otras propiedades opcionales:

Fecha caducidad. Fecha hasta la que la cookie estará almacenada.
Ruta. Permite establecer una ruta a la que se enviará la cookie. (ej. /necesitoCookie)
Dominio de aplicación. Permite establecer un dominio al que enviar la cookie. (ej. uhu.es)
Trasmisión. Tipo de transmisión: segura (secure) o no.

Asignar cookie:
document.cookie = *nombre*=*valor* ; expires=*fechaGMT* ; path=*ruta* ; domain=*dom* ; secure;

Consultar:
Mostar el valor de document.cookie. Por ejemplo: alert(document.cookie);

Eliminar:
Basta crearla de nuevo con una fecha anterior (p.e. 1/1/1970) para que la elimine el navegador.

Ejemplo

ej-documentCookie.html

## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto link**: Información de un enlace.

Propiedades:
target : Ventana o frame especificado en el parámetro target del enlace.
hash : Nombre del ancla usada en el enlace.
host : Nombre de dominio y número de puerto del enlace.
hostname : Nombre de dominio (o la dirección IP)
href : URL completa
pathname : Rura al recurso del enlace.
port : Número de puerto.
protocol : Protocolo usado, incluyendo los : (los dos puntos).
search : Información pasada en una llamada a un script CGI.

Ejemplo: document.links[i].href

  - **Objeto anchor**: Información de un enlace interno (anclas).

Ejemplo

ej-objetoLink.html

Propiedades:
name : (sólo netscape) URL completa
target : (sólo IE) Ventana o frame especificado en el parámetro target del enlace.

Ejemplo: document.anchors.length; document.anchors[j].name;



## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto image**: Información de imágenes del documento.

Propiedades:

name : Nombre (name) asignado a la imagen .
src : Ubicación (src) de la imagen.
width : Ancho (width) de la imagen.
height : Altura (height) de la imagen.
border : Border de la imagen.
complete : Booleano que indica si la imagen se ha descargado completamente
hspace : Parámetro 'hspace' de la imagen
vspace : Parámetro 'vspace' de la imagen
lowsrc : Parámetro 'lowsrc' de la imagen
prototype : Permite crear nuevos parámetros para este objeto

Ejemplo: document.images[0].width *= 2;

Ejemplo

ej-objetoImage.html

## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto form**: Información de formulario.

Propiedades:

name : Nombre del formulario (parámetro name).
elements : Array con todos los elementos del formulario.
action : Parámetro action del formulario.
method : Método que va a recibir/procesar la información del formulario (get/post).

Métodos:

reset() : Resetea el formulario. El mismo efecto que click en un botón de tipo reset.
submit() : Envía el formulario. El mismo efecto que click en un botón de tipo submit.

Objetos internos:

text, password y textarea
button, submit y reset
checkbox
radio buton
select



## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto text, password y textarea**.

Propiedades:
name : Nombre del elemento (parámetro name)
value : Valor del elemento (parámetro value)
defaultValue : Valor por defecto

Métodos:
blur() : Quita el foco sobre el objeto especificado
focus() : Asigna el foco sobre el objeto especificado
select() : Selecciona el texto dentro del objeto dado

  - **Objeto button, submit, reset.**

Propiedades:
name : Nombre del elemento (atributo name)
value : Valor del elemento (aributo value)

Métodos:
click(): Acción de pulsado del botón

## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto checkbox**.

Propiedades:
checked : Booleano que indica si el checkbox está pulsado.
defaultChecked : Booleano que indica si el checkbox está seleccionado por defecto.
name : Nombre del checkbox (atributo name).
value : Valor de checkbox (atributo value).
Métodos:
click() : Acción de pulsado del checkbox

  - **Objeto radio button.**

Propiedades:
checked : Booleano que indica si el radio está seleccionado.
defaultChecked : Booleano que indica si el radio está seleccionado por defecto.
length : Número de opciones dentro de un grupo de elementos radio
name : Nombre del radio button (atributo name).
value : Valor del racio button (atributo value)
Métodos:
click(): Acción de pulsado de la opción del radio button

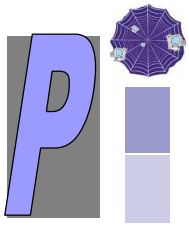


## Funciones y Objetos en JavaScript

- Objetos del Navegador
  - **Objeto select**.

Propiedades:

length : Número de opciones tiene la lista
name : Nombre de la lista (atributo name)
options : Array que contiene las opciones de la lista. Sus propiedades son:
  defaultSelected: Booleano que indica si la opción está seleccionada por defecto
  index: Posición de la opción dentro de la lista
  length: Número de opciones tiene la lista
  options: Código HTML de la lista
  selected: Booleano que indica si la opción está seleccionada
  text: Texto mostrado en la lista de una opción.
  value: Valor de una opción de la lista
selectedIndex : Posición de la opciones que está actualmente seleccionada.



# Programación en Internet

## Tema 3. JavaScript

**Contenido**





DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA


## Eventos en JavaScript

- Eventos (lo definiremos aquí como)

  Un suceso que se produce en un documento HTML y requiere tratamiento mediante un script (JavaScript)

- Posibles eventos
  - cargar la página (onLoad , unLoad)
  - clic en un enlace o botón (onClick)
  - Movimiento del ratón por un elemento (onMouseOver, onMouseOut)
  - Acciones en un formulario (onSubmit , onReset)
  - Cambio del valor de un campo (onChange)
  - Activar (foco) de un elemento (onFoco, onBlur)
  - Selección de texto (onSelect)
  - Pulsar teclas (onKeyDown, onKeyUp, onKeyPress)
  - Pulsar boton del ratón (onMouseDown, onMouseUp)

- Manejo de eventos

  *<etiqueta onEvento="codigo_javascript" …>*

  Ejemplo: <a href="http://www.uhu.es" onclick="preguntar()">UHU</a>

## Eventos en JavaScript

| Evento | Causa | Principales Etiquetas |
|---|---|---|
| onLoad | El documento se carga | <body> |
| onUnload | El documento se descarga | <body> |
| onClick | Se pulsa el boton izq del ratón | button, submit, reset, radio, checkbox, <a href> y un Area |
| onMouseOver | El ratón pasa sobre una zona | <a href> y cualquier Area |
| onMouseOut | El ratón sale de una zona | <a href> y cualquier Area |
| onSubmit | Se envía un formulario | <form> |
| onReset | Se resetea el formulario | <form> |
| onChange | Cambia el contenido | text y TextArea |
| onFocus | Se recibe el foco | text y TextArea |
| onBlur | Se pierde el foco | text y TextArea |
| onSelect | Se selecciona texto | text y TextArea |
| onAbord | Se interrumpe la carga | <body> <img> |
| onKeyDown<br>onKeyUp<br>onKeyPress | Pulsar una tecla<br>Se deja de pulsar una tecla<br>Se deja pulsada una tecla | Text y TextArea |
| onMouseDown<br>onMouseUp | Pulsar un botón del ratón<br>Dejar de pulsar un botón del ratón | <a href>, button, submit, reset |
| onResize | Redimensionar una ventana | <body> |



## Eventos en JavaScript

- Ejemplo 1: onLoad / onUnLoad

```
<html> <head>  <title>Página principal</title>

<script language="JavaScript">
var meVotastes=false;

function pedirVoto(){
   if (confirm("Votación: ¿Te gusta mi web?")){
     meVotastes=true;
     open("http://www.la-urna.com/votar.php?idvoto=xxxxx","","");
   }
}
function agradecer(){
   if (meVotastes){  alert("Muchas gracias por tu voto");  }
}
</script>
</head>

<body onload="pedirVoto()"  onUnLoad="agradecer()">
<h1>Mi Página Web</h1>
<p> Bla, bla, bla,... </p> <p> … <br /> …. </p>
</body> </html>
```

Ejemplo

ej-eventoonLoad.html

## Eventos en JavaScript

- Ejemplo 2: onClick

```
<html>  <head>    <title>Página principal</title> <script language="JavaScript">
function tratarClick(obj){
 switch(obj.id) {
   case "enl" : alert("Click en Enlace");break;
   case "tab" : alert("Click en Tabla");break;
   case "fbt" : alert("Click en boton de formulario");break;
   case "fcb" : alert("Click en checkbox de formulario");break;
  }
}
</script>  </head>  <body>  <h1>Ejemplo onClick</h1>

<p><a id="enl" href="" onclick="tratarClick(this)">Enlace</a></p>

<table id="tab" border="1" onclick="tratarClick(this)">
<tr><td>Celda1</td><td>Celda2</td></tr></table>

<p> <form id="form" name="Formulario" method="" action="">
<input type="checkbox" id="fcb" onclick="tratarClick(this)" /> CheckBox<br />
<input id="fbt" type="button" value="Enviar" onclick="tratarClick(this)" />
</form> </p>

</body> </html>
```

Ejemplo

ej-eventoonClick.html



## Eventos en JavaScript

- Ejemplo 3: onMouserOver / onMouseOut

```
<html>
<head>

<script type="text/javascript">
function colorFondo(bool) {
     var p=document.getElementById("par");
     if (bool) p.className = "conRaton";
     else p.className = "sinRaton";
}
</script>
</head>
<body>
<h1>Ejemplo onMouserOver / onMouseOut </h1>

<p id="par" class="sinRaton"
     onMouseOver="colorFondo(true)"
     onMouseout="colorFondo(false)">
Pasa el ratón por aquí
</p>
</body>
</html>
```

```
<style type="text/css">
.sinRaton {
          font-family: Arial;
          font-size: 12px;
          font-weight: bold;
          color: #FFFFFF;
          background-color: #999999;
          border: thin ridge #006666;
}

.conRaton {
          font-family: Arial;
          font-size: 12px;
          font-weight: bold;
          color: #999999;
          background-color: #993300;
          border: thin inset #00CC66;
}
</style>
```

Ejemplo

ej-eventoMouse.html

## Eventos en JavaScript

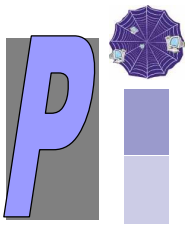
- Ejemplo 4: onSubmit

```
<html>  <head>  <title>Página principal</title>
<script language="JavaScript">

function validar(f){
  if(f.nombre.value=="") {alert("Introduce Nombre"); return false;}
  if(f.apellido.value=="") {alert("Introduce Apellido"); return false;}
  if(f.email.value=="") {alert("Introduce Correo"); return false;}
  if(f.email.value.indexOf("@", 0)==-1) {alert("Introduce Correo correcto"); return false;}
  return true;
}
</script>  </head>   <body>
<h1>Ejemplo onSubmit</h1>
<form name="Formulario" method="post" action="procesar.jsp" OnSubmit="return
    validar(this)">
Nombre: <input type="text" name="nombre"><br />
Apellido: <input type="text" name="apellido"><br />
Correo : <input type="text" name="email"><br />
<input type="submit" name="Submit" value="Enviar">
</form>
</body> </html>
```

Ejemplo

ej-eventoonSubmit.html





# Tema 3. JavaScript

**Resumen**

1. Introducción
2. Fundamentos de JavaScript
   - Tipos básicos y variables
   - Operadores
   - Estructuras de Control
3. Funciones y Objetos en JavaScript
   - Funciones
   - Objetos y Métodos
4. Eventos en JavaScript

**Referencias**

Libro: Capítulos 7-14. Internet & WWW: How to Program.
Capítulo 5 . Programación de Aplicaciones Web.

W3 Schools,
http://www.w3schools.com/js/

Universidad de Huelva
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA, SISTEMAS INFORMÁTICOS Y AUTOMÁTICA