



## JavaScript

Roberto Gómez Cárdenas

rogomez@itesm.mx

<http://homepage.cem.itesm.mx/rogomez>

Lámina 1

Roberto Gómez C.



## Introducción

- Diseñado para añadir interacción a páginas HTML.
- Es un lenguaje de tipo script
- Usualmente JavaScript se encuentra embebido directamente en las páginas HTML.
- Es un lenguaje interpretativo.
- Cualquiera puede usar JavaScript sin usar una licencia de compra.
- JavaScript NO es lo mismo que Java.

Lámina 2

Roberto Gómez C.



## ¿Qué se puede hacer con JavaScript?

- Proporciona a los diseñadores de HTML una herramienta de programación.
- Permite poner texto dinámico dentro de un página HTML.
- Puede reaccionar a ciertos eventos.
- Puede leer y escribir elementos HTML
- Puede ser usado para validar datos.
- Puede ser usado para detectar el navegador del visitante.
- Puede ser usado para crear cookies.

Lámina 3

Roberto Gómez C.



## Enunciados JavaScript

- Sensibles a mayúsculas.
- Es un comando dirigido al browser, que le indica al browser que hacer.
- Al final del enunciado se acostumbra poner un “;”.
- Los enunciados pueden estar agrupados en bloques delimitados por corchetes

```
<script type="text/javascript">
{
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
}
</script>
```

Lámina 4

Roberto Gómez C.



## JavaScript en HTML

- Para insertar código JavaScript dentro de una página HTML, se debe usar la etiqueta `<script>`.
- Se usa el atributo `type` dentro de esta etiqueta para definir el lenguaje.
- La etiqueta indica donde empieza y termina el código.

```
<html>
<body>
<script type="text/javascript">
...
</script>
</body>
</html>
```

Lámina 5

Roberto Gómez C.



## Un pequeño ejemplo

- Script que abre una ventana.

```
<html>
<body>
<h3> Primer ejemplo JavaScript</h3>
<script type="text/javascript">
window.open("http://www.google.com",
            " ", "width=550,height=420,menubar=no")
</script>
</body>
</html>
```

- A tomar en cuenta que el nombre del archivo debe contar con la extensión `html`.

Lámina 6

Roberto Gómez C.



## JavaScript en navegadores que no soportan JavaScript

- Navegadores que no soportan JavaScript, desplegará el código como contenido de la página.
- Para impedir esto, y como parte del estándar, se debe usar la etiqueta de comentarios de HTML.
  - Añadir `<!--` al inicio y `!-->` al final del código

```
<html>
<head>
  Ejemplo de JavaScript
</head>
<body>
<script type="text/javascript">
  <!--
  document.write("Hola Mundo!");
  -->
</script>
</body>
</html>
```

Lámina 7

Roberto Gómez C.



## ¿Donde poner el JavaScript?

- El código se ejecuta inmediatamente mientras la página se carga en el navegador.
- En algunas ocasiones se desea ejecutar el código cuando la página se cargue y en un evento posterior, como cuando un usuario presiona un botón del ratón.
  - En este último caso, el código se inserta dentro de una función.

Lámina 8

Roberto Gómez C.



## JavaScript en <head>

- Código a ser ejecutado cuando es invocado, o cuando un evento es activado, son dejados en funciones.
- Las funciones se definen en la sección <head>, de esta forma estarán en todos los lugares, y no interfieren con el contenido de la página.

```
<html>
<head>
<script type="text/javascript">
function message()
{
alert("Caja alerta a invocarse dado
un evento");
}
</script>
</head>
<body onload="message()">
</body>
</html>
```

Lámina 9

Roberto Gómez C.



## JavaScript en <body>

- Si el código no se define dentro de una función, o si el código debe escribir contenido de una página, este debe colocarse en la sección <body>.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
document.write("Mensaje escrito en JavaScript");
</script>
</body>
</html>
```

Lámina 10

Roberto Gómez C.



## JavaScript en <head> y <body>

- Se puede contar con varios scripts en todo el documento.

```
<html>
<head>
<script type="text/javascript">
function message()
{
alert("This alert box was called with the onload event");
}
</script>
</head>

<body onload="message()">
<script type="text/javascript">
document.write("Mensaje escrito en JavaScript");
</script>
</body>
</html>
```

Lámina 11

Roberto Gómez C.



## Usando un script externo

- Si se desea correr el mismo código en varias páginas, sin tener que escribirlo en cada página, se puede escribir en un archivo externo.

```
<html>
<head>
<script type="text/javascript" src="xxx.js"></script>
</head>
<body>
</body>
</html>
```

Lámina 12

Roberto Gómez C.



## Enunciados JavaScript

- Enunciado JavaScript es un comando para el browser.
  - Le indica al navegador que hacer.

- Ejemplo

```
document.write("Hola Mundo!");
```

- El punto y coma al final es opcional.
- Posible agrupar enunciados en un bloque usando corchetes.

```
<script type="text/javascript">
{
document.write("<h1>Hola Mundo</h1>");
document.write("<p>Adios a todos.</p>");
}
</script>
```

Lámina 13

Roberto Gómez C.



## Comentarios

- Para una línea: //

```
<script type="text/javascript">
// Write a heading
document.write("<h1>This is a heading</h1>");
// Write two paragraphs:
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
</script>
```

- Para un conjunto de líneas: /\* \*/

```
<script type="text/javascript">
/*
The code below will write
one heading and two paragraphs
*/
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
</script>
```

Lámina 14

Roberto Gómez C.



## Variables JavaScript

- Son sensibles a mayúsculas.
- Deben empezar con una letra o guión bajo.
- Es posible declarar una variable con la palabra reservada var

```
var x;  
var marcacoche;
```
- Posible asignar valores a la variable cuando se declara.

```
var x=5;  
var marcacoche="Peugeot";
```
- Si se asignan valores a variables no declaradas, estas serán automáticamente declaradas.

```
x=5;  
marcacoche="Peaugot";
```

Lámina 15 Roberto Gómez C.



## Variables

- Si se re-declara una variable, esta no pierde su valor original.

```
var x=5;  
var x
```

  - la variable x aún tendrá el valor de 5
- Aritmética en JavaScript
  - Se puede hacer operaciones aritméticas con las variables

```
y = x + 5;  
z = y + 5;
```

Lámina 16 Roberto Gómez C.



## Operadores aritméticos

Asumiendo que  $y = 5$

Operador	Descripción	Ejemplo	Resultado
+	Adición	$x=y+2$	$x=7$
-	Substracción	$x=y-2$	$x=3$
*	Multiplicación	$x=y*2$	$x=10$
/	División	$x=y/2$	$x=2.5$
%	Modulus	$x=++y$	$x=1$
++	Incremento	$x=-y$	$x=6$
--	Decremento	$x=-y$	$x=4$

Lámina 17 Roberto Gómez C.



## Operadores de asignación

Asumiendo que  $x = 10$  e  $y = 5$

Operador	Ejemplo	Igual que	Resultado
=	$x=y$		$x=5$
+=	$x+=y$	$x=x+y$	$x=15$
-=	$x-=y$	$x=x-y$	$x=5$
*=	$x*=y$	$x=x*y$	$x=50$
/=	$x/=y$	$x=x/y$	$x=2$
%=	$x%=y$	$x=x\%y$	$x=0$

Lámina 18 Roberto Gómez C.



## Ejemplo

---

```

<html>
<body>
<script type="text/javascript">
x=5+5;
document.write(x);
document.write("<br />");
x="5"+"5";
document.write(x);
document.write("<br />");
x=5+"5";
document.write(x);
document.write("<br />");
x="5"+5;
document.write(x);
document.write("<br />");
</script>
<p> La regla es: si usted suma un numero y un string, el resultado sera un string. </p>
</body>
</html>

```

Lámina 19 Roberto Gómez C.



## Cast entre variables

---

- Posible usar Number() y String() para convertir tipos de variables.
 

```

var n = 30; // number
var m = 70; // number
var s = new String();// "3070"
s = String(n) + String(m);
// "100"
s = Number(n) + Number(m);
</html>

```

  - Nota: Combinación Number() + String() produce un string
- También es posible convertir una variable a un tipo Boolean().
 

```

var n = 50; // number

var b = Boolean(n); // b is "true"
var c = Boolean(n - 50); // c is "false"

```

Lámina 20 Roberto Gómez C.



## Caso especial

- En el código de abajo, ¿a que valor booleano se evalúan las variables?

```
var s = new String(); // string nulo
var b = Boolean(s); // ¿que valor tiene b?
```

- Un string nulo evalua a true
- Si se pasa como parámetro de Boolean( ) será verdad.

Lámina 21 Roberto Gómez C.



## Operadores comparación

Asumiendo que  $x = 5$

Operador	Descripción	Ejemplo
==	Es igual a	$x == 8$ es falso
===	Es exactamente igual a (valor y tipo)	$x == 5$ es verdad $X == "5"$ es falso
!=	Diferente	$X != 8$ es verdad
>	Mayor que	$X > 8$ es falso
<	Menor que	$X < 8$ es falso
>=	Mayor que o igual a	$X >= 8$ es falso
<=	Menor que o igual a	$X <= 8$ es verdad

Ejemplo de uso

```
if (edad < 18) document.write("Menor edad");
```

Lámina 22 Roberto Gómez C.



## Operadores lógicos y operador condicional

---

- Operadores lógicos

Operador	Descripción	Ejemplo
&&	and	(x < 10 && y > 1) => Verdadero
	or	(x == 5    y == 5) => Falso
!	not	!(x == y) => Verdadero

- Operador condicional
  - Asigna un valor a una variable basada en una condición.
  - Sintaxis  
`nombrevARIABLE=(condition)?value1:value2`
  - Ejemplo  
`equipo=(lugar=="campeon")?"America":"Chivas";`  
 Si lugar tiene valor de campeon, equipo tendra un valor de America, si no de Chivas.

Lámina 23
Roberto Gómez C.



## Operadores if...else

---

```
if (condition) {
  codigo a ejecutar si la condición es verdad
}
```

```
if (condition) {
  codigo a ejecutar si la condición es verdad
}
else {
  código a ejecutar si la condición no es verdad
}
```

```
if (condition) {
  codigo a ejecutar si la condición es verdad
}
else {
  codigo a ejecutar si la condición no es verdad
}
```

Lámina 24
Roberto Gómez C.



## Ejemplo

```
<html>
<body>
<script type="text/javascript">
var d = new Date();
var hora = d.getHours();
if (hora < 10)
{
    document.write("<b>Buenos dias</b>");
}
</script>

<p>Ejemplo que demuestra el enunciado if.</p>
<p>Si la hora en su navegador es menor a 10, se
desplegara un mensaje de "Buenos dias"</p>

</body>
</html>
```

Lámina 25

Roberto Gómez C.



## Enunciado If...else if...else

```
if (condicion1) {
    codigo a ejecutar si la condicion1 es verdad
}
else if (condition2) {
    codigo a ejecutar si la condición2 es verdad
}
else {
    codigo a ejecutar si la condicion1 y condicion2
    no son verdad
}
```

Lámina 26

Roberto Gómez C.



## Ejemplo enunciado If...else if...else

```
<html>
<body>
<script type="text/javascript">
var d = new Date();
var hora = d.getHours();
if (hora < 12) {
    document.write("<b>Buenos dias</b>");
}
else if (hora >= 12 && hora < 19) {
    document.write("<b>Buenas tardes</b>");
}
else {
    document.write("<b>Buenas noches</b>");
}
</script>
<p> Ejemplo de uso del enunciado if..else if...else. </p>
</body>
</html>
```

Lámina 27

Roberto Gómez C.



## Operador switch

- Sintaxis

```
switch(n)
{
    case 1:
        bloque ejecución 1
        break;
    case 2:
        bloque ejecución 2
        break;
    default:
        codigo a ejecutar si n es diferente
        a las opciones 1 y 2
}
```

Lámina 28

Roberto Gómez C.



## Ejemplo switch (1/2)

```
<html>
<body>
<script type="text/javascript">
var d = new Date(); dia=d.getDay();
switch (dia) {
case 5:
  document.write("<b>Al fin viernes</b>");
  break;
case 6:
  document.write("<b>Super Sabado</b>");
  break;
case 0:
  document.write("<b>Domingo levanta tarde</b>");
  break;
default:
  document.write("<b>I'm really looking forward to this weekend!</b>");
}
</script>
```

Lámina 29

Roberto Gómez C.



## Ejemplo switch (2/2)

<p>This JavaScript will generate a different greeting based on what day it is. Note that Sunday=0, Monday=1, Tuesday=2, etc.</p>

```
</body>
</html>
```

Lámina 30

Roberto Gómez C.



## Cajas de JavaScript

- Cajas de alertas
  - Asegurarse que la información proviene a través del usuario.
  - Usuario debe confirmar con un “OK”.
- Cajas de confirmación
  - Cuando se desea que el usuario verifique o acepte algo.
- Cajas de captura.
  - Se desea introducir un valor antes de entrar a una página
  - Si el usuario presiona “OK” la caja regresa el valor introducido.
  - Si el usuario presiona “Cancel”, la caja regresa un valor nulo.

Lámina 31

Roberto Gómez C.



## Caja Alerta

- Sintaxis:

```
alert("texto");
```

- Ejemplo:

```
<html>
<head>
<script type="text/javascript">
function show_alert()
{
    alert("Soy una caja de alerta!");
}
</script>
</head>
<body>
```

```
<input type="button"
    onclick="show_alert()"
    value="Ejemplo caja" />
</body>
</html>
```

Lámina 32

Roberto Gómez C.



## Salida ejemplo caja alerta

---

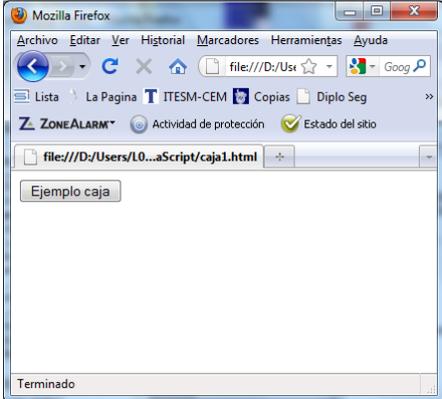
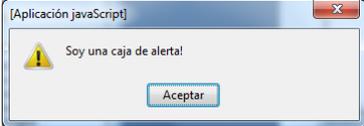



Lámina 33
Roberto Gómez C.



## Caja de confirmación

---

- Sintaxis
 

```
confirm("texto");
```
- Ejemplo
 

```
<html>
<head>
<script type="text/javascript">
function show_confirm()
{
  var r=confirm("Presione un boton");
  if (r==true) {
    alert("Presiono Aceptar!");
  }
  else {
    alert("Presiono Cancelar!");
  }
}
</script>
</head>
<body>
<input type="button"
  onclick="show_confirm()"
  value="Mostrar caja
  confirmación " />
</body>
</html>
```

Lámina 34
Roberto Gómez C.

 Ejemplo salida confirmación

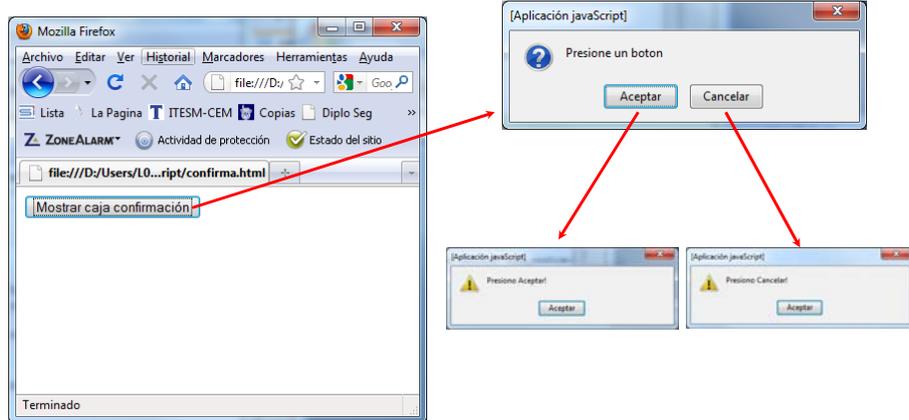


Lámina 35 Roberto Gómez C.

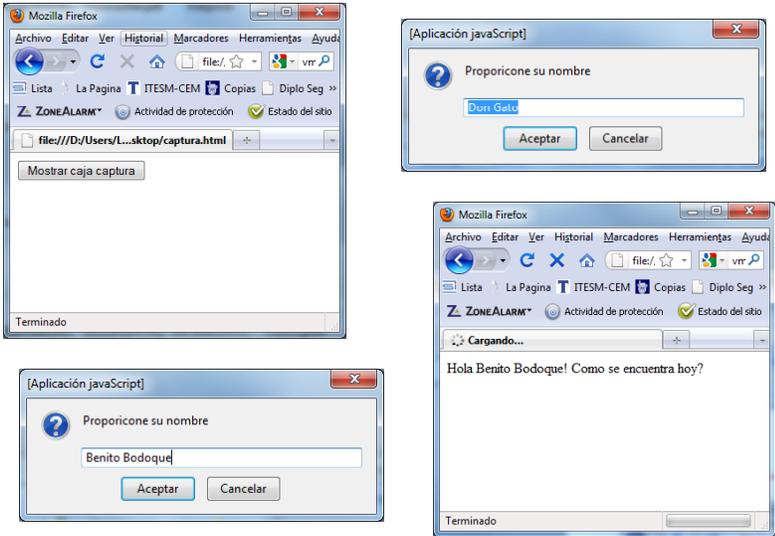
 Caja de captura

- Sintaxis  
`prompt("texto", "valor.por.default");`
- Ejemplo  
`prompt("sometext", "defaultvalue");`

Lámina 36 Roberto Gómez C.



## Salida ejemplo



The screenshot illustrates the execution of a JavaScript application. It shows three windows:

- A Mozilla Firefox browser window displaying a file named 'captura.html' with a 'Mostrar caja captura' button.
- A dialog box titled '[Aplicación JavaScript]' with the prompt 'Proporicone su nombre' and a text input field containing 'Benito Bodoque'.
- A second Mozilla Firefox browser window showing the output: 'Hola Benito Bodoque! Como se encuentra hoy?'.

Lámina 37 Roberto Gómez C.



## Funciones

- Código que sera ejecutado por un evento o por una invocación a la función.
- Se puede invocar a una función desde cualquier parte de la página.
  - O desde otras páginas si la función se encuentra en un archivo externo .js
- Las funciones pueden ser definidas dentro de las secciones de <head> o <body> del documento.
  - Asegurarse que la función es leída/cargada por el navegador antes de ser llamada.
  - Puede ser conveniente poner las funciones en la sección <head>

Lámina 38 Roberto Gómez C.



## Definiendo una función

- Sintaxis

```
function functionname(var1, var2, ..., varX)
{
  código de la función
}
```
- Los parámetros  $p_1, p_2, \dots, p_n$  son variables pasadas a la función.
- Los corchetes define el inicio y fin de la función.
- Una función sin parámetros debe incluir los paréntesis.
  - Recordar que JavaScript es sensible a mayúsculas.

Lámina 39 Roberto Gómez C.



## Ejemplo función

```
<html>
<head>
<script type="text/javascript">
function mensaje()
{
  alert("Hola Mundo!");
}
</script>
</head>

<body>
<form>
<input type="button" value="Presioname!" onclick="mensaje()" />
</form>
</body>
</html>
```

Lámina 40 Roberto Gómez C.



## El enunciado return

---

- Especifica el valor que es regresado por la función.

```

<html>
<head>
<script type="text/javascript">
function multiplica(a,b)
{
return a*b;
}
</script>
</head>
<body>
<h3> Prueba regreso valor funcion </h3>
<p> Se mando llamar multiplica(4,3) </p>
</body>
</html>

```

Lámina 41 Roberto Gómez C.



## Ciclos

---

- Dos tipos de ciclos
  - El enunciado for
 

```

for (var=startvalue;var<=endvalue;var=var+increment)
{
código a ser ejecutado
}

```
  - El enunciado while
 

```

while (var<=endvalue)
{
código a ser ejecutado
}

```

Lámina 42 Roberto Gómez C.



## Ejemplo enunciado for

```
<html>
<body>
<script type="text/javascript">
var i=0;
for (i=0;i<=5;i++)
{
document.write("El numero es" + i);
document.write("<br />");
}
</script>
</body>
</html>
```

Lámina 43

Roberto Gómez C.



## Ejemplo enunciado while

```
<html>
<body>
<script type="text/javascript">
var i=0;
while (i<=5)
{
document.write("El numero es " + i);
document.write("<br />");
i++;
}
</script>
</body>
</html>
```

Lámina 44

Roberto Gómez C.





## Enunciado continue

---

- Se salta una vuelta del ciclo
- Ejemplo

```

<html>
<body>
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
  if (i==3)
  {
    continue;
  }
  document.write("The number is " + i);
  document.write("<br />");
}
</script>
</body>
</html>

```

Lámina 47
Roberto Gómez C.



## Enunciado for ... in

---

- Recorre los elementos de un arreglo o las propiedades de un objeto
- Sintaxis

```

for (variable in objeto)
{
  código a ejecutar
}

```

- Ejemplo

```

<html>
<body>
<h3> Ejemplo for in </h3>
<script type="text/javascript">
var x;
var carros = new Array();
mycars[0] = "Honda";
mycars[1] = "Volvo";
mycars[2] = "Peugeot";

for (x in mycars)
{
  document.write(carros[x] + "<br />");
}
</script>

</body>
</html>

```

Lámina 48
Roberto Gómez C.



## Eventos

- Acciones que pueden ser detectadas por JavaScript.
- Cada elemento en una página web cuenta con eventos que pueden activar un JavaScript.
- Por ejemplo evento onClick de elemento button, indica que una función se ejecutará cuando el usuario haga un click en el botón.

Lámina 49

Roberto Gómez C.



## Ejemplos de eventos

- Un click del ratón.
- Una página web o una imagen cargandose.
- Seleccionar un campo de entrada en una forma HTML.
- Someter una forma HTML.
- El presionar una tecla.

Lámina 50

Roberto Gómez C.



## Evento onclick

- Crea un evento cuando texto activo, imágenes o cajas son “activadas” presionando el ratón.
- Ejemplo

```
<A HREF="http://www.htmlgoodies.com"
onClick="alert('Hello out there!');">HTML Goodies</A>
```

Lámina 51

Roberto Gómez C.



## Ejemplo onClick

```
<html>
<head>
<script>
function ejecutaEventoOnClick(){
  x = 30
  alert('El valor de x es ' + x)
}
</script>
</script>
</head>

<body>
<form>
<input type=button value=Pulsame onclick="ejecutaEventoOnClick()">
</form>
</body>
</html>
```

Lámina 52

Roberto Gómez C.



## Ejemplo onClick sobre una referencia

```
<html>
<head>
<script>
function zona(){
alert("Esta es una zona restringida");
}
</script>
</script>
</head>

<body>
<form>
Prueba evento onClick sobre una liga. Visite la siguiente liga:
  <a href="/codigo/prueba/zonarestringida.html" onClick="zona()">
    Zona Restringida  </a>
</form>
</body>
</html>
```

Lámina 53 Roberto Gómez C.



## Eventos onLoad y onUnload

- Son activado cuando el usuario entra o deja una página.
- Evento onLoad es usado para verificar el tipo y versión del navegador, y cargar la apropiada versión de la página web basada en dicha información.

Lámina 54 Roberto Gómez C.



## Ejemplo onLoad y onUnload

```

<HTML>
<HEAD>
<TITLE>Ejemplo con funciones</TITLE>
<SCRIPT LANGUAGE="JavaScript">
var name = ""
function Hola() {
  nombre = prompt('Introduzca su nombre:','')
  alert('¡Hola ' + nombre + '!')
}

function Adios() {
  alert('¡Adios ' + nombre + '!')
}
</SCRIPT>
  
```

</HEAD>  
 <BODY onLoad="Hola()" onUnload="Adios()">

Esta pagina es de ejemplo de los eventos de carga (onLoad) y descarga (onUnload).

</BODY>  
 </HTML>

Lámina 55 Roberto Gómez C.



## Eventos onFocus, onBlur y onChange

- Son usados en combinación con la validación de campos de una forma.
- Ejemplo uso de evento onChange.
  - La función checkEmail() será invocada cuando el usuario cambie el contenido del campo:

```

<input type="text" size="30" id="email"
      onchange="checkEmail()">
  
```

Lámina 56 Roberto Gómez C.



## Ejemplo onChange

```
<html>
<head>
function mensaje() {
  alert("Se modifiko la selección ");
}
</head>
<body>
<form>
<SELECT onchange=mensaje(>
  <OPTION value="opcion1"> Opcion 1
  <OPTION value="opcion2"> Opcion 2
</SELECT>
</form>
</body>
</html>
```

Lámina 57

Roberto Gómez C.



## Evento onSubmit

- Usada para validar TODOS los campos de la forma antes de enviarla.
- Ejemplo de uso:
  - La función checkForm() es invocada cuando el usuario de un click al botón de envío de la forma.
  - Si los valores no son aceptados, la forma será cancelada.
  - La función checkForm() regresa falso o verdadero.
    - Si regresa verdad la forma será enviada, sino sera cancelada.

```
<form method="post" action="xxx.htm"
      onsubmit="return checkForm()">
```

Lámina 58

Roberto Gómez C.



## Eventos onMouseOver y onMouseOut

---

- Usados para crear botones “animados”.
- Ejemplo
  - Una caja de alerta aparece cuando un evento onMouseOver es detectado.

```

<a href="http://www.w3schools.com"
  onmouseover="alert('An onMouseOver event');
  return false">
</a>

```

Lámina 59
Roberto Gómez C.



## Ejemplo evento

---

- Se despliega una caja con un mensaje diferente, dependiendo si se pasa el ratón o se presiona el ratón.

```

<html>
<head>
<script type="text/javascript">
function mensaje1() {
    alert("Hola Mundo")
}
function mensaje2() {
    alert("Hello World")
}
</script>
</head>
<body>
<h3> Ejemplo de un evento </h3>
<input type="button" value="Presioname!"
onclick="mensaje1()"><br />
<a href="#" onmouseover=""
onMouseout="mensaje2()">
Saludos!</a>
</body>
</html>

```

Lámina 60
Roberto Gómez C.



## Lista de eventos

Eventos	Descripción	Elementos que lo admiten
onLoad	Terminar de cargarse una página	<BODY...> <FRAMESET...>
onUnLoad	Salir de una página (descargarla)	<BODY...><FRAMESET...>
onMouseOver	Pasar el ratón por encima	<A HREF...> <AREA...>
onMouseOut	Que el ratón deje de estar encima	<A HREF...> <AREA...>
onSubmit	Enviar un formulario	<FORM...>
onClick	Pulsar un elemento	<INPUT TYPE="button, checkbox, link, radio" ...>
onBlur	Perder el cursor	<INPUT TYPE="text" ...> <TEXTAREA...>
onChange	Cambiar de contenido o perder el cursor	<INPUT TYPE="text" ...> <TEXTAREA...>
onFocus	Conseguir el cursor	<INPUT TYPE="text" ...> <TEXTAREA...>
onSelect	Seleccionar el texto	<INPUT TYPE="text" ...> <TEXTAREA...>

Lámina 61 Roberto Gómez C.



## Atrapando errores: Try ... Catch

- Enunciado permite probar un bloque de código para ver si tiene errores.
- Objetivo
  - Evitar que se desplieguen errores en el navegador del usuario.
- Sintaxis
 

```
try
  {
    // Correr código en esta parte
  }
catch(err)
  {
    //Manejar errores en esta parte
  }
```

Lámina 62 Roberto Gómez C.



## Ejemplo

---

```

<html>
<head>
<script type="text/javascript">
var txt="";
function message()
{
try
{
addAlert("Bienvenido!");
}
catch(err)
{
txt="Hay un error en esta pagina.\n\n";
txt+="Descripcion del error: " + err.description + "\n\n";
txt+="Presione OK para continuar.\n\n";
alert(txt);
}
}

```

```

</script>
</head>
<body>
<input type="button"
value="View message"
onclick="message()" />
</body>
</html>

```

Lámina 63 Roberto Gómez C.



## El enunciado throw

---

- Permite crear una excepción.
- En combinación con enunciado try ... catch, puede controlar el flujo de programa y generar mensajes de error más precisos.
- Sintaxis
  - throw(exception)
  - La excepción puede ser un string, un entero, booleano o un objeto.

Lámina 64 Roberto Gómez C.



## Ejemplo enunciado throw

- Determina valor de x.
  - Si x es mayor que 10, menor que 0, o no es un número, se arroja un error.
  - El error es capturado y se despliega un mensaje de error.

```

<html>
<body>
<script type="text/javascript">
var x=prompt
  ("Introduzca un numero entre 0 y 10:","");
try {
  if (x>10) {
    throw "Err1";
  }
  else
    if (x<0) {
      throw "Err2";
    }
  else
    if ( isNaN(x) ) {
      throw "Err3";
    }
}

```

Lámina 65 Roberto Gómez C.



## Continuación ejemplo enunciado throw

- Determina valor de x.
  - Si x es mayor que 10, menor que 0, o no es un número, se arroja un error.
  - El error es capturado y se despliega un mensaje de error.

```

catch(er) {
  if(er=="Err1") {
    alert("Error! El valor es muy alto");
  }
  if(er=="Err2") {
    alert("Error! El valor es muy bajo");
  }
  if(er=="Err3") {
    alert("Error! El valor no es un número");
  }
}
</script>
</body>
</html>

```

Lámina 66 Roberto Gómez C.



## Caracteres especiales

- La diagonal invertida (\) es usada para insertar apostrofes, nuevas líneas y caracteres especiales en un texto.

Lámina 67 Roberto Gómez C.



## A tomar en cuenta

- JavaScript es sensible a mayúsculas.
- Espacios en blanco
  - Los espacios extras son ignorados.

```
name="Hege";  
name = "Hege";
```
- Se puede “romper” una línea de código y continuar en la parte de abajo.
  - Se usa la diagonal invertida para esto.

```
document.write("Hello \  
World!");
```
  - Sin embargo lo siguiente no es posible

```
document.write \  
("Hello World!");
```

Lámina 68 Roberto Gómez C.



## Objetos

---

- JavaScript es un lenguaje orientado a objetos.
- Se cuentan con objetos ya creados por JavaScript, con sus metodos y propiedades.
  - Propiedades: valores asociados con un objeto.
  - Métodos: Acciones llevadas a cabo por un objeto.

Lámina 69
Roberto Gómez C.



## Ejemplo

---

- Ejemplo propiedades
 

```

<script type="text/javascript">
var txt="Hola Mundo!";
document.write(txt.length);
</script>
```
- Ejemplo metodos
 

```

<script type="text/javascript">
var str="Hola Mundo!";
document.write(str.toUpperCase());
</script>
```

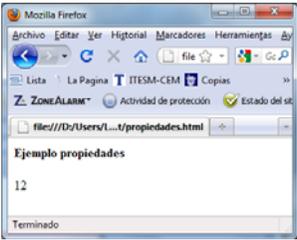
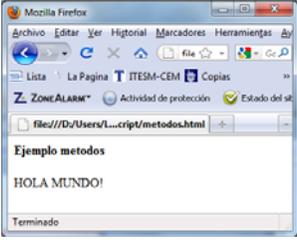



Lámina 70
Roberto Gómez C.



## Objeto navigator

---

- Contiene información sobre el navegador.
- Ejemplo:

```

<html>
<body>
<script type="text/javascript">
document.write("Browser CodeName: " + navigator.appCodeName);
document.write("<br /><br />");
document.write("Browser Name: " + navigator.appName);
document.write("<br /><br />");
document.write("Browser Version: " + navigator.appVersion);
document.write("<br /><br />");
document.write("Cookies Enabled: " + navigator.cookieEnabled);
document.write("<br /><br />");
document.write("Platform: " + navigator.platform);
document.write("<br /><br />");
document.write("User-agent header: " + navigator.userAgent);
</script>
</body>
</html>

```

Lámina 71
Roberto Gómez C.



## Ejemplo salida ejemplo

---

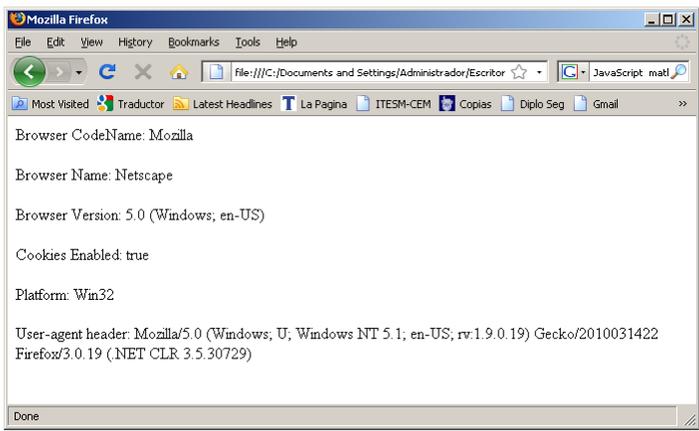


Lámina 72
Roberto Gómez C.



## El objeto String

- Usado para manipular textos
- Propiedades
  - Length
- Metodos
  - toUpperCase()
  - toLowerCase()
  - indexOf
  - lastIndexOf
  - charAt()
  - Split()
  - Substring()

Lámina 73 Roberto Gómez C.



## Ejemplos

- Ejemplo propiedad length

```
var txt="Hello world!";
document.write(txt.length);
```
- Ejemplo metodo toUpperCase

```
var txt="Hello world!";
document.write(txt.length);
```

Lámina 74 Roberto Gómez C.



## Objeto Date

- Usado para trabajar con fechas y tiempos.
- Los objetos son creados por el constructor Date( ).
- Cuatro formas de instanciar una fecha:
  - new Date() // current date and time
  - new Date(milliseconds) //milliseconds since 1970/01/01
  - new Date(dateString)
  - new Date(year, month, day, hours, minutes, seconds, milliseconds)

Lámina 75

Roberto Gómez C.



## Ejemplos instanciación date()

```
today = new Date()  
d1 = new Date("October 13, 1975 11:13:00")  
d2 = new Date(79,5,24)  
d3 = new Date(79,5,24,11,33,0)
```

Lámina 76

Roberto Gómez C.



## Asignar fechas

- Se asigna una fecha específica a un objeto

```
var myDate=new Date();  
myDate.setFullYear(2010,0,14);
```

- Asignando a un objeto la fecha que será en cinco días.

```
var myDate=new Date();  
myDate.setDate(myDate.getDate()+5);
```

Lámina 77

Roberto Gómez C.



## Comparando dos fechas

- Posible usar objeto Date para comparar dos fechas.

```
var myDate=new Date();  
myDate.setFullYear(2010,3,29);  
var today = new Date();  
if ( myDate> today ) {  
    alert("Hoy es una fecha anterior al 29 abril 2010");  
}  
else {  
    alert("Hoy es una fecha posterior al 29 abril 2010");  
}
```

Lámina 78

Roberto Gómez C.



## Más métodos

---

Método	Descripción
getHours( )	Hora del día (0-23)
getUTCHours( )	Hora del día
getMinutes( )	Minutos en una hora
getUTCMinutes( )	Minutos en una hora
getSeconds( )	Segundos en un minuto
getUTCSeconds( )	Segundos en un minuto
getMilliseconds( )	Miles de segundos en un segundo
getUTCMilliseconds( )	Miles de segundos en un segundo
getDay( )	Valor entero del día de la semana (0-6)
getUTCDay( )	Valor entero del día de la semana (0-6)
getDate( )	Día del mes
getUTCDate( )	Día del mes
getMonth( )	Mes del año
getUTCMonth( )	Mes del año
getFullYear( )	Año en cuatro dígitos
getUTCFullYear( )	Año en cuarto dígitos

Lámina 79
Roberto Gómez C.



## Métodos asignación

---

- setDate()** day of month
- setUTCDate()** day of month
- setMonth()** month of year (0 - 11)
- setUTCMonth()** month of year (0 - 11)
- setFullYear()** four digit year
- setUTCFullYear()** four digit year
- setHours()** hour of day (0 - 23)
- setUTCHours()** hour of day (0 - 23)

- setMinutes()** minutes into the hour
- getUTCMinutes()** minutes into the hour
- setSeconds()** seconds into the minute
- setUTCSeconds()** seconds into the minute
- setMilliseconds()** thousandths of seconds into the second
- setUTCMilliseconds()** thousandths of seconds into the second

Lámina 80
Roberto Gómez C.



## Otros métodos útiles

- **getTime()** milliseconds since midnight 1st January, 1970
- **UTC()** milliseconds since midnight 1st January, 1970 UTC
- **getTimezoneOffset()** minutes difference between local time and UTC
- **parse()** convert a date/time string into milliseconds since midnight 1st January, 1970
- **toLocaleString()** local date and time in string format
- **toString()** local date and time in string format
- **toGMTString()** converts a date to a string, using the Internet GMT conventions
- **toUTCString()** converts a date to a string, using UTC conventions

Lámina 81

Roberto Gómez C.



## Objeto arreglo

- Objeto utilizado para almacenar múltiples valores en una sola variable.
- Creando un arreglo
  - Creando un arreglo de la forma regular
 

```
var myCars=new Array(); // regular array (add an optional integer
myCars[0]="Saab"; // argument to control array's size)
myCars[1]="Volvo";
myCars[2]="BMW";
```
  - Arreglo condensado
 

```
var myCars=new Array("Saab","Volvo","BMW");
```
  - Arreglo literal
 

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

Lámina 82

Roberto Gómez C.



## Accediendo a un arreglo

- Se accede a un elemento del arreglo, referenciado el número.
- El índice empieza en cero.
- Ejemplo

```
var misCarros=new Array("Ford","Volvo","BMW");  
document.write(misCarros[0]);
```

- Salida ejemplo

Ford

Lámina 83

Roberto Gómez C.



## Modificando valores arreglo

- Asignar valor al elemento del arreglo que se desea modificar
- Ejemplo

```
var misCarros=new Array("Ford","Volvo","BMW");  
misCarros[0]="Peugeot";  
document.write(misCarros[0]);
```

- Salida

Peugeot

Lámina 84

Roberto Gómez C.



## Objeto Boolean

- Usado para convertir un valor no-booleano a un valor booleano (true/false)
- Creando un objeto booleano  

```
var myBoolean=new Boolean();
```
- Ejemplo creación objeto booleano con valor inicial de false y de true

```
var myBoolean=new Boolean();
var myBoolean=new Boolean(0);
var myBoolean=new Boolean(null);
var myBoolean=new Boolean("");
var myBoolean=new Boolean(false);
var myBoolean=new Boolean(NaN);
```

```
var myBoolean=new Boolean(1);
var myBoolean=new Boolean(true);
var myBoolean=new Boolean("true");
var myBoolean=new Boolean("false");
var myBoolean=new Boolean("Richard");
```

Lámina 85 Roberto Gómez C.



## Objeto Math

- Permite llevar a cabo tareas matemáticas.
- Incluye métodos y constantes matemáticas.
- Sintáxis propiedades/metodos de Math

```
var pi_valor=Math.PI;
var sqrt_valor=Math.sqrt(16);
```

Lámina 86 Roberto Gómez C.



## Constantes matemáticas

- Math.E
- Math.PI
- Math.SQRT2
- Math.SQRT1\_2
- Math.LN2
- Math.LN10
- Math.LOG2E
- Math.LOG10E

Lámina 87 Roberto Gómez C.



## Métodos matemáticos

- Math.abs(a) // the absolute value of a
- Math.acos(a) // arc cosine of a
- Math.asin(a) // arc sine of a
- Math.atan(a) // arc tangent of a
- Math.atan2(a,b) // arc tangent of a/b
- Math.ceil(a) // integer closest to a and not less than a
- Math.cos(a) // cosine of a
- Math.exp(a) // exponent of a
- Math.floor(a) // integer closest to and not greater than a
- Math.log(a) // log of a base e
- Math.max(a,b) // the maximum of a and b
- Math.min(a,b) // the minimum of a and b
- Math.pow(a,b) // a to the power b
- Math.random() // pseudorandom number in the range 0 to 1
- Math.round(a) // integer closest to a
- Math.sin(a) // sine of a
- Math.sqrt(a) // square root of a
- Math.tan(a) // tangent of a

Lámina 88 Roberto Gómez C.



## Ejemplo uso

- Ejemplo de redondeo

```
document.write(Math.round(4.7));
```

- Ejemplo de numero aleatorio

```
document.write(Math.random());
```

- Ejemplo numero aleatorio y de floor( )

```
document.write(Math.floor(Math.random()*11));
```

Lámina 89

Roberto Gómez C.



## Objeto RegExp

- Expresión regular: objeto que describe un patrón de caracteres.

- Sintaxis

```
var txt=new RegExp(pattern,modifiers);
```

o más simple:

```
var txt=/pattern/modifiers
```

- Pattern: especifica el patrón de una expresión.
- Modifiers: especifica si la búsqueda es global, sensible a mayúsculas, etc.

Lámina 90

Roberto Gómez C.



## Ejemplo búsqueda no sensible a mayúsculas.

- Búsqueda del string toto en un string

```
<html>
<body>

<script type="text/javascript">
var str = "Toto le hero es una pelicula francesa de 1991";
var patt1 = /toto/i;
document.write( str.match(patt1) );
</script>

</body>
</html>
```

Lámina 91

Roberto Gómez C.



## Ejemplo búsqueda global

- Búsqueda global del string es

```
<html>
<body>

<script type="text/javascript">

var str="Is this all there is?";
var patt1=/is/g;
document.write( str.match(patt1) );

</script>

</body>
</html>
```

Lámina 92

Roberto Gómez C.



## Tercer ejemplo

- Combinando

```
<html>
<body>

<script type="text/javascript">

var str="Is this all there is?";
var patt1=/is/gi;
document.write( str.match(patt1) );

</script>

</body>
</html>
```

Lámina 93

Roberto Gómez C.



## El metodo test( )

- Busca un valor específico dentro de un string.
  - Regresa true o false, dependiendo del resultado.
- Ejemplo:

```
<html>
<body>
<script type="text/javascript">
var patt1=new RegExp("e");

document.write(patt1.test("The best things in life are free"));
</script>

</body>
</html>
```

Lámina 94

Roberto Gómez C.



## El metodo exec()

- Busca un valor en un string, y regresa el texto del valor encontrado.
- Si no se encuentra nada, regresa null.
- Ejemplo: búsqueda carácter e

```
<html>  
<body>
```

```
<script type="text/javascript">  
var patt1=new RegExp("e");
```

```
document.write(patt1.exec("The best things in life are free"));  
</script>
```

```
</body>  
</html>
```

Lámina 95

Roberto Gómez C.



## JavaScript

Roberto Gómez Cárdenas

rogomez@itesm.mx

<http://homepage.cem.itesm.mx/rogomez>

Lámina 96

Roberto Gómez C.