

Ejercicios 4: JavaScript

En esta esta tanda de ejercicios vamos a seguir trabajando con *JavaScript* resolviendo 6 ejercicios prácticos. En todos los ejercicios se proporciona un fichero .html y un fichero .js, dentro de la carpeta js, que el alumno tiene que modificar para cumplir los requisitos solicitados en cada enunciado.

Una vez finalizados todos los ejercicios, el alumno debe comprimir todos los ficheros en un único archivo .zip o .rar y subirlo a la plataforma virtual del curso. El nombre del fichero tiene que ser de la forma:

- 04_apellido1_apellido2_nombre.zip
- 04_apellido1_apellido2_nombre.rar

Ejercicio 4-1: Manejo de arrays y objeto Date

Abriendo el fichero Ejercicio_4-1.html que acompaña este enunciado en la pantalla del navegador se muestra:

Ejercicio 4-1

En este ejercicio vamos a generar código JavaScript que escriba la fecha actual:

Hoy es lunes 27 de abril de 2020

Dentro de este fichero a su vez se encuentra la línea:

```
<SCRIPT TYPE="text/javascript" SRC="js/Ejercicio_4-1.js"> </SCRIPT>
```

que como se sabe incorpora el código JavaScript existente en el fichero Ejercicio_4-1.js que también se proporciona en el enunciado. En este fichero .js se ha definido la función `escribe()` que al llamarla con una cadena de texto con código HTML, lo sustituye por la última línea de la imagen anterior. Por ejemplo, si hacemos la siguiente llamada `escribe("Hola mundo")` en la pantalla del navegador se mostraría:

En este ejercicio vamos a generar código JavaScript que escriba la fecha actual:

Hola mundo

En este ejercicio hay que crear la función `ponfecha()`. Esta función tiene que generar una cadena de texto con el código HTML con la fecha actual, en el mismo formato al mostrado en la primera imagen, y luego llamar a la función `escribe()` para que lo muestre en pantalla.

Dentro del fichero Ejercicio_4-1.html se ha escrito el código:

```
<body onload="ponfecha()">
```

que lo que hace es llamar automáticamente a la función `ponfecha()`, que se ha creado en este ejercicio, cuando la página esté totalmente cargada.

Ayuda:

En este ejercicio se va a manejar el objeto **Date** para obtener la fecha actual y para luego ir extrayendo los distintos elementos de la fecha y la hora.

http://www.w3schools.com/jsref/jsref_obj_date.asp

Por ejemplo:

- `var d = new Date();` Asigna a la variable `d` la información con la fecha y hora actuales.
- `var dia_semana = d.getDay();` Asigna a la variable `dia_semana` el número con el día de la semana de la fecha almacenada en la variable `d`.

Tanto `getDay()` como `getMonth()` devuelven un número con el día de la semana (0 domingo, 1 lunes, 2 martes, etc.) y el mes (0 enero, 1 febrero, 2 marzo, etc.) respectivamente, pero para escribir la fecha necesitamos los nombres correspondientes. Para resolver este problema se debe crear un array con el nombre de los días de la semana y otro con el nombre de los meses del año. Por ejemplo:

```
var semana = ["domingo", "lunes", "martes", "miércoles",
    "jueves", "viernes", "sábado"];
```

Para obtener el nombre del día de la semana de la fecha actual habría que escribir el siguiente código:

```
var dia_semana = d.getDay();
var texto_dia_semana = semana[dia_semana];
```

Para el mes habría que hacer algo similar.

http://www.w3schools.com/js/js_obj_array.asp

Ejercicio 4-2: Reloj

Siguiendo el mismo esquema del ejercicio anterior, en éste hay que crear la función `ponhora()`. Esta función tiene que generar una cadena de texto con el código HTML con la hora actual y acabar llamando a la función `escribe()` para que escriba esta cadena en la página, tal y como se muestra en la imagen.

Ejercicio 4-2

En este ejercicio vamos a generar código JavaScript que escriba la hora actual:

19:11:41

Para añadir un poco de dificultad al ejercicio, a la hora de representar tanto la hora como los minutos y segundos, siempre hay que usar dos dígitos numéricos por lo que habrá que añadir un cero a la izquierda cuando tengan un valor inferior a 10.

Ayuda:

En este ejercicio se va a manejar el objeto Date para obtener la hora actual, para luego ir extrayendo los distintos elementos que componen la hora y, finalmente, generar la cadena de texto con el código HTML que la escriba.

Para que la hora se vaya actualizando automáticamente, es necesario llamar periodicamente a la función `ponhora()`. Esto se consigue con `setTimeout()`.

http://www.w3schools.com/jsref/met_win_settimeout.asp

La última linea de la función `ponhora()` debe ser, por ejemplo:

```
t=setTimeout(function(){ponhora()},500);
```

Que lo que hace es volver a ejecutar la función `ponhora()` al cabo de 500 milisegundos (0,5 segundos).

DOM (Document Object Model)

En los siguientes de 4 ejercicios se va a trabajar con el DOM (*Document Object Model*) que va a permitir al programador de JavaScript acceder a cualquiera de los elementos que componen una página web, tanto para consultar/modificar sus atributos y propiedades, como para aplicarle un método o una función. En la anterior tanda de ejercicios ya se comenzó a utilizar y ahora vamos a seguir profundizando.

El primer paso para interactuar desde JavaScript con un objeto es conseguir llegar al objeto, es decir, seleccionarlo y distinguirlo del resto de objetos que componen la página. Existen varias formas de hacerlo, pero actualmente la más habitual consiste en usar alguno de los tres métodos siguientes que el objeto `document` proporciona para ello:

- `getElementsByName()`
- `getElementsByTagName()`
- `getElementById()`

En la sección “5.3. Acceso directo a los nodos”, página 43 y siguientes, del documento PDF entregado, se puede encontrar una explicación de cada uno de estos métodos:

Una vez que se ha accedido a un objeto de la página, el siguiente paso consiste en acceder y/o modificar sus atributos y propiedades. Mediante DOM, es posible acceder de forma sencilla a todos los atributos XHTML y todas las propiedades CSS (hojas de estilo) de cualquier elemento de la página.

En la sección “5.5. Acceso directo a los atributos XHTML”, página 46 y siguientes, del documento PDF entregado, se puede encontrar una explicación de cómo acceder y/o modificar los atributos y propiedades de un objeto de la página web.

En esta colección de ejercicios vamos a practicar a usar JavaScript para modificar objetos de una página web. Los dos primeros son ejercicios propuestos y resueltos en el documento PDF entregado, pero por su interés, es fundamental intentar hacerlos y, sobre todo, acabar entendiendo. El resto de ejercicios de esta tanda el

alumno tiene que resolverlos por sus propios medios aunque planteando cualquier duda o sugerencia en los foros del curso.

Ejercicio 4-3:

En este ejercicio se proporciona el fichero Ejercicio_4_3.html con una página web con tres párrafos de texto falso en el que se han añadido algunos enlaces con la etiqueta <a> de HTML.

A partir de la página web proporcionada y utilizando las funciones DOM, mostrar por pantalla la siguiente información:

- Número de enlaces de la página
- Dirección a la que enlaza el **tercer** enlace
- Numero de enlaces que **no** enlazan a <http://prueba>
- Número de enlaces del **segundo** párrafo

Esta información hay que escribirla dentro de un bloque <div> creado justo antes de los tres párrafos de texto falso. A este bloque se le ha definido el atributo id="informacion" por lo que para acceder a él se podría usar:

```
document.getElementById("informacion")
```

Una de las propiedades del bloque <div> es innerHTML que es el contenido HTML del bloque. Por tanto, para añadir código HTML a ese bloque simplemente habría que modificar esta propiedad ejecutando las siguientes instrucciones:

```
var info = document.getElementById("informacion");
info.innerHTML = "<h3>Información pedida en el enunciado</h3>";
```

Si después queremos **añadir** más información, habrá que poner:

```
info.innerHTML = info.innerHTML + Número de enlaces: " + num;
```

Es decir, el nuevo contenido HTML del bloque es, el anterior, más el nuevo contenido.

Ayuda:

- Este ejercicio, con alguna pequeña modificación (ver negritas), se corresponde con el Ejercicio 11 del documento PDF entregado:
- window.onload = function() { } garantiza que el código no se ejecuta hasta que no se haya cargado totalmente la página. Muy importante y error muy común y difícil de detectar. Es una alternativa a <body onload="...."> empleado en ejercicios anteriores.

Ejercicio 4-4:

En este ejercicio se proporciona el fichero Ejercicio_4-4.html con una página web con un párrafo de texto falso. Hay que observar que parte del párrafo está agrupado dentro de una etiqueta a la que se le ha asignado el atributo id="adicional" y el atributo class="oculto". Este último hace referencia a

una clase definida al comienzo del fichero. A su vez, al enlace se le ha asignado el atributo `id="enlace"`.

Para resolver el ejercicio hay que completar el código JavaScript de la función `muestra()` proporcionada para que cuando se pinche sobre el enlace se muestre completo el contenido de texto. Además, el enlace debe dejar de mostrarse después de pulsarlo por primera vez. La acción de pinchar sobre un enlace forma parte de los "Eventos" de JavaScript que se ven en el capítulo 6 del documento entregado. En este ejercicio, sólo se debe saber que al pinchar sobre el enlace, se ejecuta la función `muestra()`.

Ayuda:

- La propiedad que almacena la clase de un elemento es `className`
- Este ejercicio se corresponde con el Ejercicio 12 del documento PDF entregado:

Ejercicio 4-5:

Al abrir el fichero `Ejercicio_4-5.html` con un navegador web se muestra el siguiente contenido:

The screenshot shows a simple web form with a title 'Cálculo de la letra del DNI'. Below the title is a label 'DNI:' followed by a text input field. At the bottom is a button labeled 'Calcula la letra'.

Modificar el fichero para que al pulsar sobre el botón “Calcula la letra”, lea el valor introducido en el formulario, calcule la letra del DNI y escriba de nuevo en el formulario el DNI completo, con la letra al final.

Ayuda:

- Al pulsar el botón se llama a la función `obtenerletra()` que habrá que completar.
- Para el cálculo de la letra, seguir las indicaciones dadas en el ejercicio 1-2, ya resuelto en la primera tanda de ejercicios. Para leer o modificar el valor introducido en el formulario hay que utilizar la propiedad `value`, como ya se ha hecho anteriormente. Quizás haya que añadir algún atributo al formulario para poder acceder al cuadro de texto donde el usuario teclea el DNI empleando el método `getElementById()`.

Ejercicio 4-6:

Al abrir el fichero Ejercicio_4-6.html con un navegador web se muestra el siguiente contenido:



En la carpeta `img` se han incluido 10 ficheros del mismo tamaño, 770x130 pixels, con las imágenes de varias facultades y escuelas de la UNED.

En este ejercicio hay que hacer las modificaciones necesarias en el fichero `Ejercicio_4-6.html` y en el fichero `Ejercicio_4-6.js`, situado dentro de la carpeta `js`, para que cada 2 segundos:

- Cambie aleatoriamente la imagen por cualquier otra de los 10 ficheros proporcionados.
- Cambie el texto para que haga referencia a la imagen que se muestra.

Ayuda:

- Para cambiar la imagen que muestra el objeto `img`, hay que modificar la propiedad `src`. Para cambiar el texto del párrafo `<p>` se puede usar la propiedad `text` o la propiedad `innerHTML`.
- Para ejecutar periódicamente una función, se puede usar el método `setTimeout`. Por ejemplo:

```
tiempo=setTimeout("cambiarFoto()", 2000);
```

llama a la función `cambiarFoto()` cada 2000 milisegundos (2 segundos)

- Antes de empezar a ejecutar el código JavaScript hay que tener garantizado que la página está cargada completamente. Para ello utilizar el evento `onload` como ya se mencionó en un ejercicio anterior.

Comentario final:

Cualquier dificultad, cuestión, sugerencia, etc. sobre el desarrollo de estos ejercicios, o sobre el contenido del curso, puede y debe plantearla en el foro del curso.