

Ejercicios 3: JavaScript

En esta esta tanda de ejercicios vamos a seguir a trabajando con *JavaScript* resolviendo tres ejercicios prácticos. En todos los ejercicios se proporciona un fichero `.html` que el alumno puede modificar para cumplir los requisitos solicitados en el 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:

- **03**_apellido1_apellido2_nombre.zip
- **03**_apellido1_apellido2_nombre.rar

Requisito general para todos los ejercicios propuestos:

En los ejercicios realizados hasta ahora, el código JavaScript que hemos creado lo hemos introducido directamente dentro del fichero HTML entre las etiquetas `<SCRIPT>` y `</SCRIPT>`, definiendo en la primera de ellas además el atributo `TYPE`.

Es práctica habitual en los desarrolladores de JavaScript, cuando el código es muy extenso y/o se quiere utilizar en varias páginas o simplemente para estructurar mejor el trabajo, crear un fichero externo con extensión `.js` con el código JavaScript desarrollado (ver sección 1.4.2 de la documentación entregada y la página https://www.w3schools.com/js/js_where_to.asp).

En esta tanda de ejercicios se van a definir funciones en JavaScript que después podrían ser utilizadas por más de una página Web, por lo que es el momento de introducir esta forma de trabajar. Por tanto, en todos los ejercicios propuestos el código JavaScript creado hay que introducirlo en un fichero externo con extensión `.js` y hay que hacer la correspondiente llamada en el fichero HTML.

Por ejemplo, en el Ejercicio 3-1 se va a escribir en el fichero `Ejercicio_3-1.js` el código JavaScript necesario para resolverlo, y dentro del fichero `Ejercicio_3-1.html` hay que añadir la línea siguiente para referenciarlo:

```
<SCRIPT TYPE="text/javascript" SRC="js/Ejercicio_3-1.js"> </SCRIPT>
```

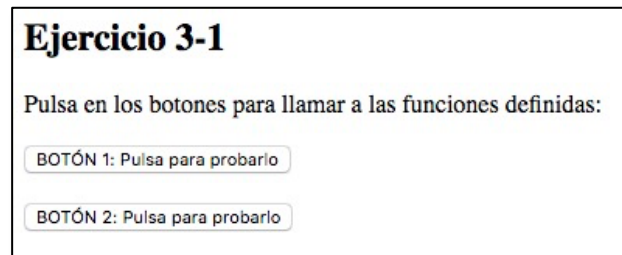
Dentro del fichero `.js` hay que escribir directamente código JavaScript, sin las etiquetas `<SCRIPT>` y `</SCRIPT>`.

Página a consultar sobre funciones:

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

Ejercicio 3-1: Funciones sin parámetros

Abriendo el fichero `Ejercicio_3-1.html` que acompaña este enunciado en la pantalla del navegador se muestra:



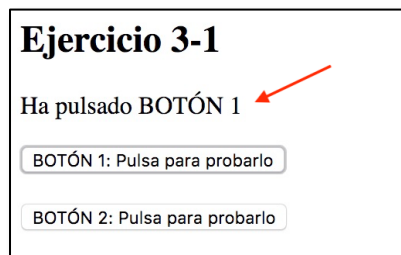
Y dentro se han añadido las líneas de código siguiente para definir los dos botones que aparecen en la página:

```
<button onclick="mensaje1()">BOTÓN 1: Pulsa para probarlo</button>
```

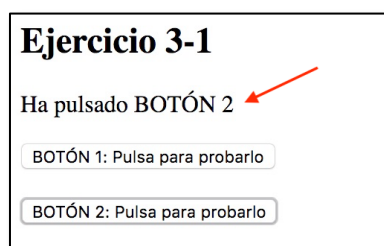
```
<button onclick="mensaje2()">BOTÓN 2: Pulsa para probarlo</button>
```

que al hacer click en ellos llaman, respectivamente, a las funciones JavaScript `mensaje1` y `mensaje2` (ver <https://www.w3schools.com/jsref/event onclick.asp>)

En este ejercicio tenemos que crear las dos funciones mencionadas: `mensaje1()` y `mensaje2()` que simplemente cambian el texto mostrado en la página. Por ejemplo, cuando se pulse sobre BOTÓN 1:



y cuando se pulse sobre BOTÓN 2:



Como se ha mencionado anteriormente, el código JavaScript de la definición de las funciones `mensaje1()` y `mensaje2()` tiene que estar en el fichero `Ejercicio_3-1.js`.

Ayuda:

La función `getElementById()` es muy utilizada cuando se desarrollan aplicaciones web dinámicas para acceder directamente a elementos de una página y así poder leer o modificar sus propiedades, por ejemplo: la imagen mostrada, el

color del texto, tamaño del texto, hacer visible un bloque, contenido HTML, URL de un enlace, etc, etc.....

En este ejercicio tenemos que modificar el contenido HTML que se encuentra dentro de un bloque `<DIV>` al que se le ha asignado el ID "texto":

```
<div id="texto">
<p>Pulsa en los botones para llamar a las funciones definidas:</p>
</div>
```

Por tanto el contenido HTML de este bloque es:

```
<p>Pulsa en los botones para llamar a las funciones definidas:</p>
```

Y queremos que cuando se pulse el primer botón pase a ser:

```
<p>Ha pulsado BOTÓN 1</p>
```

y cuando se pulse sobre el segundo botón sea:

```
<p>Ha pulsado BOTÓN 2</p>
```

Todo el contenido HTML de ese bloque DIV está almacenado en la propiedad `innerHTML` (tener especial cuidado con las minúsculas y mayúsculas). Si queremos que contenga otro contenido HTML, como los mencionados antes, hay que modificar el valor de esa propiedad. Por ejemplo, la función `mensaje1()` debería tener la línea de código:

```
document.getElementById("texto").innerHTML = "<p>Ha pulsado BOTÓN 1</p>";
```

y la función `mensaje2()` el código:

```
document.getElementById("texto").innerHTML = "<p>Ha pulsado BOTÓN 2</p>";
```

Con esta ayuda el ejercicio está prácticamente hecho.

Ejercicio 3-2: Funciones con un parámetro de entrada

En este ejercicio vamos a crear funciones a las que llamaremos con un parámetro de entrada y nos devolverán otro valor de salida.

Abriendo el fichero `Ejercicio_3-2.html` que acompaña este enunciado en la pantalla del navegador se muestra:

Cálculo del factorial:

Número:

El factorial es:

Conversión de grados centígrados a grados Fahrenheit:

Introduce grados centígrados:

Conversión a grados Fahrenheit:

Cálculo del factorial:

En el fichero están definidos los cuadros de texto de esta parte como:

Número: `<input type="text" id="num1" size="20">
`

El factorial es: `<input type="text" id="resultado" size="20" readonly>
`

Como la cadena introducida en un cuadro de texto está almacenada en la propiedad `value` de ese elemento, para acceder a ella hay que usar la expresión:


```
document.getElementById("id_del_elemento").value
```

Por tanto, en nuestro caso si queremos conocer el dato introducido en el cuadro de texto con ID `num1` tenemos que usar:


```
document.getElementById("num1").value
```

Y análogamente para cambiar el valor que se muestra en el cuadro de texto con ID `resultado` tenemos que usar:

```
document.getElementById("resultado").value
```

Dentro del formulario vemos que el botón  está definido como:

```
<input type="Button" value="Calcula"
onclick='document.getElementById("resultado").value =
factorial(document.getElementById("num1").value)' >
```

Donde vemos que hacer click en el botón  se llama a la función `factorial()` definida en código JavaScript. Esta función tiene un parámetro de entrada (en nuestro caso el valor introducido en el formulario) y devuelve un valor (que se le asigna al cuadro de texto que muestra el resultado del factorial)


Por tanto, para resolver esta parte del ejercicio se va a crear la función `factorial()` que calcula el factorial de un número. Para calcular el factorial de un número, hay que multiplicar este número por todos los anteriores a él hasta el 1. Por ejemplo:

- El `factorial(4)` se calcula con el producto $4 \cdot 3 \cdot 2 \cdot 1 = 24$
- Además `factorial(0) = 1`


Como se ha mencionado anteriormente todo el código JavaScript, incluyendo la definición de la función `factorial()`, tiene que estar en el fichero `Ejercicio_3-2.js`.

Conversión de grados centígrados a grados Fahrenheit:

Esta parte del ejercicio es similar a la anterior y no vamos a entrar en tantos de detalles.

En este caso el botón  está definido como:

```
<input type="Button" value="Pulsa para calcular grados Fahrenheit"
onclick='document.getElementById("idFahrenheit").value =
convertir(document.getElementById("idCentigrados").value)' >
```

Donde vemos que hacer click en el botón  se llama a la función `convertir()` definida en código JavaScript. Esta función tiene un parámetro de entrada que en nuestro caso es el valor introducido en el formulario con ID `idCentigrados`. Y esta función devuelve un valor que, en nuestro caso, se le asigna al cuadro de texto con ID `idFahrenheit`

Por tanto, para resolver esta parte del ejercicio se va a crear la función `convertir()` que calcula el factorial de un número. La fórmula de conversión es muy sencilla y el alumno la tiene que buscar.

Ayuda:

Además de los múltiples comentarios hechos en el enunciado, conviene mencionar uno más, origen de múltiples errores.

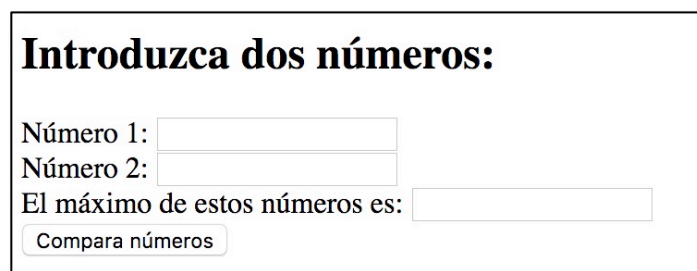
La propiedad `value` empleada para saber qué dato ha introducido el usuario en el formulario devuelve una cadena de texto, por lo que para evitar posibles errores conviene convertirla a un valor numérico si efectivamente se va a utilizar como un número, como es el caso en este ejercicio.

Para hacer esta conversión de cadena de texto a número, se pueden usar las funciones:

- `parseInt()` (ver https://www.w3schools.com/jsref/jsref_parseint.asp)
- `parseFloat()` (ver https://www.w3schools.com/jsref/jsref_parsefloat.asp)

Ejercicio 3-3: Función con dos parámetros de entrada y un valor de retorno

Abriendo el fichero `Ejercicio_3-3.html` que acompaña este enunciado en la pantalla del navegador se muestra:



Dentro del código HTML utilizado para definir el formulario, el botón se ha creado con el siguiente código:

```
<input type="Button" value="Validar datos"
onclick='document.getElementById("resultado").value =
calculoMax(document.getElementById("num1").value,
document.getElementById("num2").value) ' >
```

Como ya hemos mencionado en el anterior enunciado:

- `document.getElementById("num1").value` hace referencia al valor del primer campo del formulario con ID `num1`.
- `document.getElementById("num2").value` hace referencia al valor del primer campo del formulario con ID `num2`.
- `document.getElementById("resultado").value` hace referencia al valor del tercer campo del formulario con ID `resultado`.

Por tanto este código llama a la función `calculoMax()` con los valores introducidos en el formulario (ID="num1" e ID="num2"), y el valor que devuelve la función se escribe en la tercera casilla (ID="resultado") de ese mismo formulario.

En este ejercicio hay que desarrollar la función `calculoMax()` con dos parámetros de entrada y con un valor de retorno. El valor de retorno de la función `calculoMax()` es el máximo de los dos parámetros de entrada. Por ejemplo:

- `calculoMax(34, 7)` devuelve el valor 34
- `calculoMax(23, 45)` devuelve el valor 45

Como se ha mencionado anteriormente, el código JavaScript de la definición de la función `calculoMax()` tiene que estar en el fichero `Ejercicio_3-3.js`.

Ayuda:

Simplemente recordar lo mencionado en el ejercicio anterior sobre convertir las cadenas de texto en números. Si no lo hacemos nos vamos encontrar con que “92” es mayor que “725”, dado que comparando cadenas de texto así sería.

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.