

# Ejercicios iniciales JavaScript

## Notas

Para poder usar JavaScript en nuestras páginas HTML, usaremos una etiqueta script en el head con el atributo defer para que se cargue el código JavaScript de manera asíncrona mientras se sigue procesando el documento. Además defer garantiza que el código JavaScript solo se comienza a ejecutar cuando se haya cargado el documento.

```
<script src="nombreFichero.js" defer></script>
```

Luego en el fichero JavaScript debemos enlazar los eventos necesarios a los elementos deseados. Para ello, por ahora usaremos el id de los elementos con **document.getElementById**:

```
const nombre = document.getElementById("idDeseado");
```

Luego usaremos el método **addEventListener** del elemento para indicar qué evento "escucharemos" y qué función ejecutar cuando se produzca:

```
nombre.addEventListener("evento", nombreFunción);
```

Siendo evento **click**, **change**, **mouseover**, **keypress**, ....

Para crear una función, por ahora, usaremos el siguiente Código:

```
function nombreDeseado() {  
  }  
}
```

Si no necesitamos más la variable o constante, podemos hacer lo anterior en una sola línea:

```
document.getElementById("idDeseado").nombre.addEventListener("evento", nombreFunción);
```

## Comprobar si un cuadro de texto contiene números

Si necesitamos comprobar que el valor introducido en un cuadro de texto es un número, tenemos varias opciones.

### Números enteros

Es decir, números sin decimales.

Tenemos la función **parseInt** que convierte un texto en número o devuelve NaN (not a number) si no puede hacerlo. Esta función convierte mientras encuentre espacios o números, parando en cuanto no los haya, pero devolviendo todo lo que puede.

Ejemplos:

parseInt("3")	devuelve 3
parseInt(" 3")	devuelve 3
parseInt(" 32a")	devuelve 32
parseInt("a3")	devuelve NaN
parseInt("")	devuelve NaN

Con lo cual puede que no nos interese usarla, al menos por si sola.

Para comprobar si un valor es NaN usaremos la función **isNaN**.

# Ejercicios iniciales JavaScript

Ejemplo:

```
if(isNaN(parseInt("3a"))) {  
    console.log("No es un número");  
}
```

## Números reales

Es decir, números con decimales.

Tenemos la función **parseFloat** que convierte un texto en número o devuelve NaN (not a number) si no puede hacerlo. Esta función convierte mientras encuentre espacios o números, parando en cuanto no los haya, pero devolviendo todo lo que puede.

Ejemplos:

parseFloat("3")	devuelve 3
parseFloat("3.2")	devuelve 3.2
parseFloat("3,2")	devuelve NaN
parseFloat("3.2e10")	devuelve el 32000000000 pues E es notación exponencial
parseFloat(" 3")	devuelve 3
parseFloat(" 32a")	devuelve 32
parseFloat("a3")	devuelve NaN
parseFloat("")	devuelve NaN

Con lo cual puede que no nos interese usarla, al menos por si sola.

## Números enteros y reales

Podemos usar **Number**(expresión), que admite también espacios pero si encuentra un valor no válido devuelve NaN pero con el inconveniente de que devuelve 0 con una cadena vacía.

Ejemplo:

Number("3")	devuelve 3
Number("3.2")	devuelve 3.2
Number("3,2")	devuelve NaN
Number("3.2e10")	devuelve el 32000000000 pues E es notación exponencial
Number(" 3")	devuelve 3
Number(" 32a")	devuelve NaN
Number("a3")	devuelve NaN
Number("")	devuelve 0
Number(" ")	devuelve 0

## Ejemplo2

Se debe calcular el total al pulsar el botón.

El formulario, titulado "Cálculos", contiene los siguientes elementos:

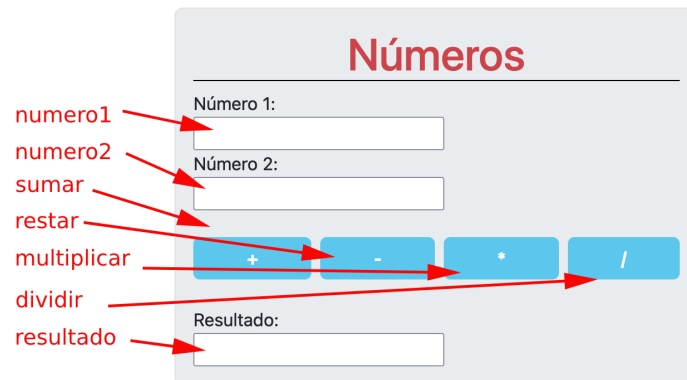
- Un campo de entrada etiquetado "Importe:" con una flecha roja que apunta a él desde la etiqueta "importe".
- Un campo de entrada etiquetado "Cantidad:" con una flecha roja que apunta a él desde la etiqueta "cantidad".
- Un botón verde con el texto "Calcular total" con una flecha roja que apunta a él desde la etiqueta "calcular".
- Un campo de entrada etiquetado "Total:" que muestra el valor "0", con una flecha roja que apunta a él desde la etiqueta "total".

## Ejemplo3

En este ejercicio realizamos las operaciones sumar, restar, multiplicar y dividir con el número 1 y el número 2.

# Ejercicios iniciales JavaScript

El operador + sirve tanto para sumar números como para concatenar cadenas. Y como la propiedad value de un input text o number devuelve siempre un string (texto) aunque se introduzcan números, debemos convertir esos value en números con la función parseInt.



## Ejemplo 4

Debemos convertir de grados Celsius a Farenheit y viceversa según el botón pulsado. Con el botón reiniciar, borraremos el contenido de los cuadros de texto. Para ello le asignaremos la cadena vacía "" a la propiedad value de los mismos.

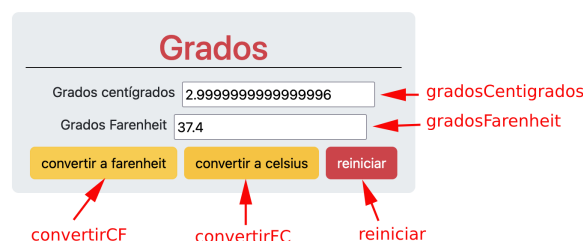
- para convertir de Celsius a Farenheit la fórmula es:

```
gradosCentígrados * (9 / 5) + 32
```

- para convertir de Farenheit a Celsius la fórmula es:

```
(gradosFarenheit - 32) * (5 / 9);
```

- en ambos casos usamos paréntesis para que se haga la operación dentro de los mismos antes que las otras. Por defecto la multiplicación y división tienen más prioridad que la suma y la resta. Si hay una multiplicación y una división, la primera que aparezca por la izquierda; lo mismo para la suma y la resta.
- También podemos poner directamente 1,8 en lugar de 9 / 5 y 0,55 en lugar de 5 / 9.



## Ejemplo5

El usuario introducirá una nota y le diremos si está aprobado (un 5 o más) o suspenso. Este mensaje se mostrará en un párrafo que al principio aparece oculto. Además, si está aprobado, mostraremos ese texto y el texto de la nota en verde, sino en rojo.

También comprobaremos si la nota se deja vacía, comparando su valor con la cadena vacía "". Si está vacía, pondremos en el párrafo del texto un mensaje indicando que introduzca la nota.

# Ejercicios iniciales JavaScript

Al pulsar el botón comprobaremos que la nota esté entre 0 y 10, mostrando un mensaje en el párrafo si no es así y volveremos a darle el foco a la nota, con su método **focus()** y seleccionaremos el texto de la nota, con su método **select()**, para que se pueda escribir otra sin tener que borrarla primero.

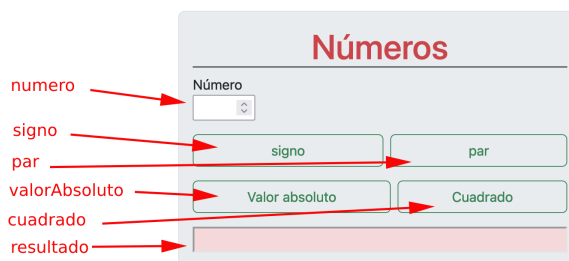


## Ejemplo6

- el botón signo mostrará si el número es positivo, negativo o cero en resultado
- el botón par mostrará si el número es par o impar en resultado
- el botón valor absoluto mostrará en resultado el valor absoluto del número, es decir si es positivo 0 cero, lo mostrará tal cual, si es negativo lo mostrará sin signo
- el botón cuadrado mostrará el cuadrado del número (el número multiplicado por si mismo) en resultado.
- También comprobaremos por primera vez que en número esté introducido un número, no letras o que haya quedado vacío. En algunos navegadores, al usar un elemento number ya no deja escribir letras pero en otros sí; además ninguno evita que podamos pegar texto aunque no sean números. Para ello usaremos el siguiente código:

```
if (Number.isInteger(parseInt(valor)) == false)
```

Cambiando valor por el valor deseado. También false por true según nos interese (false para ver si no es un número, true para ver si lo es).



## Ejemplo7

En este ejercicio calcularemos el área de un rectángulo o de un triángulo.

- la del triángulo:  $\text{base} * \text{altura} / 2$
- la del rectángulo:  $\text{base} * \text{altura}$

También comprobaremos que la base y la altura sean números enteros válidos, usando **Number.isInteger** y **parseInt** de una forma similar a esta:

```
if(Number.isInteger(parseInt(valor)) == true) {  
  }  
}
```

Cambiando valor por el deseado y true por false si queremos comprobar lo contrario.

# Ejercicios iniciales JavaScript

De no serlo mostraremos un mensaje en el cuadro área indicándolo.

Diagrama de un formulario web titulado "Areas". El formulario contiene los siguientes elementos:

- Base:
- Altura:
- Triángulo:
- Rectángulo:
- Área:

Las etiquetas de los elementos son:

- base
- altura
- triangulo
- rectangulo
- area

## Ejemplo8

Ejercicio similar al anterior pero usando unos controles radio para saber qué área calcular. Para comprobar si un radio está marcado usaremos su propiedad **checked** la cual tendrá el valor true si lo está.

Diagrama de un formulario web titulado "Areas". El formulario contiene los siguientes elementos:

- Base:
- Altura:
- Triángulo: ☐
- Rectángulo: ☒
- Calcular:
- Área:

Las etiquetas de los elementos son:

- base
- altura
- radioRectangulo
- radioTriangulo
- btnCalcular
- area

## Ejemplo9

En este ejercicio el texto de la parte inferior debe ir cambiando según el usuario seleccione los diferentes elementos.

- añadiremos un listener al range del tipo **input** en lugar de click, para que se vaya produciendo mientras el usuario mueva el deslizador
- añadiremos un listener a color, fuente, negrita y cursiva de tipo **change** en lugar de click para que se produzca cuando el usuario elige un tipo de letra o marca o desmarca los checkbox
- usaremos `.style.fontWeight`, `.style.fontStyle`, `.style.color` y `.style.fontFamily` para cambiar las propiedades adecuadas del elemento texto.
- usaremos la propiedad **value** del elemento para obtener el color o valor seleccionado de la lista
- para comprobar si un checkbox está marcado usaremos su propiedad **checked** que contendrá el valor true si lo está.
- para asignar el tamaño debemos de tener en cuenta que el valor del range solo tiene el número, mientras que en CSS debe incluir px, em, rem, ... Para ello pondremos algo como esto:

```
elemento.style.fontSize = valor + "rem";
```

Cambiando elemento, valor y rem por los deseados.

# Ejercicios iniciales JavaScript



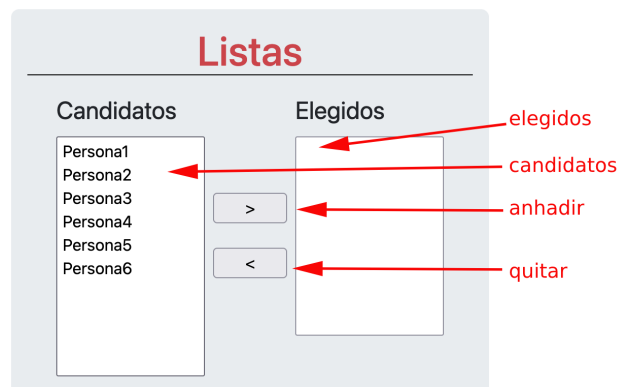
## Ejemplo10

Con el botón > moveremos el elemento seleccionado de la lista de candidatos a la lista de elegidos; con el botón < lo contrario.

- Para saber si hay un elemento seleccionado en una lista usaremos su propiedad **selectedIndex** la cual debe tener un valor distinto a -1
- Para mover el elemento seleccionado de una lista a otra usaremos algo como esto:

```
lista2.appendChild(lista1.options[lista1.selectedIndex]);  
lista1.remove(lista1.selectedIndex);
```

Cambiando lista1 y lista2 por los nombres deseados.



## Ejemplo11

El usuario debe introducir el importe, la cantidad y elegir el tipo de IVA. Una vez pulsado el botón Calcular, mostraremos el IVA (importe \* cantidad \* iva / 100), el total (importe \* cantidad + iva).

- Estaría bien comprobar que no se acepten valores no numéricos en cantidad e importe
- Si queremos usar decimales en importe, podemos usar **parseFloat** en lugar de **parseInt** que usamos en otros ejemplos.
- El IVA lo podemos obtener usando la propiedad **value** del select

# Ejercicios iniciales JavaScript

Factura

Importe

Cantidad

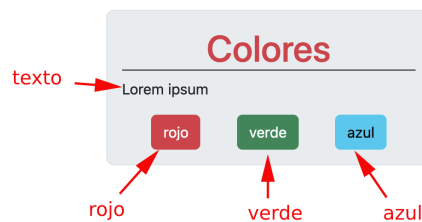
IVA

IVA

Total

## Ejemplo 12

El color del texto deberá cambiar según el botón pulsado. Para poner el color a un elemento usaremos `elemento.style.color = "color";` cambiando elemento por el nombre adecuado.



## Ejemplo 13

Como el ejercicio anterior pero ahora usando controles range:

- añadiremos un listener a cada range del tipo **input** en lugar de click, para que se vaya produciendo mientras el usuario mueva el deslizador
- con la propiedad **value** de los range obtendremos el valor actual, que en nuestro caso va de 0 a 255 (valores admitidos por rgb en CSS)
- Para cambiar el color vamos a crear la siguiente función fuera de las otras funciones. No os preocupéis de su funcionamiento por ahora:

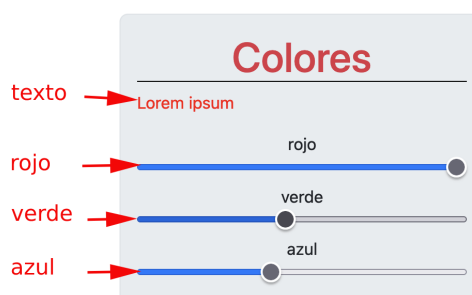
```
function rgb(rojo, verde, azul) {  
    return `rgb(${rojo}, ${verde}, ${azul})`; // El ` está después de la tecla p  
}
```

- Luego la usaremos donde corresponda de la forma:

```
elemento.style.color = rgb(valorRojo, valorVerde, valorAzul);
```

cambiando elemento, valorRojo, valorVerde y valorAzul por los adecuados

- Como el código de las tres funciones sería el mismo, podemos crear una sola y usarla en los tres addEventListener.



# Ejercicios iniciales JavaScript

## Ejemplo 14

Cuando el usuario pulse el botón Acceder debemos comprobar si el checkbox está marcado. Para ello usaremos la propiedad **checked** del mismo, que devuelve true si lo está y false si no.

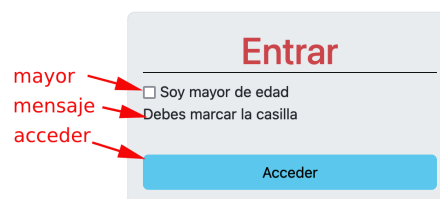
- Si no está marcado mostraremos el mensaje, que por defecto está oculto. Para ello eliminaremos la clase invisible de ese mensaje con el siguiente código:

```
elemento.classList.remove("invisible");
```

Cambiando elemento por el nombre adecuado.

- Sí está marcado iremos a una página cualquiera usando el siguiente código:

```
window.location.href = "URL deseada";
```



## Ejemplo 15

Al pulsar el botón Cambiar debemos mostrar el texto en castellano o en inglés, estando este último oculto por defecto. Para comprobar si un radio está marcado usaremos su propiedad **checked** la cual tendrá el valor true si lo está.

Para ocultar un elemento le añadiremos la clase **d-none** de Bootstrap con el siguiente código:

```
elemento.classList.add("d-none");
```

Cambiando elemento por el nombre adecuado.

Para mostrarlo haremos justo lo contrario:

```
elemento.classList.add("d-none");
```



## Ejemplo 16

El usuario elegirá un equipo en cada lista y, cuando pulse el botón, se mostrará en el párrafo partido, usando su propiedad **textContent** el equipo local y el visitante de la siguiente forma:

```
elemento.textContent = valorLocal + " - " + valorVisitante;
```



# Ejercicios iniciales JavaScript

Cambiando element, valorLocal y valorVisitante por el nombre adecuado. Con + podemos concatenar diferentes elementos.

Diagrama de la interfaz de usuario 'Partido'. Muestra dos selectores de equipo: 'Local' con 'Real Madrid' y 'Visitante' con 'RC Celta'. Debajo hay un botón 'mostrar'. Las etiquetas 'local' y 'visitante' apuntan a los selectores. Las etiquetas 'partido' y 'mostrar' apuntan al botón.

## Ejemplo 17

Cuando se pulse el botón se mostrará Hola y el nombre que se haya escrito en el cuadro de texto en un alert:

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

Si no se escribe un nombre se mostrará el mensaje debes escribir nombre. Para ello podemos comprobar si la propiedad **value** del elemento es igual a "" (cadena vacía).

Para mostrar hola y el nombre debemos usar algo como esto dentro del window.alert:

```
"Hola " + elemento.value
```

Cambiando elemento por el elemento adecuado

Diagrama de la interfaz de usuario 'Saludar'. Muestra un campo de texto 'Nombre:' y un botón 'Saludar'. La etiqueta 'nombre' apunta al campo de texto y la etiqueta 'saludar' apunta al botón.

## Ejemplo 18

Cuando se pulse el botón Calcular se mostrará el importe de descuentos realizados y el total final.

Diagrama de la interfaz de usuario 'Viaje'. Muestra un campo de texto 'Importe:' con el valor '100'. Debajo hay una sección 'Descuentos' con tres opciones seleccionadas: 'Jubilado: 10%', 'Ida y vuelta: 5%' y 'VIP: 7%'. Debajo de los descuentos hay un botón 'Calcular'. En la parte inferior hay dos campos de texto: 'Descuentos:' y 'Total:'. Las etiquetas 'importe', 'jubilado', 'idaVuelta', 'vip', 'calcular', 'descuentos' y 'total' apuntan a los respectivos elementos.

## Ejemplo 19

Al pulsar en el botón Añadir debe añadirse un li a la lista con el texto de la tarea. Para ello debemos crear un nodo li con **document.createElement**, un nodo texto con **document.createTextNode**,

[alonso.jad@gmail.com](mailto:alonso.jad@gmail.com)

# Ejercicios iniciales JavaScript

añadir el nodo texto al li con su método **appendChild(nodoTexto)** y finalmente añadir el nodo li al ul también con **appendChild(nodoLi)**, cambiando **nodoTexto** y **nodoLi** por los nombres correspondientes.

Al pulsar el botón Vaciar se debe eliminar todos los li de la lista; por ejemplo asignando "" a su **innerHTML**.

Estaría bien que no permitiese añadir tareas vacías, mostrando un mensaje en dicho caso.

Diagrama de la interfaz de usuario 'Tareas' con anotaciones de variables JavaScript:

- Tarea:** Etiqueta para el campo de entrada de texto.
- tarea**: Variable que apunta al campo de entrada de texto.
- Añadir**: Botón verde para agregar una nueva tarea.
- añadir**: Variable que apunta al botón 'Añadir'.
- Vaciar**: Botón rojo para eliminar todas las tareas.
- vaciar**: Variable que apunta al botón 'Vaciar'.
- Tareas:** Etiqueta para el contenedor de la lista de tareas.
- tareas**: Variable que apunta al contenedor de la lista de tareas.

La lista de tareas mostrada es:

- Comprar pan
- reunión de padres