Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA
UNIDAD CUAJIMALPA

JavaScript
Programación Web-Dinámico

Dr. Carlos Roberto Jaimez González

Introducción

- JavaScript básico:
  - Expresiones y tipos
  - Arreglos
  - Objetos
  - Funciones

- JavaScript en el browser
  - Ejecución de código JavaScript
  - Acceso y modificación de contenido HTML mediante JavaScript - conocido como Dynamic HTML (DHTML)

- Programas de ejemplo con JavaScript

Propiedades

- x.innerHTML. Contenido del elemento x.
- x.nodeName. Nombre del elemento x.
- x.nodeValue. Valor del elemento x.
- x.parentNode. Elemento padre del elemento x.
- x.childNodes. Elementos hijo del elemento x.
- x.attribute. Atributos del elemento x.

Propiedades

- getElementById
- getElementByName

x.innerHTML - the text value of x
x.nodeName - the name of x
x.nodeValue - the value of x
x.parentNode - the parent node of x
x.childNodes - the child nodes of x
x.attributes - the attributes nodes of x

Tipos de datos

- Boolean
- Number (sólo un tipo para números; a diferencia de Java, el cual tiene int, float, double, short, etc…)
- Array
- Associative array / Object
- Function
- Cuando se programa en un browser se tiene: Muchos tipos adicionales (por ejemplo, Elementos y Atributos HTML). Todos ellos son tipos de Object.

Expresiones

- 1.7 *// literal numérica*
- “JavaScript is Fun!” *// literal de cadena*
- true *// literal boleana*
- null *// literal del valor “null”*
- /java/ *// literal de expresión regular*
- { x:2, y:2 } *// literal de objeto*
- [2, 3, 5, 6] *// literal de arreglo*
- function(x) {return x*x} *// literal de función*
- i *// variable i*
- sum *// variable sum*

Principales características

- Muy útil para páginas interactivas
  - Validación, cálculos, mensajes, etc.
- Soportado por la mayoría de los web browsers
  - IE, Firefox, Opera, …(aunque el soporte es variable)
- ¿Dónde se utiliza?
  - Directamente en la página web
  - Mediante una referencia a un archivo separado (.js)
- Sintaxis parecida a C
- Permite acceso a la página HTML actual mediante DOM.
  - (Document Object Model)

Programación con JavaScript

- Manejo de eventos
- Uso de sentencias (como en C / Java)
- Uso de operadores
- Variables globales (default)
  - Locales (por ejemplo: var x = 1)
- Los tipos de datos pueden cambiar
  - Por ejemplo. x = 1; x = ‘Hello’;
- Definición de funciones
  - Para descomposición de problemas / reutilización
- Mensajes de alerta
- Acceso a elementos de la página mediante DOM
  - La página HTML es vista como un árbol de elementos

Document Object Model (DOM)

- Confusión: existen 2 Document Object Model
  - Legacy DOM
  - W3C DOM (Niveles 1 y 2 – no entraremos en detalle)

- Los 2 DOMs pueden hacer cosas similares, pero de manera diferente
  - Legacy DOM es conciso, pero utiliza nombres raros
  - W3C DOM no es tan conciso, pero tiene una convención de nombres consistente

Ejemplo: Función Factorial

```
<html>
   <head>
      <script language="JavaScript">
      function factorial(n) {
         //caso base
         if (n < 2) {
            return 1;
         }
         //caso recursivo
         else {
            return n * factorial(n-1);
         }
      }
      </script>
   </head>
```

Ejemplo: Función Factorial (continuación)...

```
  <body>
    <form>
      <p>
        <input type="text" id="c1"
onchange="res=factorial(this.value);
document.getElementById('c2').value= res;" />
      </p>
      <p>
        <input type="text" id="c2" />
      </p>
    </form>
  </body>
</html>
```

Función Factorial en acción

- La forma en acción con algunos resultados:

Factorial de 4 es 24
Factorial de 5 es 120
Factorial de 6 es 720

Función *eval*

- Cualquier cadena puede ser interpretada como código JavaScript utilizando la función ***eval***.

- Esto puede ser utilizado para escribir un intérprete de código JavaScript.

- Ver siguiente ejemplo.

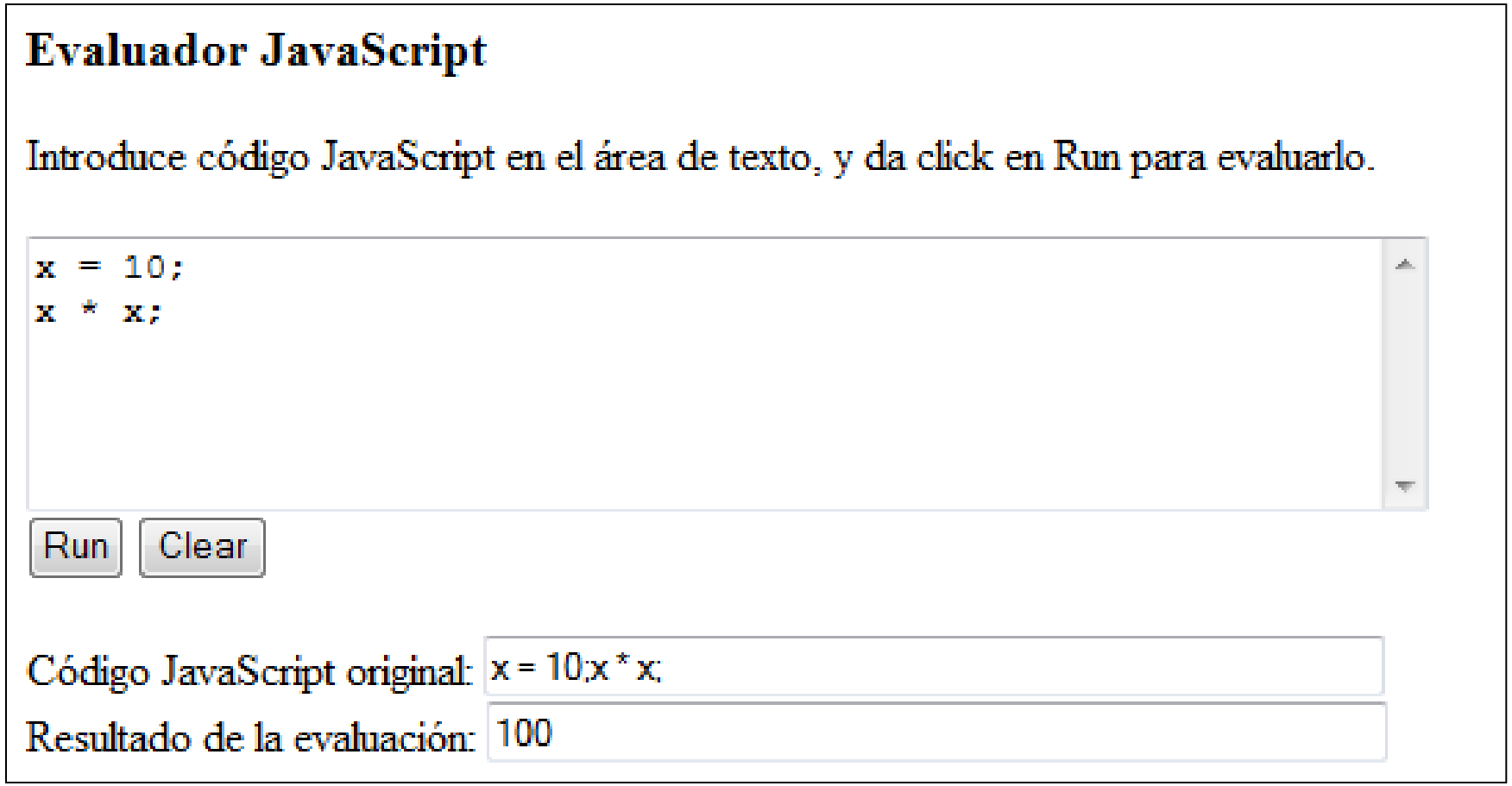
Ejemplo de la función *eval*

```
<body>
   <div>
      <h3>Evaluador JavaScript</h3>
      <p>Introduce c&oacute;digo JavaScript en el &aacute;rea
         de texto, y da click en Run para evaluarlo.</p>
   </div>

   <form>
      <p><textarea rows="6" cols="61" id="Input"></textarea> <br>
         <input type="button" value="Run" onclick="Run()" />
         <input type="reset" value="Clear" />
      </p>
   </form>

   C&oacute;digo JavaScript original:
   <input type="text" id="Output" size="50" /> <br>
   Resultado de la evaluaci&oacute;n:
   <input type="text" id="Eval" size="50" />
</body>
```

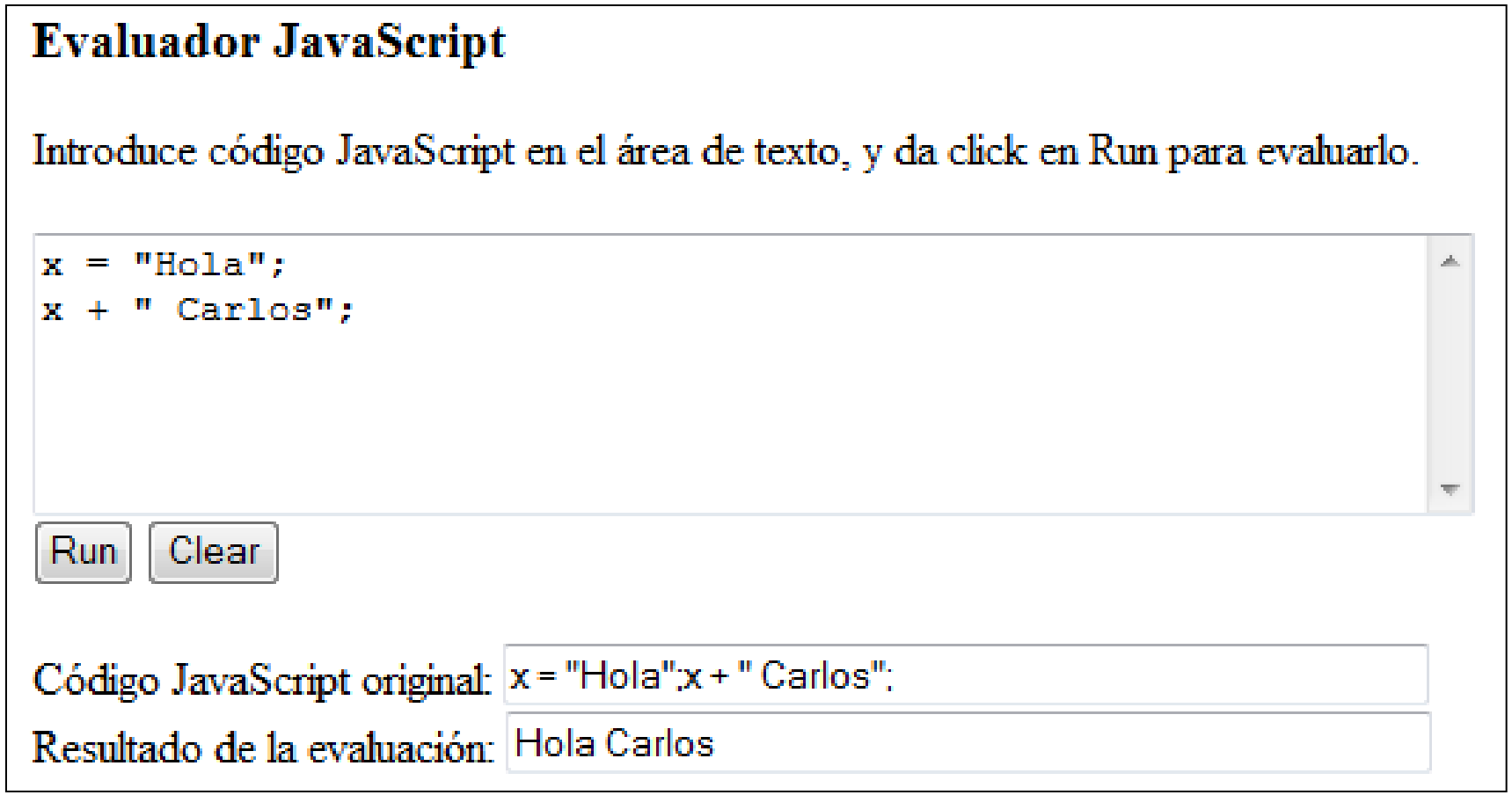
Ejemplo de la función *eval*

```
<script type="text/javascript">
    function Run(){
        //alert("Running");
        var ipNode = document.getElementById("Input");
        var opNode = document.getElementById("Output");
        opNode.value = ipNode.value;
        var evalNode = document.getElementById("Eval");

        try {
            evalNode.value = eval(ipNode.value);
        } catch (e) {
            evalNode.value = e;
        }
    }
</script>
```

Screenshot de ejemplo (Internet Explorer)



Screenshot de ejemplo (Internet Explorer)



Comunicación de múltiples ventanas

- El siguiente ejemplo ilustra la comunicación entre múltiples ventanas con JavaScript:
  - Una ventana hija es creada con cierto contenido. Esta ventana puede hablar con la ventana padre.
  - Esta funcionalidad puede ser usada con controles de Calendario, donde la comunicación entre múltiples ventanas es fundamental.

- Observa el uso de window.open para crear una nueva ventana (ventana hija).
  - El objeto document es empleado para escribir en esa vantana hija.
  - También es usado para escribir en la ventana padre (la que abrió a la ventana hija).

Ejemplo de comunicación de múltiples ventanas

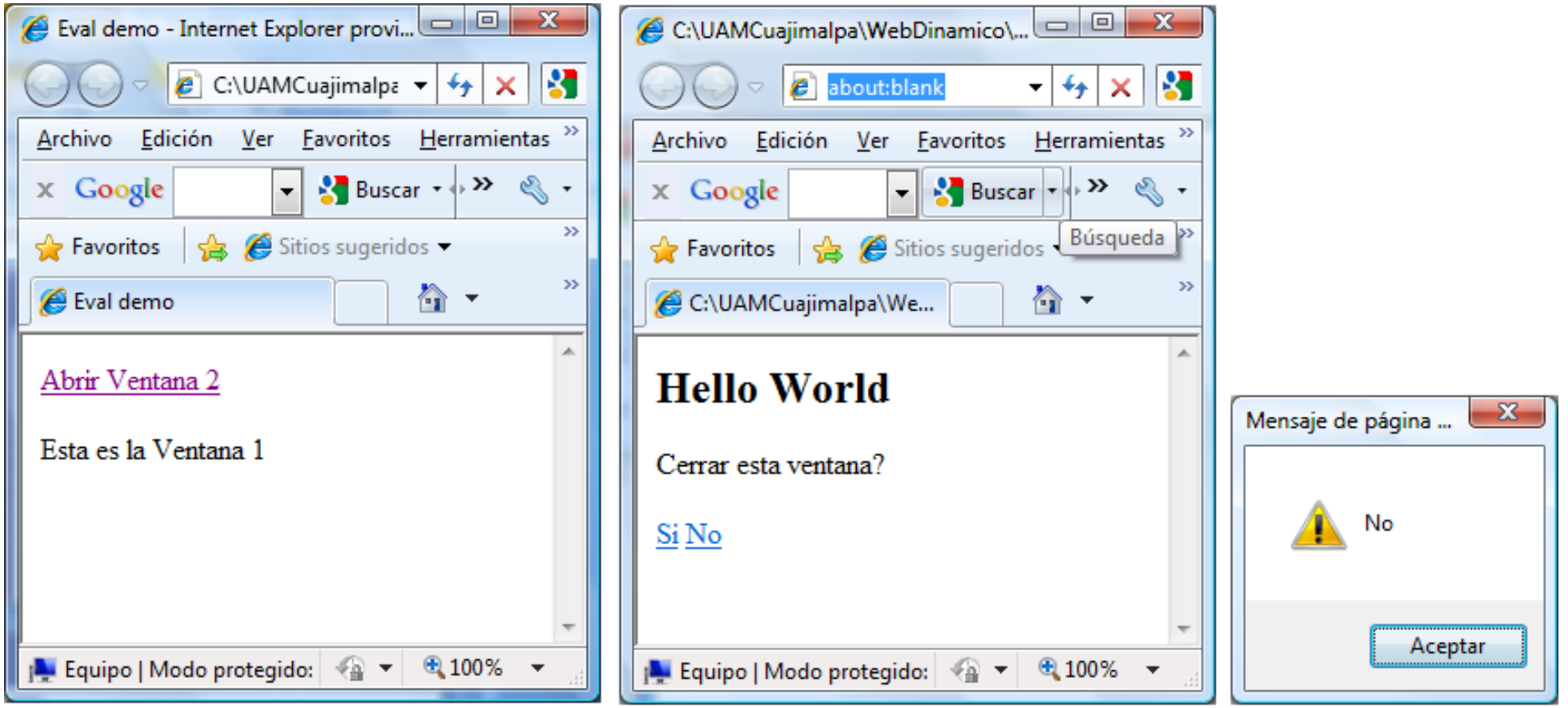
```
<body>
    <a href="javascript: hola()">Abrir Ventana 2</a>

    <div id="hi">
    </div>

    <div id="popup" class="invis" style="{display:none}">
        <p>Cerrar esta ventana?</p>
        <a href="javascript: opener.reply('Si'); close();">Si</a>
        <a href="javascript: opener.reply('No');">No</a>
    </div>

    <div id="popup2" style="{display:block}">
        <p>Esta es la Ventana 1</p>
    </div>

</body>
```

Ejemplo de comunicación de múltiples ventanas

```
<script type="text/javascript">
   function hola(){
     var w = window.open();
     var d = w.document;
     d.open();
     d.write("<h2>Hello World</h2>");
     var popup = document.getElementById("popup");
     d.write(popup.innerHTML);
     d.close();
   }

   function reply(a){
     alert(a);
     var hi = document.getElementById("hi");
     var pre = document.createElement("pre");
     hi.appendChild(pre);
     pre.appendChild(document.createTextNode("La respuesta fue "+a));
   }
</script>
```

Ejemplo de comunicación de múltiples ventanas

¿Alguien puede explicar que sucede en este ejemplo?



Resumen JavaScript

- DOM proporciona gran poder de acceso a elementos HTML, y a la modificación de una página HTML.
  - Observa el ejemplo de la Tabla de Contenido del Laboratorio.
- Se pueden buscar elementos por su ID.
- Manejo de eventos hacen las páginas interactivas.
- Uso de JavaScript para validación de formas.
- Ver también AJAX (más adelante en el curso).