

Programación Estática y Laboratorio Web

Lenguaje JavaScript



Universidad Nacional del Comahue
Facultad de Informática
Departamento de Programación



[Funciones]

- Funcionalidad generalizada
- Comúnmente agregadas en archivos externos o en head.
- Parámetros
- Valor Retornado
- Acceso a variables locales, parámetros y variables globales
- EjemploJS008Funciones.html

```
<script
type="text/javascript">
function nombre_funcion(var1,
var2, ..) {
//...
return valor
}
</script>
```

Sentencia Condicional

if – else

- Se emplea para tomar decisiones basadas en algunas condiciones

```
if (condicion) {  
    //Codigo a ejecutar si condicion es true  
}  
else{  
    //Codigo a ejecutar si condicion es es false  
}
```

Sentencia Condicional switch

- Se emplea para tomar decisiones basadas en condiciones referidas a un valor.
- La sentencia break es opcional, pero si no la ponemos a partir de que se encuentre coincidencia con un valor se ejecutarán todas las sentencias relacionadas con este y todas las siguientes
- EjemploJS009CondicionalesIfSwitch.html

```
switch (n)
{
  case 1:
    //si n==1 ejecuta bloque 1
    break;
  case 2:
    //si n==1 ejecuta bloque 2
    break;
  default:
    //si n!=1 y n!=2 ejecuta bloque
    //default
}
```

[Bucles JavaScript]

- **for.** Se emplea para crear un ciclo en base a un índice *i*
- **while/ do-while.** Se emplea para crear un ciclo en base a una condición.
- **for-in.** Se emplea para crear un ciclo que recorra un array de elementos.

Bucles JavaScript

```
for (var i=startvalue;i<=endvalue;i++){  
    //codigo a ser ejecutado  
}  
  
while (var<=endvalue){  
    //codigo a ser ejecutado  
}  
  
do{  
    //codigo a ser ejecutado  
}while (var<=endvalue)  
  
for (variable in object){  
    //codigo a ser ejecutado  
}
```

- EjemploJS010BuclesForDoWhileForIn.html

[break y continue]

- break: corta con la ejecución del bucle actual
- continue: corta con la secuencia del ciclo actual y vuelve a comenzar el bucle desde el principio
- EjemploJS011BuclesBreakContinue.html

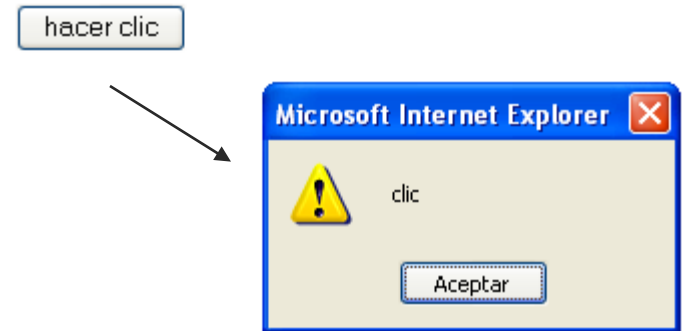
[Eventos]

- Programación basada en eventos.
- Capturar eventos y ejecutar Funciones.

```
<html>
<script type="text/javascript">
function accion() {
    alert('clic');
}
</script>
<body>
    <button onclick="accion();" >
        hacer clic
    </button>
</body>
</html>
```


[Eventos]

- Programación basada en eventos.
- Capturar eventos y ejecutar Funciones.



[Eventos body]

- onload: La página se ha cargado completamente. `<body>`.
- onunload: se abandona la página. `<body>`
- onresize: se modifica el tamaño de la ventana del navegador. `<body>`

[Eventos Key]

- onkeydown: Pulsar una tecla (sin soltar). Elementos de Formulario y <body>
- onkeypress: Pulsar una tecla. Elementos de Formulario y <body>
- onkeyup: Soltar una tecla. Elementos de Formulario y <body>
- EjemploJS012EventosBody.html

Eventos mouse

- onclick: Hacer click y soltar el botón del mouse. Todos los elementos.
- ondblclick: Hacer dos veces click. Todos los elementos.
- onmousedown: Pulsar (sin soltar) un botón del mouse. Todos los elementos.
- onmousemove: Mover el mouse. Todos los elementos.
- onmouseout: El mouse sale del elemento. Todos los elementos.
- onmouseover: El mouse entra en el elemento. Todos los elementos.
- onmouseup: Soltar el botón del mouse. Todos los elementos.
- onmousedown: Pulsar (sin soltar) un botón del mouse. Todos los elementos.
- EjemploJS012EventosMouse.html

Eventos Form

- onblur: Deseleccionar un elemento. <button>, <input>, <label>, <select>, <textarea>, <body>.
- onchange: Deseleccionar un elemento que se ha modificado. <input>, <select>, <textarea>
- onfocus: Seleccionar un elemento. <button>, <input>, <label>, <select>, <textarea>, <body>.
- onreset: Inicializar el formulario (borrar todos sus datos). <form>.
- onsubmit: Enviar el formulario. <form>
- onselect: Selección de Texto. <input>, <textarea>
- EjemploJS012EventosForm.html

[OBJETOS UTILES]

- Programación orientada a objetos.
- Acceso a propiedades y métodos de los objetos.
- Actualización de propiedades que modifican el comportamiento.
- Ejecución de acciones a través de métodos.

[Navigator]

- appName: retorna el nombre del browser.
- userAgent: retorna el header del cliente (browser).
- appVersion: retorna la versión del browser.
- systemLanguage: retorna el lenguaje utilizado por el sistema operativo.
- EjemploJS013DOMNavigator.html

[Window]

- `alert()`, `prompt()`, `confirm()`
- `resizeTo(width,height)`: cambia el tamaño de la ventana.
- `moveTo(X,Y)`: mueve la ventana con desplazamiento X e Y.
- `Close()`: cierra la ventana.
- `open(URL)`: *abre un pop-up.*
- *EjemploJS013DOMWindow.html*

[Window.history]

- Length: retorna el número de páginas visitadas.
- back(): Vuelve a la url anterior en la historia del explorador.
- forward(): Va a la url siguiente en la historia del explorador.
- Go(n): va a la url en la posición n de la historia del explorador.

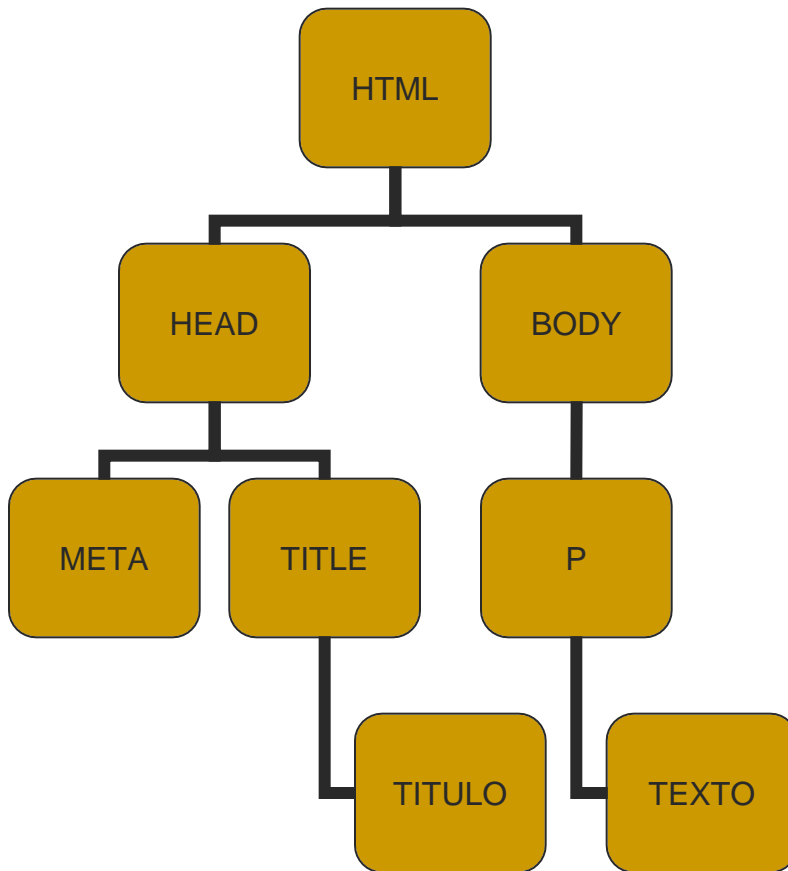
[Window.location]

- reload(): refresca el contenido de la ventana a la URL actual.
- replace(URL): reemplaza el contenido de la ventana actual por URL.
- EjemploJS013DOMLocation.html

[DOM]

- Document Object Model.
- Permite acceder a cualquiera de los objetos que forman el documento como si fuera un árbol.
- Permite modificar cualquiera de las propiedades de los objetos que forman parte del documento.

DOM



```
<!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR
xhtml1/DTD/xhtml1-transitional.dtd">

<html
xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-
1" />
    <title>TITULO</title>
  </head>
  <body>
    <p>TEXTO</p>
  </body>
</html>
```

[DOM Tipos de Nodos]

- Document, nodo raíz del que derivan todos los demás nodos del árbol.
- Element, representa cada una de las etiquetas XHTML. Se trata del único nodo que puede contener atributos y el único del que pueden derivar otros nodos.
- Attr, se define un nodo de este tipo para representar cada uno de los atributos de las etiquetas XHTML, es decir, uno por cada par atributo=valor.
- Text, nodo que contiene el texto incluido en una etiqueta XHTML.
- Comment, representa los comentarios incluidos en la página XHTML.

[DOM acceso]

- `document.getElementsByTagName(tag)`: devuelve un array de Objetos del Tag seleccionado.
- `document.getElementsByName(name)`: devuelve un array de Objetos del que tiene el atributo name seleccionado.
- `document.getElementById(id)`: Devuelve el objeto que tiene el atributo id seleccionado.
- `EjemploJS014DOMBody.html`

[Referencias y Links]

■ Tutoriales

- http://www.wikilearning.com/tutorial/completo_tutorial_de_javascript/4466
- <http://www.w3schools.com/js/default.asp>

■ Estándar

- <http://www.ecma-international.org/publications/standards/Ecma-262.htm>

■ Libro

- Introducción a JavaScript, Autor: Javier Eguíluz Pérez
http://www.librosweb.es/javascript/pdf/introduccion_javascript.pdf