

Fundamentos de páginas web dinámicas en el cliente con JavaScript.

I. OBJETIVOS

Que el estudiante:

1. Adquiera dominio de los elementos sintácticos básicos del lenguaje JavaScript.
2. Domine de las distintas formas de incluir código JavaScript dentro de un documento web.
3. Sea capaz de generar código HTML 5 desde secuencias de comando de JavaScript.
4. Pueda crear páginas web utilizando métodos básicos de interacción con el usuario.
5. Utilice métodos básicos de depuración de scripts utilizando las herramientas del navegador.

II. INTRODUCCION TEORICA

Introducir código JavaScript dentro de un documento HTML

Existen cuatro formas de agregar código JavaScript dentro de un documento web. Estas son:

1. Utilizando el elemento script

Este es el método básico para incluir código JavaScript dentro de un documento HTML. Los navegadores que reconocen scripts suponen que todo texto incluido dentro de estas etiquetas tiene que interpretarse como alguna forma de lenguaje de scripts. Casi todos suponen por defecto, que se trata de código JavaScript.

Tradicionalmente, la forma de indicar el lenguaje en un elemento script es haciendo uso del atributo *language*. De la siguiente forma:

```
<script language="JavaScript">  
  //Sentencias JavaScript  
</script>
```

De acuerdo con el consorcio W3C, en lugar del atributo *language*, debería utilizarse el atributo *type* para indicar el tipo MIME del lenguaje en uso. El tipo MIME de JavaScript es "text/javascript". De acuerdo a esta recomendación debería hacerse uso del siguiente código:

```
<script type="text/javascript">  
  //Sentencias JavaScript  
</script>
```

2. Creando scripts .js y vincularlos al documento con el elemento script

Esta es la solución ideal cuando se tienen códigos de script de muchos cientos de líneas o cuando se crean funciones JavaScript que pueden llegar a utilizarse en varias páginas. En estos casos incluir todo este código dentro de elementos script puede resultar poco productivo e ineficiente. En estos casos es más conveniente crear el script en un archivo .js independiente y luego vincularlo al documento HTML mediante el atributo *src* del elemento script.

Consideraciones a tener en cuenta al utilizar este método:

El archivo debe emplear formato de sólo texto y ser almacenado con extensión .js. Si utiliza un procesador de texto o un editor de texto asegúrese de que realmente fue guardado con esa extensión. No debe emplear el elemento script dentro del código JavaScript creado en el archivo .js.

La forma sintáctica de vincular desde el documento HTML es la siguiente:

```
<script type="text/javascript" src="script.js"></script>
```

Es importante recordar que script es un elemento HTML cerrado, por tanto debe ir acompañado con su respectiva etiqueta de cierre para que funcione correctamente.

Esta es la forma en que codificaremos en los ejemplos y ejercicios de las guías de práctica de este curso de Lenguajes Interpretados en el Cliente, salvo algunas pocas excepciones.

3. Empleando atributos HTML conocidos como manejadores de evento

La forma de crear páginas web interactivas es el uso de manejadores de eventos. Esto permite responder al usuario desde la página web como respuesta a las acciones que realiza en la página. Acciones como hacer clic sobre un botón del ratón, presionar una tecla, apuntar con el ratón a algún objeto específico en la página, etc. Una forma tradicional de aplicar manejadores de eventos en una página es utilizar atributos en ciertos elementos HTML. Algunos ejemplos de estos atributos son: *onclick*, *onmouseover*, *onchange*, *onkeypress*, etc.

Puede observar el siguiente ejemplo simple:

```
<body onUnload="decir_adios()">
```

🔔 **IMPORTANTE:** Aunque sigue siendo una forma aceptada por los distintos navegadores para implementar manejo de eventos, hoy en día se prefiere y se recomienda utilizar manejo de eventos estándar del nivel 2 del DOM, algo que profundizaremos en el transcurso del curso.

4. Haciendo uso de sintaxis de dirección pseudos-URL (javascript:)

Una dirección pseudo-URL permite crear instrucciones especiales con código JavaScript que es posible invocar desde la barra de direcciones del navegador. Sintácticamente estas direcciones comienzan con javascript: y a continuación se pueden agregar instrucciones JavaScript sintácticamente bien escritas. Por ejemplo: javascript:alert('Hola, APS I');

Las pseudos-URL se pueden utilizar en enlaces como valor para el atributo *href* y de esta forma ejecutar varias instrucciones a la vez. Como puede verse en el siguiente ejemplo:

```
<a href="javascript:x=5;y=7;alert('Suma:' + (x+y))">Sumar</A>
```

Ejemplos:

Los siguientes ejemplos ilustran cómo insertar código de JavaScript de cada una de las formas mencionadas anteriormente:

Utilizando etiquetas <script> ... </script>

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Ejemplo de JavaScript - Primera forma</title>
  <meta charset="utf-8" />
  <script type="text/javascript">
    var html;
    html = '<h3 style="text-align:center;font-family:Impact;font-size:16pt;
color:purple;">';
    html += "Una forma de incluir JavaScript dentro de una p&aacute;gina ";
    html += "web es a trav&eacute;s de las etiquetas &lt;script&gt;&lt;/script&gt;, ";
    html += "tal y como se ha hecho en este ejemplo.</h3>"; document.write(html);

  </script>
</head>
```

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

Vinculando un archivo independiente usando el atributo src de la etiqueta <script>

Archivo html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Ejemplo de JavaScript - Segunda forma</title>
  <meta charset="utf-8" />
  <script type="text/javascript" src="js/codigo.js"></script>
</head>
<body>

</body>
</html>
```

Archivo independiente de JavaScript: código.js

```
var html;
html = "<h3 style='text-align:center;font-family:Garamond;font-weight:bold;font-size:15pt;color:brown;'>";
html += "Otra forma de incluir JavaScript dentro de una p&acute;aquina ";
html += "web es usando el atributo src de la etiqueta <script></script>,";
html += "tal y como se ha hecho en este ejemplo</h3>"; document.write(html);
```

Utilizando los manejadores de eventos de algunas etiquetas HTML

```
<!DOCTYPE html>
<html lang="es">
<html>
<head>
  <title>Ejemplo de JavaScript - Tercera forma
</title> </head>
<body onload="alert('Hola desde JavaScript');">
  <h1 style="text-align:center;font-family:Georgia;font-size:15pt;color:olive">
    JavaScript te saludó
  </h1>
</body>
</html>
```

Utilizando sintaxis pseudo-URL

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Cuarto ejemplo</title>
  <meta charset="utf-8" />
  <style type="text/css">
    a:link {
      font-family:"Century Gothic";
      font-style:italic;
      font-weight:900;
      font-size:12pt;
      color:#00CD00;
      text-decoration:none;
    }
    a:visited {
      font-family:"Century Gothic";
      font-style:italic;
      font-weight:500;
      font-size:12pt;
      color:#CDBE70;
      text-decoration:none;
    }

    a:hover {
```

```

        font-family:"Century Gothic";
        font-style:italic;
        font-weight:900;
        font-size:12pt;
        color:#6959CD;
        text-decoration:underline overline;
    }
    a:active {
        font-family:"Century Gothic";
        font-style:italic;
        font-weight:700;
        font-size:12pt;
        color:#00E5EE;
        text-decoration:underline;
    }
</style>
</head>
<body>
    <a href="javascript:alert(';Me hiciste click!');">Hazme clic</a>
</body>
</html>

```

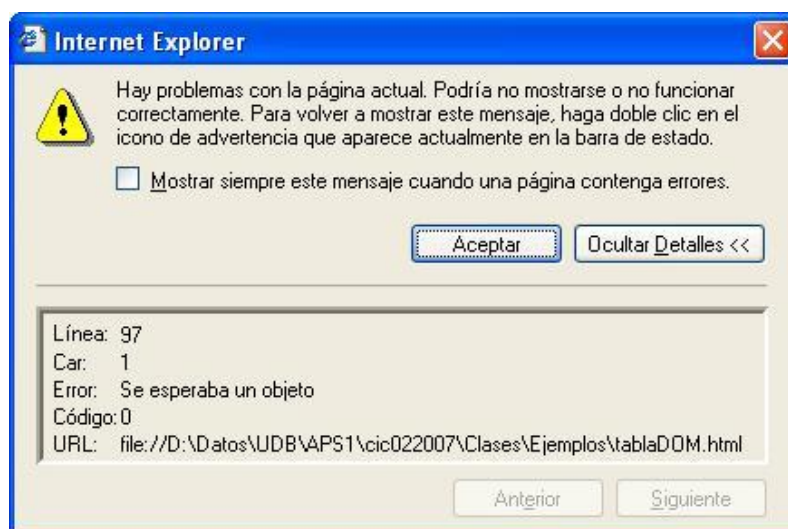
Depuración de errores con el navegador

Frecuentemente nos encontraremos con situaciones en donde será necesario corregir los scripts realizados con JavaScript. Es una tarea común en programación que la página web que contiene código JavaScript no se comporte de la manera esperada o no produzca el resultado que de acuerdo a nuestra lógica es el correcto. Generalmente, esto sucede cuando se produce algún error en la etapa de interpretación ejecutada por el navegador, ya sea debido a algún error de sintaxis o a un error en la lógica de programación. Casi todos los navegadores brindan algún mecanismo de depuración. A continuación se muestran algunos:

Si utiliza versiones antiguas de **Internet Explorer** (inferiores a la versión 9.0) notará que en la barra de estado aparece un pequeño icono de color amarillo de fondo, como símbolo de advertencia. Cuando aparezca este icono será porque existe algún error de sintaxis en el script que intenta ejecutar. Para poder leer la información acerca del problema que ocurre puede hacer doble clic en dicho icono para que se muestre en un diálogo información sobre el error. Observa las siguientes figuras:



Al hacer doble clic sobre este icono de advertencia debería aparecer un diálogo como el siguiente:

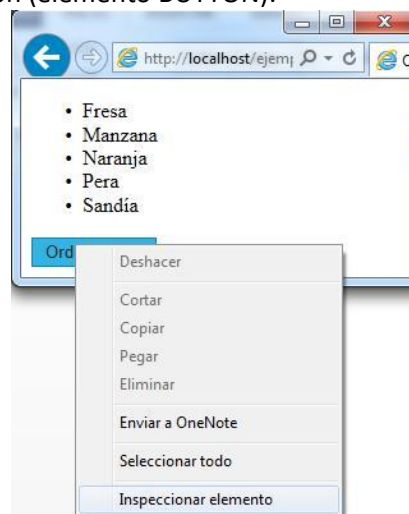


Observará que el diálogo anterior le informa sobre el número de línea en el que se ha producido el error, a

veces puede ser que sea en la línea anterior a la que se muestra, también se le informa del carácter y código de error generado. La descripción del error rara vez es de utilidad, de hecho en este ejemplo el problema es únicamente un paréntesis, como se puede ver en el código de este ejemplo:

```
function doCellModification(row,column,newValue){
    var rowNumber = parseFloat(row);
    var colNumber = parseFloat(column);
    var numberOfRowsInTable = theTable.rows.length;
    if((rowNumber >= 0) && colNumber >= 0)){
        if(rowNumber >= numberOfRowsInTable){
            alert("Imposible modificar celdas fuera de la tabla");
            return;
        }
        if(colNumber >= theTable.rows[rowNumber].cells.length){
            alert("Imposible modificar celdas fuera de la tabla");
            return;
        }
        theTable.rows[rowNumber].cells[colNumber].innerHTML = newValue;
    }
}
```

En caso de que utilice versiones de Internet Explorer 9.0 (o superiores) puede inspeccionar elementos de la página que está cargada en el navegador y examinar detenidamente cualquier problema que ocurra con la ejecución de código JavaScript. Para visualizar el explorador del DOM, proporcionado por Internet Explorer 11 (por poner un caso particular), haga clic derecho sobre cualquier elemento de la página. Para el caso de este ejemplo, se hace sobre un botón (elemento BUTTON).



En la barra lateral izquierda del explorador del DOM puede examinar un elemento de la página que no esté funcionando de acuerdo a lo esperado, como el caso del botón del ejemplo mostrado en la imagen anterior.

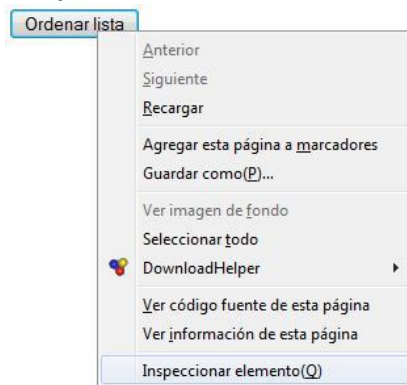


Al recargar la página y presionar el botón Ordenar lista se puede observar la notificación de un error, que deberá solventarse para que funcione correctamente. La interpretación de los errores y saberlos corregir es una habilidad que cada uno debe ir adquiriendo en base a experiencia y dedicación. Para algunos, resultará obvio, pero para otros puede que no sea así y que les tome más tiempo desarrollar esa habilidad.

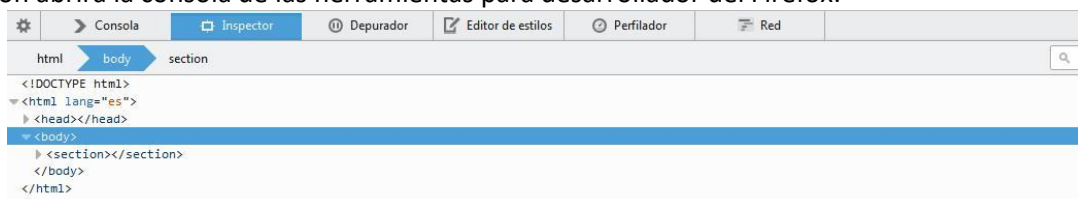


Como puede observar existen mensajes de advertencia o de error en la consola. En este caso se trata de un error de sintaxis.

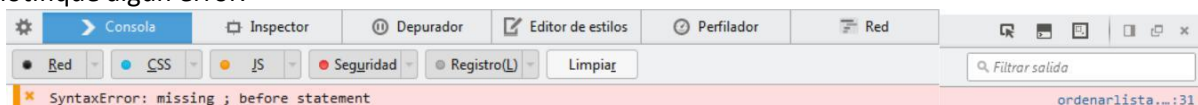
Si utiliza el **Mozilla Firefox 27.0** o superior la depuración resulta más fácil de realizar utilizando el **codemirror** que es la herramienta para desarrolladores (**DevTools**) que ofrece por defecto el navegador de Mozilla. Para activarla, se sigue un proceso similar, primero haciendo clic derecho sobre un elemento que no esté funcionando adecuadamente o que no esté produciendo el resultado esperado y a continuación en el menú desplegable se da clic sobre la opción **Inspeccionar elemento (O)**.



Esa acción abrirá la consola de las herramientas para desarrollador del Firefox:



Haciendo clic en la opción **Consola**, podrá observar alguna notificación de error en el código JavaScript. Si aparece en blanco actualice y de clic en el elemento asociado a la ejecución de código de JavaScript para que le notifique algún error.



Puede observar que el depurador le muestra el error, su descripción y el número de línea donde está el problema encontrado.

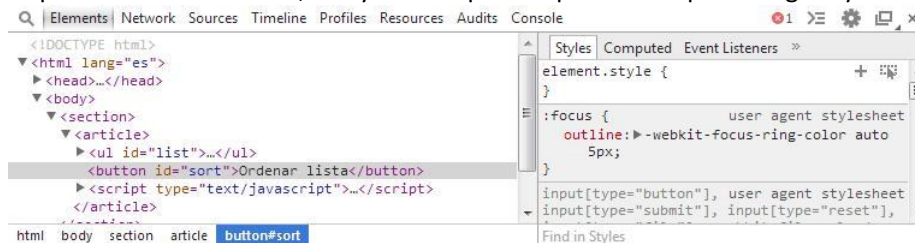
Si está utilizando **Chrome de Google**, puede invocar a las herramientas de desarrollador de Chrome (o **Chrome Developer Tools** o simplemente **DevTools**) para depurar código de JavaScript. El funcionamiento es idéntico y al invocarlo visualizará la siguiente interfaz de usuario donde puede explorar errores de JavaScript siempre en la opción **Consola**.



Al hacer clic en la opción Consola (Console) podrá visualizar los errores de JavaScript y las líneas donde se encuentran dichos errores, con alguna información adicional acerca del error.



En el caso de que utilice el navegador **Opera**, puede utilizar el Opera Dragonfly, la propuesta de este navegador para depurar errores de HTML, CSS y JavaScript. El aspecto del Opera Dragonfly es el siguiente:



Igual que con las propuestas de Firefox, Chrome e Internet Explorer se accede a través de los elementos de la página, haciendo clic derecho sobre ellos o invocando mediante método abreviado del teclado, pulsando las teclas CTRL+SHIFT+I. Para observar los errores posibles en JavaScript, debe hacer clic sobre la opción Consola (Console).



Hay que mencionar que en las últimas versiones de Opera, el aspecto del Dragonfly es casi idéntico al de las herramientas para desarrollador de Chrome. Esto es así, debido a que para la última versión de Opera se ha utilizado como núcleo el motor Chromiun, mismo que utiliza el navegador Chrome de Google.

Se plantea que investigue ud. cuál es la herramienta para desarrolladores del navegador Safari de las computadoras MAC.

Interacción básica con el usuario utilizando JavaScript

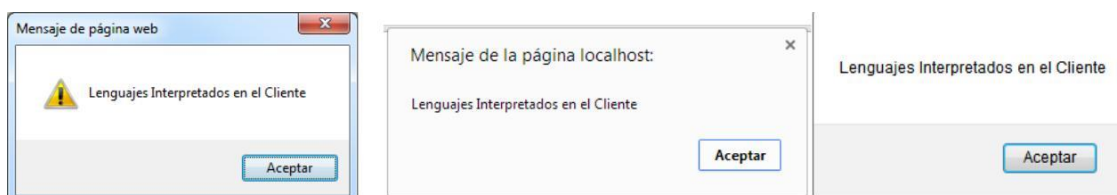
La forma más básica de interactuar con el usuario desde JavaScript es haciendo uso de los diálogos de entrada y salida simple proporcionados por los navegadores. Existen tres diálogos para entrada y salida simple. Estos son:

- Alerta (alert()),
- Confirmación (confirm()), y
- Solicitud de información (prompt()).

Veamos cada uno de ellos.

Diálogo de alerta

El cuadro de diálogo de alerta se invoca mediante el método **alert()** del objeto window, que conoceremos en futuras prácticas. Este diálogo es simple y muestra un mensaje breve utilizando una cadena de texto, un icono de advertencia y un único botón Aceptar como el que se muestra a continuación.



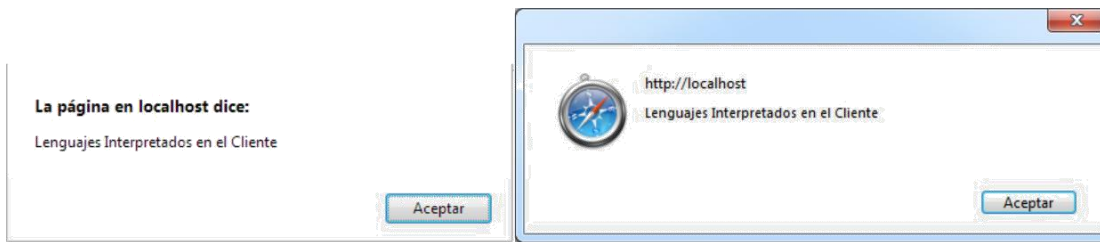


Figura 1. Ventanas de diálogo de alerta de JavaScript (alert) en los navegadores de uso más frecuente: Internet Explorer, Firefox, Chrome, Opera y Safari.

La sintaxis básica para un diálogo de alerta es:

```
window.alert(mensaje);
```

De forma abreviada se puede utilizar así:

```
alert(mensaje);
```

Donde mensaje es una cadena de texto, por lo tanto, si se introduce directamente el mensaje como un literal de cadena, debe ingresar el texto entre comillas simples o dobles. Si se utilizan variables declaradas deben ser de tipo cadena. Además, la cadena puede ser el resultado de la evaluación de una expresión, como por ejemplo, una concatenación. Cuando pasa un tipo de dato que no es cadena JavaScript lo convertirá en cadena.

Veamos algunos ejemplos:

```
alert(";Hola desde JavaScript!");
var usuario="Ricardo Ernesto";
alert("Hola " + usuario + ". Bienvenido a JavaScript");
```

Diálogo de confirmación

Un diálogo de confirmación utiliza el método ***confirm()*** del objeto Window. Este diálogo muestra una pequeña ventana con un mensaje para que el usuario responda si está de acuerdo, pulsando un botón Aceptar, o si no está de acuerdo, pulsando un botón Cancelar (OK y Cancel, en un navegador en inglés). El diálogo también muestra un icono de interrogación. El diálogo tiene un aspecto como el que se muestra a continuación:

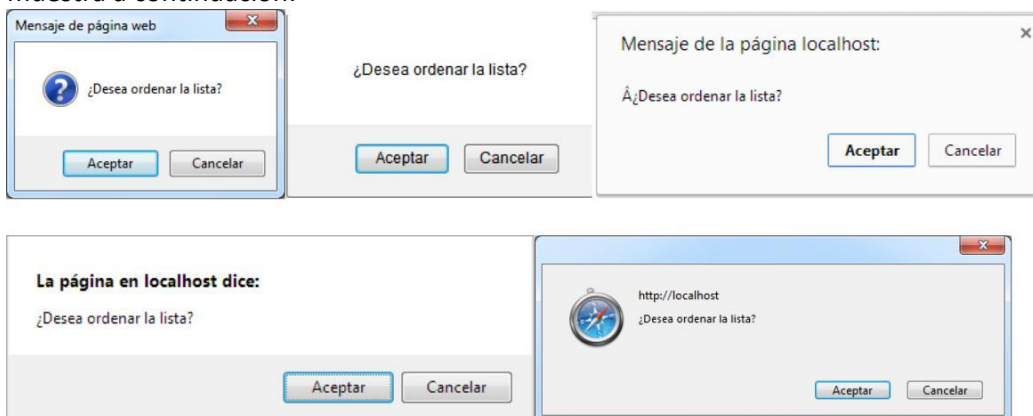


Figura 2. Ventanas de diálogo de confirmación (confirm) de los navegadores de uso más frecuente

La sintaxis básica de un diálogo de confirmación es:

```
window.confirm(mensaje);
```

O en forma resumida

```
confirm(mensaje);
```

Donde mensaje es cualquier variable válida de cadena de texto o expresión que al final de por resultado un valor de cadena. Este mensaje es el que aparecerá como pregunta en el mensaje de confirmación.

Ejemplo:

```
confirm("¿Deseas aprender JavaScript? ");
```

Diálogo de solicitud de información

Este diálogo utiliza el método ***prompt()*** del objeto Window. Esta ventana pequeña contiene un cuadro de texto donde el usuario debe escribir la información que se le solicita con un mensaje y un par de botones, uno para ingresar la información escrita y otro para cancelar la operación. Es importante mencionar que el dato ingresado por el usuario en este cuadro de texto, siempre será procesado en el script con un dato de tipo cadena de texto, sin importar, que se hayan introducido valores numéricos. El aspecto es el que se muestra a continuación:

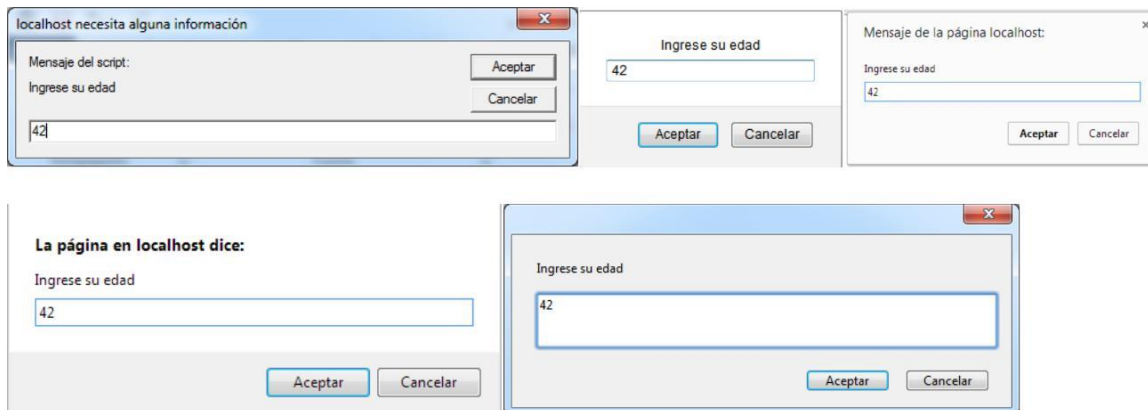


Figura 3. Ventanas de diálogo de solicitud de información de JavaScript (prompt) en Microsoft Internet Explorer 6.0 y Netscape 7.0, respectivamente.

El método ***prompt()*** tiene dos argumentos. El primero es una cadena de texto que se muestra como etiqueta para indicar al usuario el tipo de información solicitada y el segundo es un valor predeterminado que aparece en el cuadro de texto donde se escribe la información solicitada. El método **devuelve un valor de cadena** que contiene el valor que introduce el usuario en la solicitud. La sintaxis básica del diálogo solicitud de información es:

```
var result = prompt(mensaje, valor);
```

Nótese que para poder utilizar el valor devuelto por el método ***prompt()*** debe utilizarse una variable a la cual asignárselo. En la sintaxis anterior **mensaje** es la cadena de texto que aparece como indicación del tipo de información que se está solicitando y **valor** es una cadena con un valor predeterminado que aparecerá en el cuadro de texto del diálogo solicitud de información. Si no coloca un valor predeterminado en la llamada aparecerá por defecto, en algunos navegadores el valor **undefined**. Si no desea esto coloque como valor una cadena vacía. Como se muestra a continuación:

```
var edad = prompt("¿Cuál es tu color favorito? ", "");
```

Imprimir en la página web desde JavaScript

Con JavaScript se puede mandar a escribir texto, datos y código HTML de forma dinámica en una página web. Para poder hacer uso de esta característica debe hacer uso del método write del objeto document de JavaScript.

La sintaxis a utilizar para aplicar este método es la siguiente:

```
document.write(cadena o datos a imprimir);
```

Alternativamente, se puede utilizar también:

```
document.writeln(cadena o datos a imprimir);
```

Ejemplo:

```
document.write("Esta página fue modificada por última vez: " + document.lastModified);
```

Manejo básico de eventos

En JavaScript un evento es una acción que ocurre (o se dispara) automáticamente como resultado de una acción ocurrida dentro de la página web, generalmente provocada por la interacción del usuario, aunque existen casos en los que no es así.

Algunos eventos típicos son: la carga de una página, situar el puntero del ratón sobre un enlace, hacer clic sobre un botón, enviar un formulario, etc.

Los manejadores de eventos son atributos HTML que se colocan y son aceptados por algunos elementos. Por ejemplo: **onclick**, se utiliza para capturar un clic sobre un botón de formulario o sobre un enlace, **onsubmit**, se utiliza para capturar el envío de un formulario, **onmouseover**, para capturar cuando se ha posicionado el puntero del ratón sobre un objeto del documento (imagen, enlace, tabla, etc.)

Sintaxis de javascript

Es importante conocer los aspectos más importantes de la sintaxis del lenguaje JavaScript antes de ponerse a desarrollar. Los elementos de la sintaxis de JavaScript son comunes a los que proporcionan otros lenguajes:

- Variables
- Tipos de datos
- Operadores
- Sentencias
- Constantes
- Comentarios
- Estructuras de control
- Operadores
- Objetos

OBSERVACIÓN: No se dará en esta guía de práctica una explicación muy extensa de cada uno de estos elementos de sintaxis. La información ya ha sido detallada en la clase teórica.

Los aspectos de la sintaxis de JavaScript a considerar a la hora desarrollar scripts se detallan a continuación:

- JavaScript es un lenguaje sensible al uso de mayúsculas y minúsculas. Esto quiere decir, que hace distinción entre una y otra. Por ejemplo: no es lo mismo `var edad`, que `var Edad`. Se está haciendo referencia a variables diferentes. Del mismo modo la palabra reservada `var` debe digitarse (o escribirse) toda en minúsculas. Si se digita (o escribe) `Var` estará haciendo referencia a un identificador desconocido para el intérprete de JavaScript.
- Las instrucciones de JavaScript deben terminar en punto y coma. Aunque JavaScript permite que las instrucciones finalicen con un ENTER (retorno de carro), no debería dejarse llevar por este atajo. Es mucho mejor, terminar las instrucciones con el punto y coma.
- JavaScript es un lenguaje débilmente tipificado. Esto significa que no requiere que se defina un tipo de dato cuando se declaran variables. En lugar de ello, JavaScript decide el tipo apropiado para el dato en el momento en que lo va a operar o a utilizar en el contexto de un cálculo. Esto puede parecer una bondad,

pero si no se presta atención a la hora de realizar los guiones puede llevarse sorpresas inesperadas en los cálculos. Debe tenerse especial cuidado con el uso del operador +, que se utiliza tanto para concatenación como para suma de valores numéricos. Considere que si se pasan uno o dos valores de tipo cadena a un operador + este concatenará los operandos en vez de sumarlos.

- Puede utilizar dos tipos diferentes de comentarios en JavaScript, los mismos que se utilizan con el lenguaje C/C++. El comentario de una línea, que se representa con el doble signo de barra inclinada (//) y el comentario de bloque, que se representa con el signo de barra inclinada y asterisco, y que se cierra de forma inversa, colocando primero el asterisco y luego, la barra inclinada (/* */).
- Cuando se desea agrupar un grupo de instrucciones en un mismo bloque debe encerrarlas entre llaves ({ }). Esto será necesario, cuando se utilicen sentencias de control como el *if*, *else*, *switch*, *for*, *while*, *do-while* que contengan más de una instrucción.

Variables

Como en la mayor parte de lenguajes una variable tiene el propósito de almacenar datos que pueden ser modificados posteriormente. Todas las variables tienen un nombre conocido como identificador. En JavaScript las variables se declaran utilizando la palabra clave *var* seguida por el identificador de la variable. Como se muestra en los siguientes ejemplos:

```
var x;  
var peso;  
var nombre;
```

JavaScript permite la inicialización de variables al mismo tiempo que se hace la declaración, como se muestra a continuación:

```
var i = 0;  
var acumulador = 1;  
var mensaje = "";
```

Es recomendable no utilizar variables sin antes declararlas. La aparición de una variable del lado derecho de una asignación sin declararla previamente dará lugar a un error.

Tipos de datos

A pesar de que JavaScript no requiere que las variables se declaren como de un tipo de datos específico, hay que entender que una vez se le asigna un valor a una variable, en ese momento la variable pasa a ser de ese tipo de datos, aunque posteriormente ese tipo de dato pueda cambiarse si se le asigna a la variable un valor de otro tipo. JavaScript maneja tipos de datos básicos o primitivos (cadenas, números y booleanos) y tipos de datos compuestos o complejos (objetos, arreglos y funciones). La característica importante de los tipos de datos complejos es que pueden estar constituidos por valores básicos como cadenas, números o booleanos, pero también pueden estar constituidos por datos complejos.

Comentarios

Los comentarios son necesarios en la medida que los scripts se vuelven mucho más complejos y largos. JavaScript proporciona mecanismos para utilizar comentarios en sus scripts, que al igual que en C se dividen en dos categorías:

- a) Comentario de una sola línea. El comentario en una sola línea requiere la utilización de doble barra inclinada (//) y comenta todo lo que se coloque a continuación de estos caracteres, siempre que estén en

la misma línea.

b) Comentarios de bloque. Para utilizar el comentario de bloque debe abrirse la sección de comentario con barra inclinada seguida de un asterisco ("/*"), a continuación puede digitar el texto del comentario y, por último, debe cerrar la sección de comentario con un asterisco seguido de

III. PROCEDIMIENTO

Indicaciones: Crear una carpeta en la carpeta web de su servidor local (C:\wamp\www) con el nombre lic y dentro de crear otra carpeta a la que nombrará guia1 y dentro de esta todas las carpetas de los ejercicios que se le muestran a continuación:

- ejemplo1
- ejemplo2
- ejemplo3
- ejemplo4
- ejemplo5
- ejemplo6

Ejercicio 1: Cálculo de presupuesto en un hospital. Dado un valor de presupuesto, se calcula el total asignado a tres áreas del hospital: Dermatología, Traumatología y Pediatría.

Guión 1: presupuesto.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Presupuesto</title>
  <link rel="stylesheet" type="text/css" href="css/fonts.css" />
  <link rel="stylesheet" type="text/css" href="css/presupuesto.css" />
</head>
<body>
<section>
<article>
<table>
  <caption>ÁREAS</caption>
  <tr>
    <th>ÁREA</th>
    <th>PORCENTAJE DEL PRESUPUESTO</th>
  </tr>
  <tr>
    <td>Dermatología</td>
    <td> 40%</td>
  </tr>
  <tr>
    <td>Traumatología</td>
    <td>35%</td>
  </tr>
  <tr>
    <td>Pediatría</td>
    <td>25%</td>
  </tr>
```

```

</table>
<script type="text/javascript" src="js/presupuesto.js"></script>
</article>
</section>
</body>
</html>

```

Guión 2: presupuesto.js

```

var presupuesto = prompt('Ingrese el presupuesto anual','');
var derma,trau,pedi;
derma = presupuesto*0.40;
trau = presupuesto*0.35;
pedi = presupuesto*0.25;
document.write("<h1 class='moneyOff' onMouseOver='this.className=\"moneyOn\"'
onMouseOut='this.className=\"moneyOff\"'>El presupuesto asignado para Ginecología es:
$" + derma + "</h1><hr>");
document.write("<h1 class='moneyOff' onMouseOver='this.className=\"moneyOn\"'
onMouseOut='this.className=\"moneyOff\"'>El presupuesto asignado para Traumatología
es: $" + trau + "</h1><hr>");
document.write("<h1 class='moneyOff' onMouseOver='this.className=\"moneyOn\"'
onMouseOut='this.className=\"moneyOff\"'>El presupuesto asignado para Pediatría es: $"
+ pedi + "</h1><hr>");

```

Guión 3: fonts.css

```

/* Incluyendo la fuente Creative Zoo y todas sus variantes
*/ @font-face {
    font-family: 'CreativeZoo';
    src: url('../fonts/creativzoo.eot');
    src: url('../fonts/creativzoo.eot?#iefix') format('embedded-opentype'),
        url('../fonts/creativzoo.woff') format('woff'),
        url('../fonts/creativzoo.ttf') format('truetype'),
        url('../fonts/creativzoo.svg#CreativeZoo') format('svg');
    font-weight: normal;
    font-style: normal;
    font-variant: normal;
}

/* Incluyendo la fuente Sipll y todas sus variantes
*/ @font-face {
    font-family: 'Sipll';
    src: url('../fonts/sipll.eot');
    src: url('../fonts/sipll.eot?#iefix') format('embedded-opentype'),
        url('../fonts/sipll.woff') format('woff'),
        url('../fonts/sipll.ttf') format('truetype'),
        url('../fonts/sipll.svg#CreativeZoo') format('svg');
    font-weight: normal;
    font-style: normal;
    font-variant: normal;
}

```

Guión 4: presupuesto.css

```

body {
    background: rgba(121,143,59,1);
    background: -moz-linear-gradient(left, rgba(121,143,59,1) 0%,
    rgba(118,173,47,1) 100%);
    background: -webkit-gradient(left top, right top, color-stop(0%,
    rgba(121,143,59,1)), color-stop(100%, rgba(118,173,47,1)));
    background: -webkit-linear-gradient(left, rgba(121,143,59,1) 0%, rgba(118,173,47,1)
    100%);
    background: -o-linear-gradient(left, rgba(121,143,59,1) 0%, rgba(118,173,47,1) 100%);
    background: -ms-linear-gradient(left, rgba(121,143,59,1) 0%,
    rgba(118,173,47,1) 100%);
    background: linear-gradient(to right, rgba(121,143,59,1) 0%,
    rgba(118,173,47,1) 100%);
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#798f3b',
    endColorstr='#76ad2f', GradientType=1 );
}

```

```

    font-size:16px;
    margin:30px 10px;
}

caption {
    font-family:'CreativeZoo';
    font-weight:900;
    font-size:2.6em;
    color:Lavender;
    text-shadow: 0 0 2px #fff,
                 -1px -1px 0 hsl(80,70%,35%),
                 -2px -2px 1px hsl(80,70%,35%),
                 -2px -2px 2px hsl(80,10%,15%);
}

table {
    border:solid 2px navy;
    font-family:'CreativeZoo';
    margin:1em auto;
    width:48%;
}

th {
    background-color:ForestGreen;
    border:solid 2px navy;
    color:LightCyan;
    font-size:1em;
    font-weight:800;
    text-align:center;
}

td {
    background-color:DarkOrange;
    border:solid 2px navy;
    color:Bisque;
    font-size:1em;
    font-weight:400;
    text-align:center;
}

.moneyOn {
    border:outset 6px Cornsilk;
    background-color:Brown;
    text-align:center;
    font-family:'Sipll';
    font-weight:700;
    font-size:15pt;
    color:Cornsilk;
}

.moneyOff {
    background-color:MistyRose;
    border:inset 5px maroon;
    color:Sienna;
    font-family:'Sipll';
    font-size:15pt;
    text-align:center;
    font-weight:700;
}

hr {
    background-color:lavender;
    height:7px;
    width:98%;
}

```

Resultado:



Mensaje de la página localhost:

Ingrese el presupuesto anual

56000

Aceptar Cancelar

ÁREA	PORCENTAJE DEL PRESUPUESTO
Dermatología	40%
Traumatología	35%
Pediatría	25%

El presupuesto asignado para Ginecología es: \$22400.00

El presupuesto asignado para Traumatología es: \$19600.00

El presupuesto asignado para Pediatría es: \$14000.00

Ejercicio 2: Crear un documento html que mediante un script de JavaScript permita crear la tabla de multiplicar del número que un usuario ingrese mediante un diálogo prompt. Guión 1: tablamultiplicar.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8"/>
  <title>Tabla de Multiplicar</title>
  <link rel="stylesheet" href="css/tablamulti.css"/>
  <script src="js/multiplicar.js"></script>
</head>
<body>
</body>
</html>
```

Guión 2: tablamulti.css

```
body{
  background: -moz-linear-gradient(top, rgba(118,130,127,1) 0%, rgba(130,140,149,0.8)
51%, rgba(43,53,58,0.6) 100%); /* FF3.6+ */
  background: -webkit-gradient(linear, left top, left bottom, color-
stop(0%,rgba(118,130,127,1)), color-stop(51%,rgba(130,140,149,0.8)),
color-stop(100%,rgba(43,53,58,0.6))); /* Chrome,Safari4+ */
  background: -webkit-linear-gradient(top, rgba(118,130,127,1)
0%,rgba(130,140,149,0.8) 51%,rgba(43,53,58,0.6) 100%); /* Chrome10+,Safari5.1+ */
  background: -o-linear-gradient(top, rgba(118,130,127,1) 0%,rgba(130,140,149,0.8)
51%,rgba(43,53,58,0.6) 100%); /* Opera 11.10+ */
  background: -ms-linear-gradient(top, rgba(118,130,127,1) 0%,rgba(130,140,149,0.8)
51%,rgba(43,53,58,0.6) 100%); /* IE10+ */
  background: linear-gradient(to bottom, rgba(118,130,127,1) 0%,rgba(130,140,149,0.8)
51%,rgba(43,53,58,0.6) 100%); /* W3C */
  filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#76827f',
endColorstr='#992b353a',GradientType=0 ); /* IE6-9 */
  height:100%;
  font: Arial,Helvetica,"Liberation Sans";
```



```
font-size: 16px;
```

```
}
```

```
caption {
```

```
background-color: #EF7B28;
```

```
color: white;
```

```
font-family: helvetica, arial;
```

```
letter-spacing: 3px;
```

```

margin:6px 0;
text-align: center;
text-shadow: 0px -2px 0px #333,0px 2px 3px #666;
}

table {
background-color:#EFE828;
border-radius: 5px 5px 5px 5px;
margin-left:auto;
margin-right:auto;
text-align:center;
}

tr {
background: green;
}

th {
background-color:#335599;
color:white;
font-family:"Century Gothic";
font-size:12pt;
font-style:normal;
font-weight:900;
letter-spacing: 2px;
border: #d9d9d9 3px solid;
border-right:2px solid #333;
border-bottom:2px solid #333;
background: #006;
color:white;
padding: 4px 6px;
}

td {
border: white 3px solid;
border-right:2px solid #000;
border-bottom:2px solid #000;
background: green;
color:white;
font-family:"Century Gothic";
font-size:0.9em;
font-style:bold;
font-weight:500;
letter-spacing: 2px;
padding: 4px 6px;
text-align:center;
}

td:first-child {
-moz-border-radius: 10px 0 0 0;
-webkit-border-radius: 10px 0 0 0;
border-radius: 0px 0px 0px 0px;
}

td:last-child {
-moz-border-radius: 0 10px 0 0;
-webkit-border-radius: 0 0 0 0;
border-radius: 0px 0px 0px 0px;
}

```

Guión 3: multiplicar.js

```

/*
 * Materia: Programacion Web
 * Archivo: funciones.js
 * Descripción: Genera una tabla de multiplicar del
 *              número ingresado por el usuario, que
 *              debe ser un número entero entre 1 y 10
 *
 */
//Ingresando el número y capturándolo usando el método prompt()

```

```

var numero = prompt('Ingrese un número entero para obtener su tabla de multiplicar','');

//Construyendo la tabla del número ingresado
document.write("<table>\n\t");
document.write("<caption>Tabla del multiplicar del " + numero + "</caption>\n");
document.write("<thead>\n\t");
document.write("<tr>\n\t\t<th>Operación</th>\n\t\t<th>Producto</th>\n\t</tr>\n");
document.write("</thead>\n\t");
document.write("<tbody>\n\t");
document.write("<tr>\n\t\t<td>" + numero + " x 1 = </td>\n\t\t");
document.write("<td>" + (numero * 1) + "</td>\n\t</tr>\n\t");
document.write("<tr>\n\t\t<td>" + numero + " x 2 = </td>\n\t\t");
document.write("<td>" + (numero * 2) + "</td>\n\t</tr>\n\t");
document.write("<tr>\n\t\t<td>" + numero + " x 3 = </td>\n\t\t");
document.write("<td>" + (numero * 3) + "</td>\n\t</tr>\n\t");
document.write("<tr>\n\t\t<td>" + numero + " x 4 = </td>\n\t\t");
document.write("<td>" + (numero * 4) + "</td>\n\t</tr>\n\t");
document.write("<tr>\n\t\t<td>" + numero + " x 5 = </td>\n\t\t");
document.write("<td>" + (numero * 5) + "</td>\n\t</tr>\n\t");
document.write("<tr>\n\t\t<td>" + numero + " x 6 = </td>\n\t\t");
document.write("<td>" + (numero * 6) + "</td>\n\t</tr>\n\t");
document.write("<tr>\n\t\t<td>" + numero + " x 7 = </td>\n\t\t");
document.write("<td>" + (numero * 7) + "</td>\n\t</tr>\n\t");
document.write("<tr>\n\t\t<td>" + numero + " x 8 = </td>\n\t\t");
document.write("<td>" + (numero * 8) + "</td>\n\t</tr>\n\t");
document.write("<tr>\n\t\t<td>" + numero + " x 9 = </td>\n\t\t");
document.write("<td>" + (numero * 9) + "</td>\n\t</tr>\n\t");
document.write("<tr>\n\t\t<td>" + numero + " x 10 = </td>\n\t\t");
document.write("<td>" + (numero * 10) + "</td>\n\t</tr>\n\t");
document.write("</tbody>\n");
document.write("</table>");

```

Resultado:

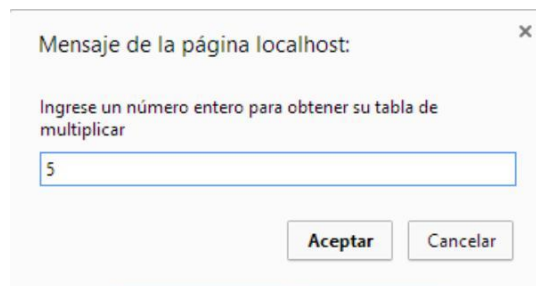


Tabla del multiplicar del 5	
Operación	Producto
5 x 1 =	5
5 x 2 =	10
5 x 3 =	15
5 x 4 =	20
5 x 5 =	25
5 x 6 =	30
5 x 7 =	35
5 x 8 =	40
5 x 9 =	45
5 x 10 =	50

Ejercicio 3: Creación de una calculadora que realiza las operaciones básicas: suma, resta, multiplicación, división y residuo.

Guión 1: calculadora.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8"/>
  <title>Calculadora Básica</title>
  <link rel="stylesheet"
href="css/calculadora.css"/> <script
src="js/calculadora.js"></script>
</head>
<body>
</body>
</html>
```

Archivo: estilos.css

```
body {
  background: rgb(247,247,121); /* Old browsers */
  background: -moz-linear-gradient(top, rgba(247,247,121,1) 0%,
  rgba(234,244,144,1) 100%); /* FF3.6+ */
  background: -webkit-gradient(linear, left top, left bottom, color-
  stop(0%,rgba(247,247,121,1)), color-stop(100%,rgba(234,244,144,1))); /*
  Chrome,Safari4+ */
  background: -webkit-linear-gradient(top, rgba(247,247,121,1)
  0%,rgba(234,244,144,1) 100%); /* Chrome10+,Safari5.1+ */
  background: -o-linear-gradient(top, rgba(247,247,121,1)
  0%,rgba(234,244,144,1) 100%); /* Opera 11.10+ */
  background: -ms-linear-gradient(top, rgba(247,247,121,1)
  0%,rgba(234,244,144,1) 100%); /* IE10+ */
  background: linear-gradient(to bottom, rgba(247,247,121,1)
  0%,rgba(234,244,144,1) 100%); /* W3C */
  filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#f7f779',
  endColorstr='#eaf490',GradientType=0 ); /* IE6-9 */
  height: 100%;
}

h1{
  background: rgba(0,155,125,0.8);
  color: #FFFFFFF;
  font: normal 42pt Arial,Helvetica,"Liberation Sans";
  padding: 6px 0;
  text-align: center;
  text-shadow: 0 1px 0 #ccc,
               0 2px 0 #c9c9c9,
               0 3px 0 #bbb,
               0 4px 0 #b9b9b9,
               0 5px 0 #aaa,
               0 6px 1px rgba(0,0,0,.1),
               0 0 5px rgba(0,0,0,.1),
               0 1px 3px rgba(0,0,0,.3),
               0 3px 5px rgba(0,0,0,.2),
               0 5px 10px rgba(0,0,0,.25),
               0 10px 10px rgba(0,0,0,.2),
               0 20px 20px rgba(0,0,0,.15);
}

.menu{
  overflow:hidden;
  width: 100%;
}

.menu a span{
  position:relative;
  top:40%;
}
```

```
.menu a:hover{
    background: #a71b15;
}

.menu li{
    display: inline;
    margin: 0;
}

.menu li a{
    background:#b72236;
    border-radius: 400px;
    -moz-border-radius: 400px;
    -webkit-border-radius: 400px;
    color: white;
    display:inline-block;
    height:100px;
    margin-right:5px;
    margin: 0;
    text-align:center;
    text-decoration: none;
    width:100px;
}

.menu li a:visited{
    color: white;
}

.menu ul{
    font: bold 14px Verdana;
    list-style-type: none;
    margin: 0;
    padding: 0;
    text-align: center;
}
```

Archivo: funciones.js

```

/* **** */
*  Materia: Programacion Web
*  Archivo: calculadora.js
*  Descripción: Realizar operaciones básicas aritméticas.
*
* **** */
*/ //Ingresar los datos de los números a operar
var op1 = prompt('Introduzca el primer numero:', ''); var
op2 = prompt('Introduzca el segundo numero:', '');

//Creando el documento HTML con puro JavaScript
haciendo //uso del método write() del objeto document
document.write("<h1>Operaciones básicas</h1><hr />");
document.write("<div class='menu'>");
document.write("<ul>");
document.write("<li><a href=\"javascript:void(0)\" onclick=\"alert(parseInt(op1)
        \ + parseInt(op2))\"><span>Sumar</span></a></li>");
document.write("<li><a href=\"javascript:void(0)\" onclick=\"alert(parseInt(op1)
        \ - parseInt(op2))\"><span>Restar</span></a></li>");
document.write("<li><a href=\"javascript:void(0)\" onclick=\"alert(parseInt(op1) \
        * parseInt(op2))\"><span>Multiplicar</span></a></li>"); document.write("<li><a
        href=\"javascript:void(0)\" onclick=\"alert(parseInt(op1) \
        /
        parseInt(op2))\"><span>Dividir</span></a></li>");
document.write("<li><a href=\"javascript:void(0)\" onclick=\"alert(parseInt(op1) \
        % parseInt(op2))\"><span>Residuo</span></a></li>");

document.write("</ul>");
document.write("</div>");

```

Resultado:

<p>Introduzca el primer numero:</p> <input type="text" value="15"/>	<p>Introduzca el segundo numero:</p> <input type="text" value="6"/>
<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>	<input type="checkbox"/> Prevenir esta página desde la creación de cuadros de diálogo adicionales <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>

Operaciones básicas

Sumar

Restar

Multiplicar

Dividir

Residuo

21

9

90

2.5

Ejercicio 4: Realización de un calculador de áreas geométricas, en base a un tipo de área seleccionada y a las medidas ingresadas de acuerdo al área ingresada. Las áreas a considerar serán: círculo, triángulo y rectángulo.

Guión 1: calculoareas.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8"/>
  <title>Cálculo de áreas</title>
  <link rel="stylesheet" href="css/areas.css"/>
</head>
<body>
<section id="menu">
  <nav>
    <ul>
      <li>
        <a href="circulo.html">Area de Circulo</a>
      </li>
      <li>
        <a href="triangulo.html">Area de Triangulo</a>
      </li>
      <li>
        <a href="rectangulo.html">Area de Rectangulo</a>
      </li>
    </ul>
  </nav>
</section>
</body>
</html>
```


Guión 2: círculo.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Área círculo</title>
```

```

    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/areas.css"/>
    <script type="text/javascript" src="js/circulo.js"></script>
</head>
<body>
<nav>
<ul>
    <li><a href="index.html">Regresar</a></li>
</ul>
</nav>
<section id="figura">
    <div class="circulo"></div>
</section>
</body>
</html>

```

Guión 3: triangulo.html

```

<!DOCTYPE html>
<html lang="es">
<head>
    <title>Área triángulo</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/areas.css"/>
    <script type="text/javascript" src="js/triangulo.js"></script>
</head>
<body>
<nav>
    <ul>
        <li><a href="index.html">Regresar</a></li>
    </ul>
</nav>
<section id="figura">
    <div class="triangulo"></div>
</section>
</body>
</html>

```

Guión 4: rectángulo.html

```

<!DOCTYPE html>
<html lang="es">
<head>
    <title>Área rectángulo</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/areas.css"/>
    <script type="text/javascript" src="js/rectangulo.js"></script>
</head>
<body>
<nav>
<ul>
    <li><a href="index.html">Regresar</a></li>
</ul>
</nav>
<section id="figura">
    <div class="cuadrado"></div>
</section>
</body>
</html>

```

Guión 5: areas.css

```
* {  
    margin:0;  
}  
  
body {  
background: -moz-linear-gradient(left bottom, rgb(153,217,234) 40%, |  
rgb(36,147,174) 70%, rgb(17,68,81) 95%);
```

```

        background: -ms-linear-gradient(left bottom, rgb(153,217,234) 40%,
rgb(36,147,174) 70%, rgb(17,68,81) 95%);
        background: -o-linear-gradient(left bottom, rgb(153,217,234) 40%,
rgb(36,147,174) 70%, rgb(17,68,81) 95%);
        background: -webkit-linear-gradient(left bottom, rgb(153,217,234) 40%,
rgb(36,147,174) 70%, rgb(17,68,81) 95%);
        background: linear-gradient(left bottom, rgb(153,217,234) 40%,
rgb(36,147,174) 70%, rgb(17,68,81) 95%);
        color:white;
        font-family:Arial;
        font-size:16px;
        margin:0;
        text-shadow:1px 1px 0 rgba(0,0,0,.3);
    }

div {
    display: block;
    height: 100px;
    margin: 0 auto;
    width: 100px;
}

h1 {
    text-align:center;
}

nav ul {
    font-size:0;
    padding:0;
}

nav ul li {
    background:rgba(0,0,0,.5);
    display: inline-block;
    font-size:16px;
    height:50px;
    line-height:50px;
    padding:10px 0;
    vertical-align: top;
    width:33%;
}

nav ul li:hover{
    background:rgba(0,0,0,.15);
}

nav ul li img {
    height:50px;
}

nav ul li a {
    color: #DDD;
    display: block;
    font-weight:bold;
    text-align:center;
    text-transform:uppercase;
}

#menu{
    background: -moz-linear-gradient(left bottom, rgb(39,30,107) 40%,
rgb(11,6,43) 70%, rgb(2,2,10) 85%);
    background: -ms-linear-gradient(left bottom, rgb(39,30,107) 40%,

```

```
| rgb(11,6,43) 70%, rgb(2,2,10) 85%);  
|     background: -o-linear-gradient(left bottom, rgb(39,30,107) 40%,
```

|

```

rgb(11,6,43) 70%, rgb(2,2,10) 85%);
    background: -webkit-linear-gradient(left bottom, rgb(39,30,107) 40%,
rgb(11,6,43) 70%, rgb(2,2,10) 85%);
    background: linear-gradient(left bottom, rgb(39,30,107) 40%, rgb(11,6,43)
70%, rgb(2,2,10) 85%);
    margin:0 auto;
    max-width:1000px;
}

.circulo{
    border-radius: 75px;
    -moz-border-radius: 75px;
    -ms-border-radius: 75px;
    -o-border-radius: 75px;
    -webkit-border-radius: 75px;
    background: navy;
}

.cuadrado {
    background-color: #ccc;
    width: 15%
}

.triangulo {
    border-color: transparent transparent yellow transparent;
    border-style: solid;
    border-width: 40px;
    height: 0;
    width: 0px;
}

```

Guión 5: circulo.js

```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 * Materia: Programacion Web *
 * Archivo: circulo.js *
 * Uso: Calcular el área de un círculo. *
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
var PI=3.1415926535;
var radio = prompt('Introduzca el radio del círculo:', '');
var area;
area = PI*radio*radio;
document.write("<header><h1>El área del círculo es: " + area + "</h1><hr
/><br | /></header>");

```

Guión 6: triangulo.js

```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 * Materia: Programacion Web *
 * Archivo: triangulo.js *
 * Uso: Calcular el área de un triángulo. *
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
var base = prompt('Introduzca la base del triángulo', '');
var altura = prompt('Introduzca la altura del triángulo', '');
var area;
area = (base*altura)/2;
document.write("<header><h1>El área del Triángulo es: " + area + "</h1><hr
/><br | /></header>");

```



```

* Materia: Programacion Web
* Archivo: rectangulo.js
* Uso: Calcular el área de un rectángulo.
*
* * * * *
var base= prompt('Introduzca la base del rectángulo','');
var altura = prompt('Introduzca la altura del rectángulo','');
var area;
area=base*altura
document.write("<header><h1>El área del rectángulo es: " + area +
"</h1><hr /><br /></header>");

```

Resultado:



Ejercicio 5: Cálculo del número de días para el mes y año seleccionado.

Guión 1: diasmes.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Días de cada mes</title>
  <script type="text/javascript" src="js/calculodias.js"></script>
</head>
<body>
</body>
</html>

```

Guión 2: calculodias.js

```

var mes = prompt("Ingresa el mes: ", "");
var year = prompt("Ingresa el año: ", "");
var diasdelmes;
var bisiesto = false;
if(year%4 == 0 && (year%100 != 0 || year%400 == 0)){
  bisiesto = true;
}
switch(mes) {
  case "Enero":

```

```

        case "enero":
        case "Marzo":
        case "marzo":
        case "Mayo":
        case "mayo":
        case "Julio":
        case "julio":
        case "Agosto":
        case "agosto":
        case "Octubre":
        case "octubre":
        case "Diciembre":
        case "diciembre":
            diasdelmes = 31;
            break;
        case "Abril":
        case "abril":
        case "Junio":
        case "junio":
        case "Septiembre":
        case "septiembre":
        case "Noviembre":
        case "noviembre":
            diasdelmes = 30;
            break;
        case "Febrero":
        case "febrero":
            if(!bisiesto)
                diasdelmes = 28;
            else
                diasdelmes = 29;
            break;
        default:
            alert("El mes " + mes + " no existe");
            mes = "No existe";
            diasdelmes = 0;
    }
    if(bisiesto == true){
        var msg = "El año " + year + " es bisiesto. El mes de " + mes + " tiene " +
        diasdelmes + " días.";
    }
    else{
        var msg = "El mes de " + mes + " tiene " + diasdelmes + " días.";
    }
    alert(msg);

```

Ejercicio 6: Cálculo de la prima y la cuota a pagar por una casa, solicitando al usuario su ingreso salarial por mes.

Guión 1: cuotacasa.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Cuota prima de casa</title>
    <link rel="stylesheet" type="text/css" href="css/tittle.css">
</head>
<body>
<script type="text/javascript">

```

```
var salario, preciocasa, prima,  
    cuotamensual; var annios;  
salario = parseFloat(prompt('Ingrese su salario mensual',''));  
annios = 12*15; //numero de años a pagar por la casa
```

```

preciocasa = parseFloat(prompt('Ingrese el precio de la casa',''));
if(salario <= 400){
    if(preciocasa > 8000){
        alert('Lo siento el valor de la casa es muy alto para su sueldo');
    }
    else{
        prima = (preciocasa*0.1);
        cuotamensual = (preciocasa-prima)/años;
        document.write("<h1 class='moneyOff' onMouseOver='this.className=\"moneyOn\"'
onMouseOut='this.className=\"moneyOff\"'>La prima que usted debe pagar por la casa es = $ " +
Math.round(prima * Math.pow(10,2)) / Math.pow(10,2) + "</h1><hr>");
        document.write("<h1 class='moneyOff' onMouseOver='this.className=\"moneyOn\"'
onMouseOut='this.className=\"moneyOff\"'>La cuota mensual a pagar durante 30 años es de = $ "
+ Math.round(cuotamensual * Math.pow(10,2)) / Math.pow(10,2) + "</h1><hr>");
    }
}
else{
    prima = (preciocasa*0.2);
    cuotamensual = (preciocasa-prima)/años;
    document.write("<h1 class='moneyOff' onMouseOver='this.className=\"moneyOn\"'
onMouseOut='this.className=\"moneyOff\"'>La prima que usted debe pagar por la casa es
= $ " + Math.round(prima * Math.pow(10,2)) / Math.pow(10,2) + "</h1><hr>");
    document.write("<h1 class='moneyOff' onMouseOver='this.className=\"moneyOn\"'
onMouseOut='this.className=\"moneyOff\"'>La cuota mensual a pagar durante 15 años es de
= $ " + Math.round(cuotamensual * Math.pow(10,2)) / Math.pow(10,2) + "</h1><hr>");
}
}
</script>
</body>
</html>

```

Guión 2: title.css

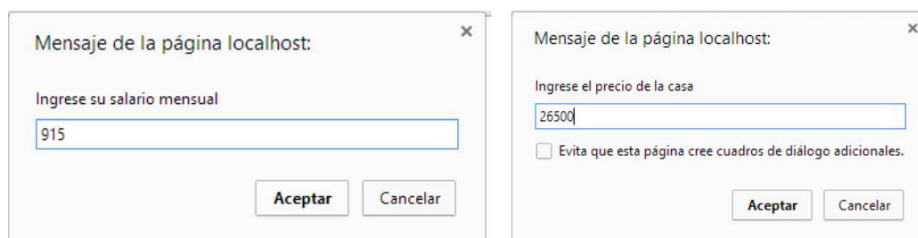
```

.moneyOn {
    border: outset 6px DarkViolet;
    background-color: Thistle;
    text-align: center;
    font-family: "Franklin Gothic Medium";
    font-weight: 700;
    font-size: 15pt;
    color: BlueViolet;
}

.moneyOff {
    border: inset 5px maroon;
    background-color: MistyRose;
    text-align: center;
    font-family: "Franklin Gothic Medium";
    font-weight: 700;
    font-size: 15pt;
    color: Sienna;
}

```

Resultado:



La prima que usted debe pagar por la casa es = \$ 5300

La cuota mensual a pagar durante 15 años es de = \$ 117.78

IV. DISCUSION DE RESULTADOS

1. Investigue el uso de document.write(), y con dicho método, genere la estructura del siguiente documento:



Recuerde que la maquetación se hará con CSS3.

V. INVESTIGACION COMPLEMENTARIA

1. Investigue a qué se le llama JavaScript no invasivo, no obstructivo o no intrusivo (unobtrusive JavaScript). Muestre un ejemplo.
2. Investigue qué es Null y undefined en JavaScript

VI. BIBLIOGRAFIA

Flanagan, David. JavaScript La Guía Definitiva. 1ra Edición. Editorial ANAYA Multimedia. 2007. Madrid, España.
Powell, Thomas / Schneider, Fritz. JavaScript Manual de Referencia. 1ra Edición. Editorial McGraw-Hill. 2002. Madrid, España.

McFedries, Paul. JavaScript Edición Especial. 1ra Edición. Editorial Prentice Hall. 2002. Madrid, España.

Tom Negrino / Dori Smith. JavaScript & AJAX Para Diseño Web. 6ª Edición. Editorial Pearson – Prentice Hall. 2007. Madrid España.