



DESARROLLO WEB EN ENTORNO CLIENTE

JAVASCRIPT PROGRAMACIÓN
AVANZADA 2.0

INTRODUCCIÓN

Esta segunda práctica corresponde al tema 4 del manual que estamos utilizando como guía. En este tema se ven las funciones, se habla del ámbito de las variables, de las sentencias break y continue y de otras estructuras de control (while, do...while, switch). Hay algunos puntos importantes que no vienen incluidos en el manual cómo es el tratamiento de los objetos predefinidos del navegador: window, document, history, navigator, screen y location, los vamos a ver también en esta práctica, para ello os incluyo un nuevo manual que además de servir de ampliación para lo visto hasta ahora, os servirá de referencia para ver algunos métodos y propiedades de estos objetos. También vemos con más profundidad el objeto Date, ya que es uno de los más utilizados en las aplicaciones de JavaScript.

PRACTICA 1- Programación Avanzada

En el tema 4 veréis, como he comentado anteriormente, las funciones, las sentencias break y continue y las estructuras de control: while, do..while y switch. Estas estructuras son sencillas de utilizar y están bien explicadas en el tema. Respecto a las funciones, sabemos cómo se utilizan, vamos a ver cómo llamarlas. Existen diferentes maneras de llamar a una función:

- Desde Javascript: Lo más correcto es crear una función en la cabecera del programa, así nos aseguramos que cuando cargamos la página este ya accesible. La llamada a la función estará en el cuerpo del programa dentro de las etiquetas <script> y </script>. Entre las etiquetas debemos poner el nombre de la función seguido del paréntesis y dentro de estos, si son necesarios, los argumentos para ejecutarla.

```
<html>
  <head>
    <title>ejemplo llamada función </title>
    <script type="text/javascript">
      function nom_fun([argumentos]) {
        // instrucciones a realizar
      }
    </script>
  </head>
  <body>
    <script type="text/javascript">
      nom_fun([argumentos])
    </script>
  </body>
</html>
```

- Desde HTML: La llamada a la función la realizamos como si fuese un atributo de HTML. Es habitual utilizar este sistema cuando se producen los eventos onload, onclick, etc.

```
<html>
  <head>
    <title>ejemplo llamada función </title>
    <script type="text/javascript">
      function nom_fun([argumentos]) {
        // instrucciones a realizar
      }
    </script>
  </head>
  <body onload="nom_fun([argumentos])">
  </body>
```

</html>

El punto 4.2 , ámbito de las variables, lo leéis detenidamente pues suele ser fuente de algunos errores cuando se trabaja con funciones ya que si no está accesible la variable de la función, tal como está explicado en los ejemplos, no podrá ejecutarse correctamente.

Como ampliación al tema os adjunto otro ejemplo de la estructura switch que puede ser útil en el caso de que varios valores deban ejecutar las mismas instrucciones.

<html>

<head>

<title>ejemplo de javascript </title>

<script type="text/javascript">

var dia=prompt("dia de la semana:", "");

switch(parseInt(dia)){

case 1:

case 2:

case 3:

case 4:

case 5:

document.write("Es dia laboral");

break;

case 6:

case 7:

document.write("Es fin de semana");

break;

default:

document.write("No has introducido un número de dia correcto");

break;

}

</script>

</head>

<body>

<p>La página que estás viendo es un ejemplo de la estructura Switch

</body>

</html>

La expresión entre paréntesis que evalúa debe ser entera. Si por el contrario lo que evaluamos es una cadena, recordad ponerla entre comillas:

<html>

<head>

<title>ejemplo de javascript </title>

<script type="text/javascript">

var dia=prompt("dia de la semana:", "");

switch(dia){

case "lunes":

case "martes":

case "miércoles":

case "jueves":

case "viernes":

document.write("Es dia laboral");

break;

case "sábado":

case "domingo":

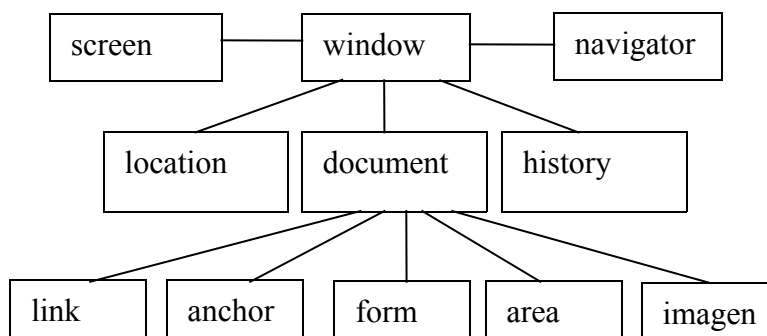
```

        document.write("Es fin de semana");
        break;
    default:
        document.write("No has introducido un número de día correcto");
        break;
    }
</script>
</head>
<body>
    <p>La página que estás viendo es un ejemplo de la estructura Switch
</body>
</html>

```

JavaScript aunque no sea un lenguaje orientado a objetos como puede ser Java, si que trabaja con objetos. Realmente ya trabajamos con objetos en la práctica anterior, la clase String, la clase Array, en realidad son clases de JavaScript. Tienen métodos, propiedades, etc. Estos son objetos predefinidos de JavaScript pero aparte de estos objetos tenemos los objetos predefinidos de navegador: window, document, history, navigator y screen, con sus métodos y propiedades, que son los que nos van a permitir realizar aplicaciones más completas en este lenguaje. Como referencia para el estudio de estos objetos debéis leer el punto 15 (Objetos predefinidos del navegador) del nuevo tutorial que os he colgado, veréis los métodos y propiedades de cada uno de ellos, no obstante en Internet podéis obtener mucha más información sobre cada uno de ellos.

Los navegadores acceden a los objetos siguiendo una estructura jerárquica en la que existen unos niveles. El esquema de dicha estructura es el siguiente:



El objeto principal de la jerarquía es el objeto *window* que representa una ventana del navegador. Se considera un objeto implícito y no es necesario nombrarlo para acceder a los objetos que se encuentran en un nivel inferior. Por ejemplo, si queremos acceder a la propiedad que cambia el color de fondo de una página podemos hacerlo de este modo: *window.document.bgcolor* pero también podemos hacerlo escribiendo *document.bgcolor*.

El objeto *navigator* identifica las características de la plataforma sobre la cual esta ejecutándose la aplicación y el objeto *screen* corresponde a la pantalla que está utilizando el usuario.

El objeto *document* se refiere a los documentos que se cargan en la ventana del navegador. Posiblemente sea uno de los objetos más utilizados en la programación de este lenguaje, ya que de él dependen a su vez otros objetos muy importantes como son los vínculos, puntos de anclaje, los formularios o las imágenes.

El objeto *history* almacena las referencias de todos los sitios web visitados y el objeto *location* corresponde a la URL de la página en uso.

Algunas propiedades y métodos del objeto location, no incluido en el manual, son los siguientes:

Propiedades

- hash: Muestra la cadena que representa el anclaje de la URL.
- host: Muestra la cadena que representa el nombre del dominio del servidor y el número del puerto dentro de la URL.
- hostname: Muestra la cadena que representa el nombre del dominio del servidor dentro de la URL.
- href: Muestra la URL completa.
- pathname: Muestra la cadena que sigue al nombre del servidor.
- port: Muestra el número del puerto de la URL.
- protocol: Muestra el protocolo utilizado por la página
- search: Muestra la cadena de búsqueda que se muestra después del signo de interrogación en la URL

Métodos

- assign(): Carga un nuevo documento.
- reload(): Carga de nuevo el documento actual.
- replace(): Cambia la URL del documento actual por otra URL

Veamos un ejemplo con algunos de los objetos que hemos visto:

```
<head>
```

```
<title>ejemplo de javascript </title>
```

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

```
Document.write("Tu Navegador es: "+navigator.appName + "<br>");
```

```
if (navigator.javaEnabled()){
```

```
    document.write("Tu navegador soporta los Applets de Java"+"<br>");
```

```
}
```

```
else {
```

```
    document.write("Tu navegador no soporta los Applets de Java"+"<br>");
```

```
}
```

```
document.write("La altura de tu pantalla es: "+screen.height+"<br>");
```

```
document.write("La anchura de tu pantalla es: "+screen.width+"<br>");
```

```
document.write("La URL completa es: "+location.href+"<br>");
```

```
</script>
```

```
</head>
```

```
<body>
```

```

<noscript>

  <p>La página que estás viendo requiere para su funcionamiento el uso de JavaScript

</noscript>

<form>

  <input type="button" value="WIKIPEDIA" onclick="location.assign('http://Wikipedia.es');">

</form>

</body>

</html>

```

El resultado de la ejecución de este ejemplo sería el siguiente:

Tu Navegador es: Netscape
 Tu navegador no soporta los Applets de Java
 La altura de tu pantalla es: 600
 La anchura de tu pantalla es: 800
 La URL completa es: file:///C:/Documents%20and
 %20Settings/Mati/Escritorio/JAVASCRIPT/EJERCICIOS/pruebaobjetosdelnavegador.html

WIKIPEDIA

(Al hacer click sobre el botón se carga la página de la wikipedia).

Para terminar comentar que el método document.write no se limita sólo a escribir un texto tal como hemos visto hasta ahora. Podemos utilizarlo para crear y manipular todo tipo de objetos. Esto es una práctica habitual en las páginas web que utilizan ventanas emergentes.

Ejemplo:

```

<script type="text/javascript">
  var nuevaVentana;
  function crearnueva() {
    nuevaVentana=window.open("", "pagina2",
    "toolbar=yes,location=no,menubar=yes"+
    "width=300,height=300" );
    nuevaVentana.document.write("<HTML><HEAD><TITLE>" +
    "Sin Título</TITLE></HEAD>\n");
    nuevaVentana.document.write("<BODY><form>\n");
    nuevaVentana.document.write("<input type='button' "+
    "value='Cerrar' onClick='window.close();>\n");
    nuevaVentana.document.write("</form>\n");
    nuevaVentana.document.write("</BODY></HTML>\n");
  }
</script>

```

Los ejercicios que debéis entregar son los siguientes:

1. Crear una página web que nos muestre el mensaje “Bienvenido a mi página” con formato H1, nos muestre la plataforma del navegador que estamos utilizando, y con la ventana de Confirm nos pida si queremos continuar. En caso de que así sea saldrá en pantalla el mensaje “continuamos...”

2. Crear una página con el botón “Fecha Actual”. Al pinchar sobre el botón debe salir la fecha actual del sistema pero con el día y el mes en letra, utiliza los métodos para fecha y hora, punto 16 de los apuntes extras. Para activar el botón podéis utilizar el evento onclick, aparece en el apartado 6.2 libro y también en el punto 6 de los apuntes extra. Ejemplo:

Lunes, 15 de Octubre de 2012

3. Crear una página con el botón “Hora Actual”. Al pinchar sobre el botón debe salir un mensaje del tipo:

Buenos días, son las 12:30 horas

El saludo debe variar según la hora del día. Buenos días hasta las 14:00 horas, Buenas tardes hasta las 20:00 y a partir de esta hora debe salir Buenas noches.

4. Crea una página con tres botones con el texto Azul, Verde y Rojo. Al pinchar sobre cada botón debe cambiar el fondo de la página al color indicado. Utilizaremos el evento onclick, en temas posteriores veremos otros eventos.

5. Mostrar en pantalla un documento que nos muestre:

- La URL del documento actual.
- El pathname de la página (corresponde a la cadena que sigue al nombre del servidor).
- El protocolo utilizado por la página web.

Debéis insertar también un botón para que al pinchar sobre él se cargue la página de Google.

6. Crear una página que contendrá el botón Abrir. Al pinchar sobre el botón debe abrirse una pestaña nueva en el navegador y esta contendrá, a su vez, el botón Cerrar. Al pinchar sobre él debemos volver a la página inicial sin cerrar la pestaña actual. Puedes utilizar el objeto window