

# JavaScript

---

LENGUAJE DE SCRIPTING DEL LADO DEL CLIENTE

# Introducción a JavaScript

---

- Desarrollado por Netscape en 1995.
- Se ejecuta del lado del cliente (Client Side).
- Características del lenguaje:
  - Lenguaje completo de programación.
  - Multiparadigma: Estructurado y Orientado a Objetos.
  - Interpretado: por el navegador web.
  - Débilmente tipado: Variables de tipos dinámicos.
- JavaScript (Netscape, ECMA/ISO) / JScript (Microsoft)

# JavaScript: Para qué se utiliza

---

- Animación a los elementos de la página (ej: menús desplegables, slider de fotos, pestañas, transiciones, popups personalizados).
- Contenido interactivo (juegos, audio y video).
- Validación de formularios.
- AJAX: Comunicación con el servidor sin recargar la página.

# JavaScript: Dónde se escribe

---

- Embebido en el código HTML, entre etiquetas "script"

`<script> ... </script>`

- En un archivo externo, incluido en el HTML

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

# JavaScript: Sintaxis

---

- Basada en la sintaxis del lenguaje C y Java.
- Las sentencias se separan con punto y coma (;) o saltos de línea.

```
<script>  
instrucción1;  
instrucción2;  
instrucción3;  
</script>
```

- Se puede escribir un programa en una sola línea de código.  

```
<script> instrucción1; instrucción2; instrucción3; </script>
```
- Distingue mayúsculas y minúsculas (Case Sensitive).
- Comentarios de una línea // y multilínea /\* \*/

# JavaScript: Elementos principales

---

- Variables
- Operadores
- Estructuras de control
- Funciones
- DOM

# JavaScript: Variables

---

- No se definen los tipos de variable.
- Ámbito de las variables:
  - Locales (dentro de una función), se declaran con la instrucción `var`.
  - Globales (a todo el programa), se declaran sin la instrucción `var`.
- Su nombre sólo puede estar formado por letras, números, símbolos \$ y \_ (guion bajo), y el primer carácter no puede ser un número.

# JavaScript: Variables

---

- Tipos de datos:
  - Numeric (para cualquier tipo de número)
  - String (cadena de texto o char)
  - Boolean (true/false)
  - Array/Object
  - Undefined (variables sin asignar)
  - Null
- Se puede saber el tipo de una variable utilizando la función `typeof(variables)`



# JavaScript: Strings

---

- Las cadenas se concatenan con el operador +

```
var cadena = "Hola " + nombre;
```

- Caracteres especiales:

```
\n  \t  \'  \"  \\  \r  \b  \f
```

- Funciones para manejo de strings:

```
.length / .concat() / .toUpperCase() / .toLowerCase() / .charAt(x) /  
.indexOf("a") / .lastIndexOf("a") / replace("a", "b") / .split("&") /  
.substr(inicio, long) / .parseInt() / .parseFloat()
```

# JavaScript: Arrays

---

- Declaración y carga

```
var autos = new Array();  
autos[0] = "Saab";  
autos[1] = "Volvo";  
autos[2] = "BMW";
```

```
var autos = new Array("Saab", "Volvo", "BMW");
```

```
var autos = ["Saab", "Volvo", "BMW"];
```

- Funciones para manejo de arrays

```
.length / .concat() / .join(",") / .pop() / .push() / .reverse() / .shift() /  
.unshift() / .sort() / .slice(from, to) / .splice(from, to)
```

# JavaScript: Arrays

---

- Arrays asociativos / Objetos

```
var persona = {  
    nombre:    "Juan",  
    apellido:  "Perez",  
    edad:      25  
};  
  
persona["altura"] = 1.78;  
  
persona.peso = "70 kg";
```

# JavaScript: Objeto Date

---

- Crear fechas

```
var fechaActual = new Date();
```

```
var fecha1 = new Date("January 1, 2016 12:00:00");
```

```
var fecha2 = new Date(16,0,1,12,0,0);
```

- Funciones para manejo de fechas

```
.getFullYear() / .getMonth() / .getDate() / .getDay() / .getHours() /
```

```
.getMinutes() / .getSeconds() / .getMilliseconds() / .getTime()
```

# JavaScript: Operadores

---

- Aritméticos: ( +, -, \*, /, ++, --, % )
- Asignación: ( =, +=, -=, \*=, /= )
- Cadenas: ( + )
- Lógicos: ( !, &&, || )
- Comparación: ( ==, !=, <, >, <=, >= )
- Binarios: ( &, ^, | )

# JavaScript: Estructuras de control

---

- Condicional IF-ELSE

```
if (condición) {  
    verdadero;  
}  
else {  
    falso;  
}
```

```
variable = condición ? verdadero : false;
```

# JavaScript: Estructuras de control

---

- Condicional SWITCH

```
switch (variable) {  
  case valor1:  
    operaciones;  
    break;  
  case valor2:  
    operaciones;  
    break;  
  default:  
    operaciones;  
    break;  
}
```

# JavaScript: Estructuras de control

---

- Ciclo FOR

```
for (inicio; condición; incremento) {  
    operaciones;  
    break;  
    continue;  
}
```

- Ciclo FOR-IN

```
var persona =  
    { nombre:"Juan", apellido:"Perez", edad:25 };  
  
for (x in persona) {  
    document.write(x + ":" + persona[x] + " ");  
}
```



# JavaScript: Estructuras de control

---

- Ciclo WHILE

```
while (condición) {  
    operaciones;  
}
```

- Ciclo DO-WHILE

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

# JavaScript: Estructuras de control

---

- Manejo de excepciones

```
try {  
    //instrucciones  
}  
catch (err) {  
    //salida por error  
}
```

# JavaScript: Funciones

---

- Declaración de una función

```
function nombreFuncion (parámetro1, parámetro2) {  
    operaciones;  
    return resultado;  
}
```

```
nombreFuncion = function (parámetro1, parámetro2) {  
    operaciones;  
    return resultado;  
}
```

- Llamado a la función

```
nombreFuncion(x, y);
```

```
var resultado = nombreFuncion(x, y);
```

# JavaScript: Funciones

---

- Funciones con cantidad variable de parámetros

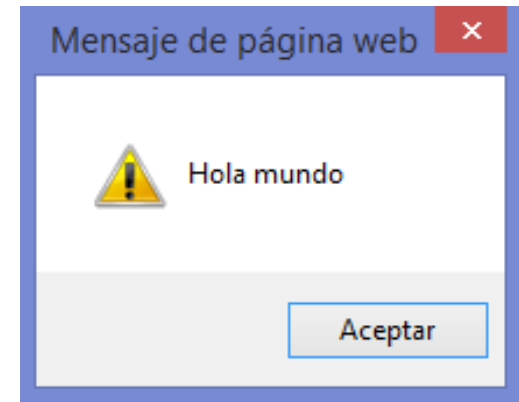
```
function argumentosVariables() {  
    var i;  
    for (i in arguments) {  
        document.write(arguments[i] + " ");  
    }  
}  
  
argumentosVariables(1, 123, 500, 115, 44);
```

- Los parámetros de tipos primitivos se pasan por copia.
- Los objetos se pasan por referencia.

# JavaScript: Popups

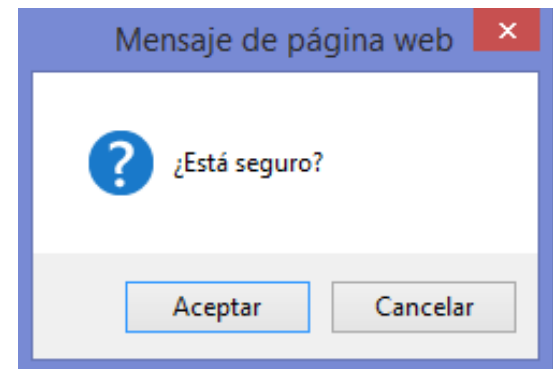
- Mensaje de alerta

```
alert("Hola mundo");
```



- Cuadro de confirmación

```
confirm("¿Está seguro?");
```



- Cuadro de ingreso de texto

```
prompt("Ingrese su nombre:");
```

# Ejemplo #1

## Funciones y arrays

Mostrar un listado de usuarios por pantalla.

```
<head>
  <title>JavaScript</title>
  <script type="text/javascript">

    function mostrarUsuarios(cant) {
      var usuarios =
        ["Juan","Carla","Ana","Manuel","Ary"];
      usuarios.sort();
      cant = cant != null ? cant : usuarios.length;
      var lista = "Usuarios \n===== \n";
      for (i=0; i<usuarios.length && i<cant; i++) {
        lista += usuarios[i] + "\n";
      }
      alert(lista);
    }

    mostrarUsuarios();
    mostrarUsuarios(3);

  </script>
</head>
...
```

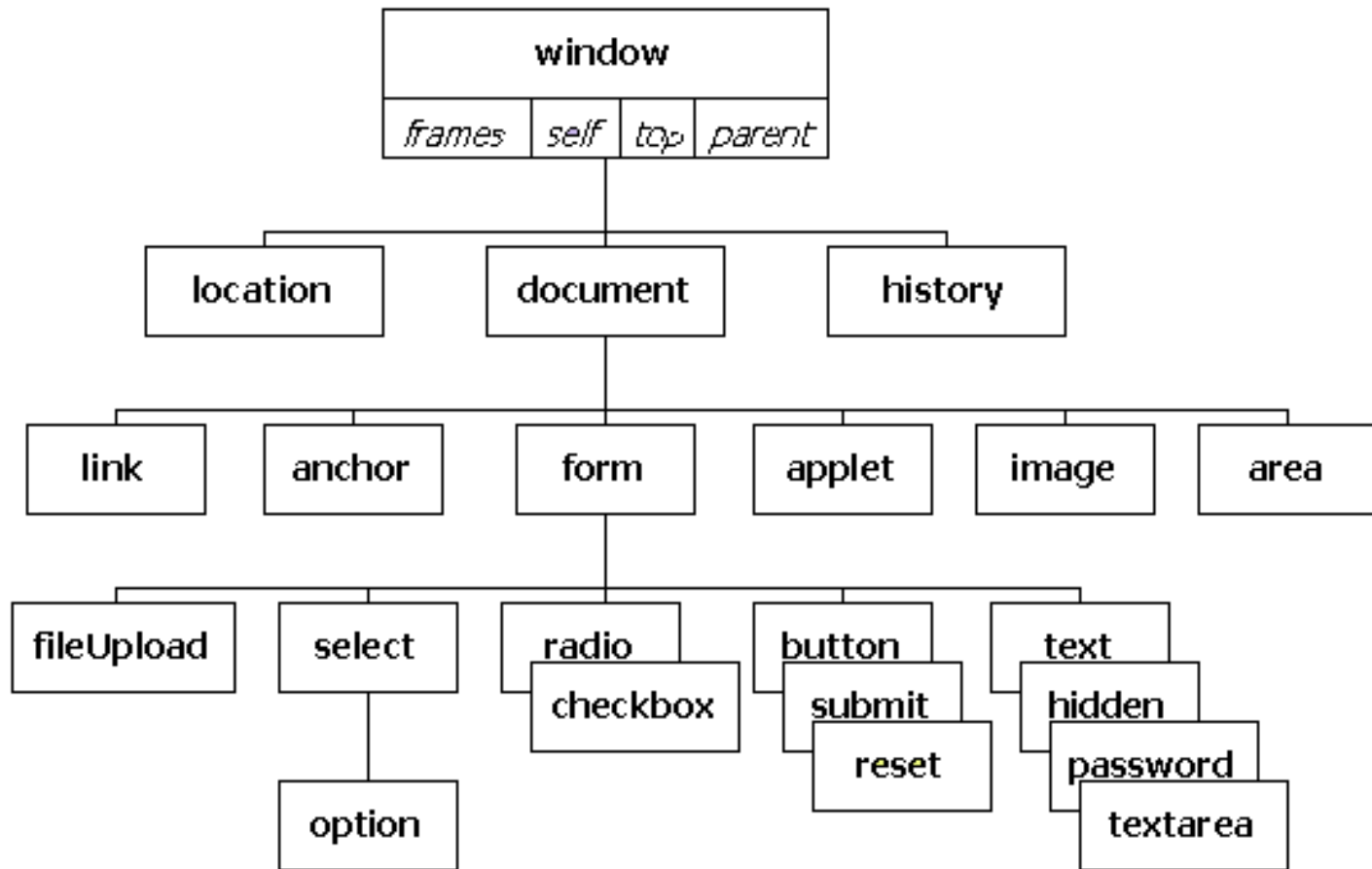
# JavaScript: DOM

---

- Document Object Model
- Creado por el navegador cuando se carga la página web.
- Estructura de árbol.
- El DOM permite a JavaScript acceder y modificar el contenido, la estructura y los estilos del documento.

# JavaScript: DOM

---





# JavaScript: DOM

---

- Obtener elementos HTML por id, clase o etiqueta:
  - `document.getElementById("id")`
  - `document.getElementsByClassName("clase")`
  - `document.getElementsByTagName("div")`

```
<div id="id" class="clase">Texto</div>
```

# JavaScript: DOM

---

- Obtener elementos HTML por colección de objetos:
  - `document.images` // Array de `<img>`
  - `document.forms` // Array de `<form>`
  - `document.anchors` // Array de `<a>`
  - `document.title` // Elemento `<title>`
  - `document.URL` // URL completa de la página
  - Entre otros...

# JavaScript: DOM

---

- Modificación de los elementos:
  - *elemento.innerHTML* = "nuevo contenido html";  
Ejemplo: document.title = "Introduccion a JS";
  - *elemento.atributo* = *valor*;  
Ejemplo: document.images[0].src = "imagen1.png";
  - *elemento.style.propiedad* = *valor*;  
Ejemplo: document.getElementById("boton").style.color = "red";
- Otros métodos útiles:
  - document.write("Bienvenido");
  - window.location.href = "http://misitio.com"
  - window.history.back();

# JavaScript: Eventos

---

- Acciones que realiza el usuario sobre los elementos de la página: click, doble click, foco, presionar una tecla, enviar un formulario, arrastrar un elemento, etc.
- A cada evento se puede asociar código JavaScript.
- Llamados a funciones desde atributos HTML.

```
<div onmouseover="cambiarColor()">TEXTO</div>
```

```
<select onchange="recargarValores()">...</select>
```

```
<button onclick="calcular()" value="OK" />
```

```
<a href="javascript:validarForm()">ENVIAR</a>
```

LISTADO DE EVENTOS

## Ejemplo #2

Validación de  
formularios.

Evento onclick y  
función submit

```
<head>
  <script type="text/javascript">

    function validarFormulario() {
      var nombre = document.getElementById("nombre");
      var dni = document.getElementById("dni");
      if (nombre.value.length == 0) {
        nombre.style.backgroundColor = "red";
      }
      else if (dni.value.length == 0 || isNaN(dni.value)) {
        dni.style.backgroundColor = "red";
      }
      else {
        alert("Correcto!!!");
        document.getElementById("formu").submit();
      }
    }
  </script>
</head>

<body>
<form id="formu" name="formu" method="get" action="#">
  Nombre:<input type="text" id="nombre" name="nombre" />
  DNI:<input type="text" id="dni" name="dni" />
  <a href="#" onclick="validarFormulario();">Enviar</a>
</form>
</body>
```

## Ejemplo #3

Modificación de  
estilos.

Evento onmouseover  
y onmouseout

```
<head>
  <script type="text/javascript">

    function enfasis(over) {
      if (over) {
        document.getElementById("bienvenida").style
          .backgroundColor = "green";
        document.getElementById("bienvenida").style
          .color = "white";
      }
      else {
        document.getElementById("bienvenida").style
          .backgroundColor = "white";
        document.getElementById("bienvenida").style
          .color = "black";
      }
    }
  </script>
</head>

<body>
  <span id="bienvenida" onmouseover="enfasis(true) ;"
onmouseout="enfasis(false) ;">HOLA MUNDO</span>
</body>
```

## Ejemplo #4

Secuencia de  
ejecución.

Evento onload

```
<head>
  <script type="text/javascript">

    function mensaje(){
      document.write("Hola mundo 3<br>");
    }
    document.write("Hola mundo 1<br>");

    window.onload = function() {
      alert("Hola mundo 5");
    }

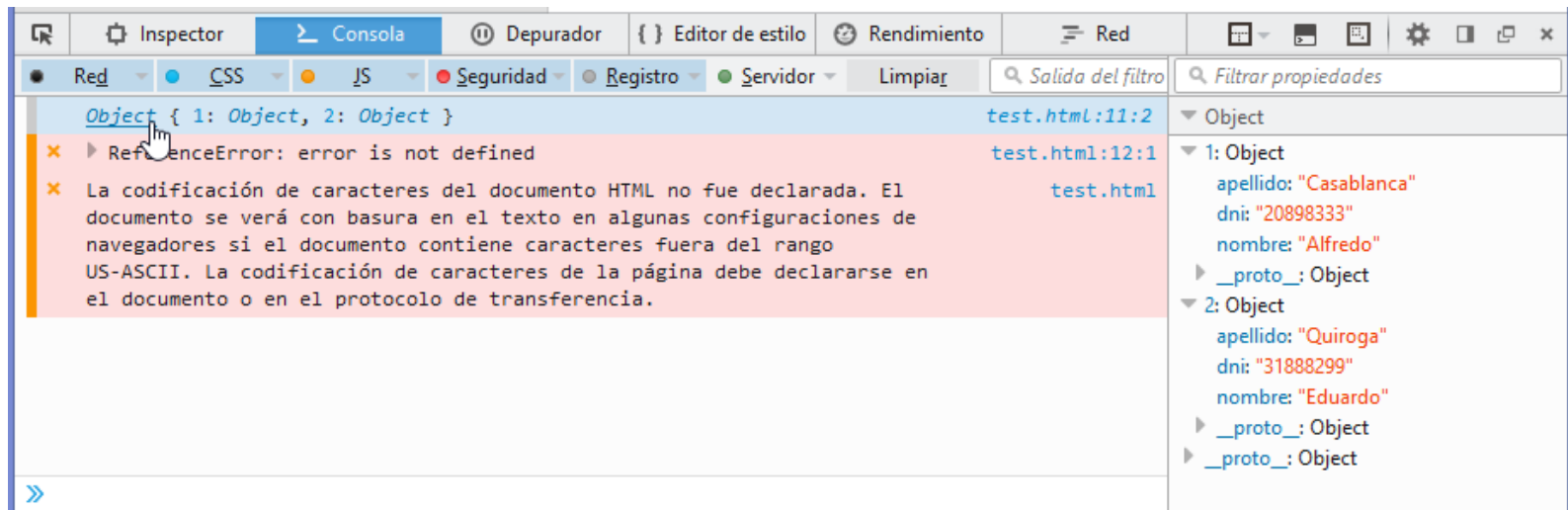
  </script>
</head>
<body>
  Hola mundo 2<br>
  <script>mensaje() ;</script>
  Hola mundo 4<br>
</body>
```



# JavaScript: Debug

¿Cómo detectar errores en nuestro código JavaScript?

- Consola del navegador (<F12>)
- Método `console.log(...)`





# Referencias

---

- <http://www.w3schools.com/js/default.asp>
- <https://es.wikipedia.org/wiki/JavaScript>
- [https://es.wikipedia.org/wiki/Document Object Model](https://es.wikipedia.org/wiki/Document_Object_Model)
- [http://www.w3schools.com/js/js htmlDOM document.asp](http://www.w3schools.com/js/js_htmlDOM_document.asp)
- <http://www.htmlquick.com/es/reference/events.html>

AUTOR: LEANDRO MORRONE