

# Javascript

## Introducción

Surgió como una iniciativa de Netscape. **No hay ninguna relación entre Java y Javascript**. Algunas características de Javascript son:

- No se convierte en bytecodes, lo interpreta el navegador
- Está estandarizado aunque no todos los navegadores se ciñen al 100% al estándar. El nombre del estándar es «ECMAScript»
- No está orientado a objetos, sino que está basado en objetos.
- Es de tipado débil: esto significa que podemos cambiar una variable de tipo sin problemas, el intérprete intentará hacer las conversiones correctas.

## Tipos de datos

Javascript acepta los siguientes tipos de datos:

- Números.
- Cadenas
- Booleanos (lógicos)
- undefined: se utiliza cuando intentamos acceder a una variable que no contiene nada porque no se ha creado.
- null: se utiliza habitualmente para indicar algo vacío.

## El uso de `use strict`

Tradicionalmente Javascript permitía acceder a variables sin definir, lo que causaba infinidad de problemas. Si deseamos que Javascript sea más **estricto** en como se usa el lenguaje podemos incluir al principio de nuestros programas en Javascript la cadena «use strict».

## Constantes y variables

Podemos definir *constantes*, que son etiquetas que tienen dentro un valor que no cambia y *variables* que son etiquetas cuyo valor no cambia:

```
"use strict"
const gravedad = 9.81;
let posicion = 42;
```

¡Cuidado!, durante mucho tiempo, en lugar de `let` se usaba `var` pero el sistema de asignación de variables se modificó hace tiempo y el uso de `var` en general se considera **obsoleto**.

## Concatenación de cadenas

Tradicionalmente en Javascript las cadenas se han concatenado de la misma manera que en otros lenguajes:

```
"use strict"
let nombre = "Carmen";
let ap1 = "Diaz";
let ap2 = "Sanchez";
//Unimos todo...
let nombreCompleto=nombre+ " " + ap1 + " " + ap2;
```

Con ES6 se introdujeron las «plantillas» que permiten unir cadenas más cómodamente.

```
"use strict"
let nombre = "Carmen";
let ap1     = "Diaz";
let ap2     = "Sanchez";
//Ahora usamos una plantilla para unir...
let nombreCompleto=``;
```

## Advertencia:

Estas plantillas usan la «coma invertida», la que está a la derecha de la p en el teclado

## Incrustando Javascript

Javascript se inserta en HTML con la etiqueta `<script>`. Esta etiqueta puede ir en cualquier sitio del HTML, dentro de `<head>` o dentro de `<body>`.

Un programa muy simple sería este:

```
"use strict"
let una_variable
una_variable=42
document.write(una_variable)
```

## Decisiones

Las decisiones se toman con la sentencia `if` que funciona exactamente igual que en Java. Se pueden utilizar los mismos operadores `&&` y `||` al igual que en Java.

```
"use strict"
if (una_variable > otra_variable) {
    document.write ("La primera es mayor que la segunda")
} else {
    document.write ("La segunda es mayor que la primera")
}
```

## Vectores o Arrays

En Javascript los arrays pueden almacenar elementos de distinto tipo. Al crearlos podemos indicar el tamaño o no, pero no habrá problemas si queremos almacenar más elementos de los previstos.

```
"use strict"
/* Una forma de crear un array*/
vector_nombres=new Array()
vector_nombres[0]="Juan Perez"
vector_nombres[1]="Pedro Diaz"
document.write ("El primer nombre es:"+vector_nombres[0])

/* Otra forma de crearlos*/
vector_numeros=new Array(2)
vector_numeros[0]=23
vector_numeros[1]=-45.23
vector_numeros[2]=45e2
```

## Bucles

### Bucles for

## Bucles for estilo clásico

En estos bucles hay que poner la inicialización, la condición de final y la actualización:

```
"use strict"
for (let i=0; i<vector_numeros.length; i++){
    document.write("<br/>")
    document.write ("En la posición "+i)
    document.write (" está el número " + vector_numeros[i])
}
```

Obsérvese que hemos introducido el atributo `length` de la clase `Array` que nos indica la longitud del vector.

### Ejercicio

Crear un vector de 6 posiciones y rellenarlo con estos números: 9.98, 7.86, 4.53, 8.91, 5.76, 2.31.

Ordenar el vector y mostrar el contenido del vector ordenado por pantalla.

```
let v=new Array()
v=[9.98, 7.86, 4.53,
   8.91, 5.76, 2.31]

/* Vamos cogiendo cada elemento...*/
for (let i=0; i<v.length; i++){
    /* Y se compara con
       * todos los demás*/
    for (let j=0; j<v.length; j++) {
        if (v[j]>v[i]) {
            aux=v[i]
            v[i]=v[j]
            v[j]=aux
        } /* Fin del if*/
    } /* Fin del for interno*/
} /* fin del for externo*/

/* Se imprime el contenido*/
for (let i=0; i<v.length; i++){
    alert ("Pos "+i+ ":"+v[i])
}
```

## Bucles foreach

Funciona igual que el anterior pero es mucho más corto.

```
"use strict"
for (let posicion in vector_numeros) {
    document.write("<br/>")
    document.write ("En la posición "+posicion)
    document.write (" está el número " + vector_numeros[posicion])
}
```

## Bucles while

Los bucles `while` funcionan igual que en Java

```
"use strict"
let posicion=0
while (posicion<vector_numeros.length){
    document.write("<br/>")
    document.write ("En la posición "+posicion)
    document.write (" está el número " + vector_numeros[posicion])
```

```

        posicion++
    }
}
```

## Ejercicio: media aritmética

Crear un programa que calcule la media aritmética del vector de números.

```

"use strict"
let suma=0
for (var pos in vector_numeros){
    suma=suma+vector_numeros[pos]
}
let media=suma / vector_numeros.length
document.write("<br/>La media es:" + media)
```

## Ejercicio: desviación media

Crear un programa que calcule la desviación media del vector de números.

```

"use strict"
/* Para calcular La desviación media*/
let suma=0
for (let pos in vector_numeros) {
    let desviacion= Math.abs ( vector_numeros[pos] - media )
    suma = suma + desviacion
}
/* En este punto La variable suma contiene La suma de Las desviaciones*/
let desv_media = suma / vector_numeros.length
document.write("<br/>La desv media es:"+desv_media)
```

## Ejercicio: la mediana

Calcular la mediana del vector

```

"use strict"
if (v.length%2==0) {
    let pos1=v.length/2
    let pos2=pos1-1
    let elem1=v[pos1]
    let elem2=v[pos2]
    let mediana=(elem1+elem2)/2
} else {
    let pos_central=(v.length-1)/2
    let mediana=v[pos_central]
}
document.write("La mediana es:"+mediana)
```

## Funciones

Para crear una función usaremos la palabra `function`, pondremos el nombre, luego los parámetros, dentro irá el código de la función, y si queremos devolver algo usaremos `return`.

```

"use strict"
/* Función a la que Le pasamos un vector de números y que
 * nos devuelve La media de sus valores*/
function calcularMedia(vector_valores){
    let suma=0
    for (let pos in vector_valores){
        suma = suma + vector_valores[pos]
    }
    return suma / vector_valores.length
```

```

        }

let vector=new Array(4)
vector[0]=5
vector[1]=2
vector[2]=7
vector[3]=8

let media=calcularMedia(vector)
document.write("<br/>La media es:"+media)

```

Una cuestión importante es que las funciones son valores asignables. Cuando queramos asignar una función a una variable **no pondremos paréntesis**. Cuando sí queramos ejecutar una función (ya sea con su nombre original o con el de la variable, sí pondremos los paréntesis con los parámetros que queramos pasar\*\*.

```

"use strict"
function saludar(nombre){
    document.write("Hola "
                  +nombre+"<br/>")
}
function despedir(nombre){
    document.write("Adios "
                  +nombre+"<br/>")
}
saludar("Antonio")
despedir("Antonio")
/* Las funciones son valores
   * asignables*/
let funcion=despedir
funcion("Tomas")

```

## Valores por defecto

Desde ES6 las funciones aceptan valores por defecto. Si a la función no le pasamos un cierto argumento, entonces la función usará los asignados por defecto:

```

"use strict"
function saludar(tipoSaludo="Hola", persona="Juan"){
    alert(` ${tipoSaludo} ${persona}`)
}
//Escribe "Hola Juan"
saludar()
//Escribe "Bienvenido Juan"
saludar("Bienvenido")
//Escribe "Me alegro de verte Carmen"
saludar("Me alegro de verte", "Carmen")

```

## Ejercicio

Crear un programa que tenga una función que calcule la desviación media de valores de un vector.

```

"use strict"
/* Función que calcula la desviación media de
   * un vector de valores numéricos*/
function calcularDesviacionMedia(vector_valores){
    let media=calcularMedia(vector_valores)
    let suma=0
    for (let pos in vector_valores){
        suma= suma + Math.abs ( vector_valores[pos] - media )
    }
    return suma / vector_valores.length
}

```

## Ejercicio

Crear un programa que tenga una función que calcule la moda.

```

"use strict"
/* Este vector nos dice cuantas veces aparece un número en un vector*/
function calcularFrecuencia(numero, vector){
    var num_vezes=0
    for (let pos in vector) {
        if (vector[pos]==numero) {
            num_vezes++
        }
    }
    return num_vezes
}

/* Dado un vector de números se nos devuelve La posición
 * del número mayor*/
function obtenerPosMayor(vector_valores){
    let posMayor=0
    let numMayor=vector_valores[0]
    for (let pos in vector_valores){
        if (vector_valores[pos]>numMayor) {
            numMayor=vector_valores[pos]
            posMayor=pos
        }
    }
    return posMayor
}
/* Función que devuelve el número "moda" de un vector*/
function obtenerModa(vector_valores){
    let frecuencias=new Array(vector_valores.length)
    for (let pos in vector_valores){
        let numero=vector_valores[pos]
        frecuencias[pos]=calcularFrecuencia(numero, vector_valores)
    }
    let posModa=obtenerPosMayor(frecuencias)
    return vector_valores[posModa]
}

let vector=new Array(4)
vector[0]=7
vector[1]=7
vector[2]=7
vector[3]=5
var moda=obtenerModa(vector)
document.write("<br/>La moda es:"+moda)

```

## Programación OO

Se ha dicho anteriormente que Javascript es «basado en objetos» y no «orientado a objetos», es decir la POO es optativa. No por ello es menos potente.

En primer lugar, es posible crear objetos sin crear clases.

```

"use strict"
let empleado={
    nombre:"Pepe Perez",
    edad:27,
    fijo:true,
    estaJubilado:function (){
        if (this.edad>65) {
            return true
        } else {
            return false
        }
    }
}

```

```

        }
    }
    document.write("<br/>El nombre es:" + empleado.nombre)
    document.write("<br/>¿Jubilado?" + empleado.estaJubilado() )
}

```

## Ejercicio

Añadir un método llamado `nivelExperiencia` que nos diga una de estas cosas:

- Nos debe devolver «junior» si la edad está entre 18 y 25
- Nos debe devolver «asociado» si la edad está entre 26 y 45
- Nos debe devolver «senior» si la edad está entre 46 y 60
- Nos debe devolver «experto» si la edad está entre 61 y 65
- Nos debe devolver «no aplicable» si la edad es mayor de 65

```

"use strict"
let empleado={
    nombre:"Pepe Perez",
    edad:27,
    fijo:true,
    estaJubilado:function (){
        if (this.edad>65) {
            return true
        } else {
            return false
        }
    },
    nivelExperiencia:function(){
        if ( (this.edad>18)  && (this.edad<=25) ){
            return "junior"
        }
        if ( (this.edad>=26)  && (this.edad<=45) ){
            return "asociado"
        }
    }
}

```

## Ejercicio

Crear una clase `GestorVectores` que tenga los principales métodos estadísticos vistos hasta ahora: media, desviación media, mediana y moda.

```

"use strict"
gestor_vectores={
    vector_numeros:new Array(),
    setDatos:function(vector){
        this.vector_numeros=vector
    }
    , //Importante: separar métodos y atributos con ,
    getMedia:function(){
        let suma=0
        let media=0
        for (pos in this.vector_numeros) {
            suma=suma + this.vector_numeros[pos]
        }
        media=suma / this.vector_numeros.length
        return media
    }
    ,
    getModa:function(){
}

```

```

        }
        ,
        getMediana:function(){
            this.vector_numeros.sort()
        }
    }

let vector_prueba=new Array(3)
vector_prueba[0]=5
vector_prueba[1]=10
vector_prueba[2]=8
gestor_vectores.setDatos ( vector_prueba )
let media=gestor_vectores.getMedia()
document.write ("La media es:"+media)

```

## Programación con JQuery

Existen muchos navegadores que a veces muestran pequeñas diferencias entre ellos. Para evitar problemas los programadores tenían que incluir muchos código para comprobar qué navegador ejecutaba su JS y en función de eso actuar. Para resolver estas diferencias John Resig creó JQuery.

### Inicio

A partir de ahora todos nuestros archivos HTML tendrán que cargar al comienzo la biblioteca JQuery con una etiqueta como esta:

```

<script src="jquery.js" language="Javascript">
</script>
<script src="nuestroprograma.js" language="Javascript">
</script>

```

### **El orden es importante**

### La función \$

La función \$ selecciona elementos de la página para que podamos hacer cosas con ellos. Es la función más utilizada de JQuery y veremos que podemos pedir que nos seleccione grupos de cosas de forma muy sencilla.

La función \$ devuelve siempre objetos. Los atributos y métodos de esos objetos los iremos aprendiendo poco a poco.

En general, antes de poder procesar un elemento, deberemos seleccionarlo utilizando los mismos selectores que en CSS.

### Gestión de eventos

Utilizando click podemos indicar a la biblioteca que queremos que cuando alguien haga click en un elemento se ejecute una cierta función. El siguiente código HTML y JS ilustra una posibilidad

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script src="ejemplo.js"></script>
<title>Ejemplos</title>
<style>
    div#texto{
        background-color:yellow;
    }

```

```

</style>
</head>

<body>
<form>
    <input type="button" value="fadeOut" id="botonizq">
    <input type="button" value="fadeIn" id="botonder">
</form>

<div id="texto">
    Texto texto texto
</div>

</body>
</html>

```

El código Javascript asociado al HTML anterior es este.

```

/* Esperaremos hasta que el documento esté cargado y Listo
 * para ser procesado por nuestro programa*/

let obj_documento = $(document)

/* Cuando esté cargado ejecutaremos la función cuyo nombre aparezca aquí*/
obj_documento.ready(inicio)

/** Error gravísimo/
//obj_documento.ready( inicio() )

function inicio(){
    let obj_izq=$("#botonizq")
    obj_izq.click ( fn_click_izq )
    let obj_der=$("#botonder")
    obj_der.click ( fn_click_der )
}

function fn_click_izq(){
    let obj_div=$("#texto")
    obj_div.fadeOut()
}

function fn_click_der(){
    let obj_div=$("#texto")
    obj_div.fadeIn()
}

```

## Solución HTML (párrafos)

```

<div data-role="content">
    <div class="ui-grid-c">
        <div class="ui-block-a">
            <input type="submit"
                id="mostrar_pares"
                value="Mostrar pares">
        </div>
        <div class="ui-block-b">
            <input type="submit"
                id="ocultar_pares"
                value="Ocultar pares">
        </div>
        <div class="ui-block-c">
            <input type="submit"
                id="mostrar_impares"
                value="Mostrar impares">
        </div>
        <div class="ui-block-d">

```

```

<input type="submit"
       id="ocultar_impar"
       value="Ocultar impares">
</div>
</div>
<p class="p_impar">
    Soy un párrafo impar
</p>
<p class="p_par">
    Soy un párrafo par
</p>
<p class="p_impar">
    Soy un párrafo impar
</p>
<p class="p_par">
    Soy un párrafo par
</p>
<p class="p_impar">
    Soy un párrafo impar
</p>
<p class="p_par">
    Soy un párrafo par
</p>
</div>

```

## Solución Javascript (párrafos)

```

$(document).ready(main)

function mostrar_pares(){
    let objetos=$(".p_par")
    objetos.slideDown()
}
function mostrar_impar(){
    let objetos($(".p_impar"))
    objetos.slideDown()
}
function ocultar_pares(){
    let objetos($(".p_par"))
    objetos.slideUp()
}
function ocultar_impar(){
    let objetos($(".p_impar"))
    objetos.slideUp()
}

function main(){
    $("#mostrar_pares").click(mostrar_pares)
    $("#mostrar_impar").click(mostrar_impar)

    $("#ocultar_pares").click(ocultar_pares)
    $("#ocultar_impar").click(ocultar_impar)

}

```

Existen diversos eventos aunque los más utilizados son:

- click
- dblclick
- mouseover

## Ejercicio

Crear un programa que tenga varios párrafos con 4 botones que permitan que cuando se haga click en ellos ocurran distintas cosas

- Habrá un botón con el texto «Ocultar pares». Cuando se hace click en él se ocultan los párrafos pares.
- Habrá un botón con el texto «Ocultar impares». Cuando se hace click en él se ocultan los párrafos impares.
- Habrá un botón con el texto «Mostrar pares». Cuando se hace click en él se muestran los párrafos pares (que tal vez estaban ocultos).
- Habrá un botón con el texto «Mostrar impares». Cuando se hace click en él se muestran los párrafos impares (que tal vez estaban ocultos).

## Ejercicio

Crear una página en la que hay un div con texto y al pasar el ratón por encima de ella, la caja cambia de color.

Antes de poder resolver este ejercicio, hay que echar un vistazo a varias posibilidades de JQuery.

## Procesado de atributos

En JQuery sabemos que podemos procesar elementos utilizando su `id` con cosas como esta:

```
var objeto=$("#identificador1")
objeto.metodo( ... )
```

Una de las cosas que se puede hacer es leer y escribir diversos atributos de los objetos. Además, se pueden leer propiedades especiales como comprobar si un radio o un checkbox están en el estado `checked`.

Supongamos este formulario:

```
<form>
<input type="radio" name="sexo" value="h" id="opc_h">Hombre
<br/>
<input type="radio" name="sexo" value="m" id="opc_m">Mujer
<br/>
<input type="text" id="informe">
<br/>
<input type="checkbox" name="medios[]" id="bus">Autobús
<br/>
<input type="checkbox" name="medios[]" id="coche">Coche
<br/>
<input type="checkbox" name="medios[]" id="moto">Moto
<br/>
<input type="checkbox" name="medios[]" id="bici">Bici
<br/>
<input type="checkbox" name="medios[]" id="tren">Tren
<br/>

</form>
```

Podemos usar el método `val` para cambiar el valor de un objeto cualquiera:

```
function inicio(){
    let opc_h=$("#opc_h")
    opc_h.click ( click_hombre )

    let opc_m=$("#opc_m")
    opc_m.click ( click_mujer )
}

function click_hombre() {
    let cuadro_texto=$("#informe")
```

```

        cuadro_texto.val("Bienvenido Sr.")
    }

    function click_mujer(){
        let cuadro_texto=$("#informe")
        cuadro_texto.val("Bienvenido Sra/Srta.")
    }

```

Por ejemplo, en los checkboxes y en los radios, podemos comprobar si uno de ellos está marcado comprobando la propiedad «checked» con el método prop.

Supongamos que deseamos saber cuantos checkboxes se marcan. Si se marcan cero, una o dos, mostraremos el texto «poca variedad», si se marcan tres mostraremos «cierta variedad» y si se marcan cuatro o cinco, mostraremos «mucha variedad».

Aquí hay dos posibles soluciones, siendo una de ellas más corta y flexible que la otra.

La primera:

```

let opc_coche=$("#coche")
opc_coche.click ( cuantas_pulsadas )

let opc_moto=$("#moto")
opc_moto.click ( cuantas_pulsadas )

let opc_bici=$("#bici")
opc_bici.click ( cuantas_pulsadas )

let opc_bus=$("#bus")
opc_bus.click ( cuantas_pulsadas )

let opc_tren=$("#tren")
opc_tren.click ( cuantas_pulsadas )

function cuantas_pulsadas(){
    //Aqui contariamos cuantas tienen la propiedad checked
}

```

El segundo implica que todos los controles tengan el mismo atributo class. Ahora la solución tendría un HTML como este:

```

<input type="checkbox" name="medios[]" id="bus" class="medio">Autobús
<br/>
<input type="checkbox" name="medios[]" id="coche" class="medio">Coche
<br/>
<input type="checkbox" name="medios[]" id="moto" class="medio">Moto
<br/>
<input type="checkbox" name="medios[]" id="bici" class="medio">Bici
<br/>
<input type="checkbox" name="medios[]" id="tren" class="medio">Tren
<br/>

```

Y el JS sería así:

```

let medios_de_locomocion$(".medio")
medios_de_locomocion.click ( cuantas_pulsadas )
function cuantas_pulsadas(){
    let cuantas_marcadas=0
    let vector_ids=["#bus", "#coche", "#moto", "#bici", "#tren"]

    for (let pos in vector_ids){
        var objeto = $( vector_ids[pos] )
        if (objeto.prop("checked")){
            cuantas_marcadas=cuantas_marcadas+1
        }
    }
}

```

```

        }

        if ((cuantas_marcadas>=0 ) && (cuantas_marcadas<=2)){
            alert ("Poca variedad")
        }
        if (cuantas_marcadas==3) {
            alert ("Variedad media")
        }
        if (cuantas_marcadas>=4){
            alert ("Mucha variedad")
        }
    }
}

```

## Ejercicio: recuento de medios de locomoción

Crear un programa que permita al usuario indicar cinco posibles medios de locomoción (usar checkboxes), a saber: coche, moto, bus, tren y avión. El programa debe contabilizar cuantos se usan en informar del número de medios usados en un textbox.

## Solución: recuento de medios (HTML)

```

<div class="ui-grid-d">
<div class="ui-block-a">
    <input type="checkbox"
        name="medio"
        id="coche">
    <label for="coche">Coche</label>
</div>
<div class="ui-block-b">
    <input type="checkbox"
        name="medio"
        id="moto">
    <label for="moto">Moto</label>
</div>
<div class="ui-block-c">
    <input type="checkbox"
        name="medio"
        id="bus">
    <label for="bus">Bus</label>
</div>
<div class="ui-block-d">
    <input type="checkbox"
        name="medio"
        id="tren">
    <label for="tren">Tren</label>
</div>
<div class="ui-block-e">
    <input type="checkbox"
        name="medio"
        id="avion">
    <label for="avion">Avión</label>
</div>
</div>
<input type="text" id="informe">

```

## Solución: recuento de medios (JS)

Variante 1: Sin vectores, implica usar muchos if. Aunque funcione supone cortar y pegar, que aunque en este caso no sea un trabajo muy grande nos obliga a adoptar malos hábitos

```

$(document).ready(main)

function contar(){
    let contador=0

```

```

    if ($("#coche").prop("checked"))
    {
        contador=contador+1
    }
    if ($("#moto").prop("checked"))
    {
        contador=contador+1
    }
    let mensaje="Medios:"+contador
    $("#informe").val(mensaje)

}

function main(){
    $("#coche").click(contar)
    $("#moto").click(contar)
    $("#bus").click(contar)
    $("#tren").click(contar)
    $("#avion").click(contar)

}

```

Variante 2: Con vectores

```

$(document).ready(main)
function contar(){
    let contador=0
    let ids=new Array()
    let ids=["#coche", "#moto",
             "#bus", "#tren",
             "#avion"]

    for (let pos in ids){
        let medio=$("#"+ids[pos])
        if (medio.prop("checked"))
        {
            contador=contador+1
        }
    }
    let mensaje="Medios:"+contador
    $("#informe").val(mensaje)

}

function main(){
    $("#coche").click(contar)
    $("#moto").click(contar)
    $("#bus").click(contar)
    $("#tren").click(contar)
    $("#avion").click(contar)
}

```

## Ejercicio configurador

Se desea tener una aplicación que permita configurar un equipo al gusto del usuario:

- Se debe elegir entre un procesador Intel o AMD. El primero cuesta 250 euros y el segundo 230.
- Se debe elegir entre 2, 4 y 8 GB de memoria. El coste es respectivamente 90, 145, 210
- Hay extras que se pueden elegir o no, ya que son completamente optativos (es decir, usar checkboxes). En concreto se puede tener un grabador de Blu-ray (190 euros), tarjeta gráfica aceleradora (430 euros) y un monitor LED (185 euros).

## Solución HTML configurador

```

<label for="intel">Intel i5</label>
<input type="radio"
       name="procesador" id="intel">
<label for="amd">AMD</label>
<input type="radio"
       name="procesador" id="amd">
<label for="2gb">2GB</label>
<input type="radio"
       name="memoria" id="2gb">
<label for="4gb">4 GB</label>
<input type="radio"
       name="memoria" id="4gb">
<label for="8gb">8 GB</label>
<input type="radio"
       name="memoria" id="8gb">
<label for="bluray">Blu-ray</label>
<input type="checkbox" name="extra[]"
       id="bluray">
<label for="aceleradora">Aceleradora</label>
<input type="checkbox" name="extra[]"
       id="aceleradora">
<label for="monitor">Monitor 25</label>
<input type="checkbox" name="extra[]"
       id="monitor">
<input type="text" id="total">

```

## Solución JS configurador

```

$(document).ready(main)

function calcular_precio(){
    var precio=0
    if ($("#intel").prop("checked")) {
        precio=precio+250
    }
    if ($("#amd").prop("checked")) {
        precio=precio+210
    }
    if ($("#2gb").prop("checked")) {
        precio=precio+90
    }
    if ($("#4gb").prop("checked")) {
        precio=precio+140
    }
    if ($("#8gb").prop("checked")) {
        precio=precio+210
    }
    if ($("#bluray").prop("checked")) {
        precio=precio+190
    }
    if ($("#aceleradora").prop("checked")) {
        precio=precio+430
    }
    if ($("#monitor").prop("checked")) {
        precio=precio+185
    }

    $("#total").val(precio)
}
function main(){
    $("#intel").click (calcular_precio)
    $("#amd").click (calcular_precio)

    $("#2gb").click (calcular_precio)
    $("#4gb").click (calcular_precio)
    $("#8gb").click (calcular_precio)
}

```

```

    $("#bluray").click (calcular_precio)
    $("#monitor").click (calcular_precio)
    $("#aceleradora").click (calcular_precio)
}

```

## Ejercicio configurador de PCs ampliado

En el ejercicio anterior ocurre que por un problema hardware no es posible tener procesadores AMD con aceleradora, por lo que cuando se marque un AMD se debe desactivar el checkbox de la aceleradora y si hubiera una marca, también se debe desactivar y por supuesto recalcular el precio.

## Configurador de coches

Un fabricante de automóviles desea ofrecer a sus clientes una aplicación que les permita configurar sus vehículos según sus preferencias y ver el precio final del coche. Los precios y las restricciones son los siguientes:

- Se pueden tener dos tipos de motor: gasolina (precio base 7000 euros) y diésel (precio base 8200).
- Se pueden tener 3 potencias: 1100, 1800 y 2300 centímetros cúbicos. Los precios de cada uno son 800, 1900 y 2500. Sin embargo **no es posible fabricar motores diésel de 2300**.
- Hay dos tipos de pintura: normal y metalizada. Los precios respectivos son 750 y 1580 euros.
- Hay seis colores: negro, blanco, rojo, azul polar, verde turquesa y gris marengo. **No se pueden fabricar colores de pintura normal de ninguno de los tres últimos colores**.
- Se dispone de diversos extras: alerón deportivo (190 euros **pero solo se puede elegir si se elige pintura metalizada**), radio-CD con MP3 (230 euros más), altavoces traseros (320 euros más, **pero solo si se elige antes el Radio-CD**), y GPS incorporado (520 euros más).

Crear la aplicación que respete las restricciones exigidas por el cliente.

## HTML del configurador

```

<form>Tipo de motor<br/>
    <input type="radio" name="motor"
        id="gasolina">Gasolina <br/>
    <input type="radio" name="motor"
        id="diesel">Diésel<br/>
    <br/> Cubicaje<br/>
    <input type="radio" name="cubicaje"
        id="1100">1100m3<br/>
    <input type="radio" name="cubicaje"
        id="1800">1800m3<br/>
    <input type="radio" name="cubicaje"
        id="2300">2300m3<br/>
    <br/>Pinturas<br/>
    <input type="radio" name="pintura"
        id="normal">Normal<br/>
    <input type="radio" name="pintura"
        id="metalizada">Metaliz.<br/>
    <br/>Colores<br/>
    <input type="radio" name="color"
        id="negro">Negro
    <input type="radio" name="color"
        id="blanco">Blanco
    <input type="radio" name="color"
        id="rojo">Rojo
    <br/>
    <input type="radio" name="color"
        id="azulpolar">Azul polar
    <input type="radio" name="color"
        id="verde">Verde
    <input type="radio" name="color"

```

```

    id="gris">Gris
<br/>Extras<br/>
<input type="checkbox" id="aleron">
Alerón
<input type="checkbox" id="radiocd">
radiocd
<input type="checkbox" id="altavoces">
Altavoces
<input type="checkbox" id="gps">
GPS
<button
    onclick="calcular();return false">
    Calcular precio
</button>
</form>

```

## JS del configurador (sin JQuery)

```

function esta_checked(identificador){
    let control;
    control=
        document.getElementById(
            identificador);
    if (control.checked){
        return true;
    }
    return false;
}
function hay_errores(){
    let error=false;
    /* Si detectamos un error
    ponemos error a true*/
    if (esta_checked("diesel") && (esta_checked("2300"))){
        alert("Imposible diesel de 2300, por favor modifique sus opciones.");
        error=true;
    }
    if (esta_checked("normal") && (esta_checked("azulpolar"))){
        alert("Imposible normal+azul");
        error=true;
    }
    if (esta_checked("normal") && (esta_checked("gris"))){
        alert("Imposible normal+gris");
        error=true;
    }
    if (esta_checked("normal") && (esta_checked("verde"))){
        alert("Imposible normal+verde");
        error=true;
    }
    if (esta_checked("normal") && (esta_checked("aleron"))){
        alert("Imposible normal+aleron. Elija pintura metalizada");
        error=true;
    }
    /*Cuidado, aquí comprobamos si el altavoz está marcado pero sin haber marcado el radioCD
    if (esta_checked("altavoces") && (!esta_checked("radiocd"))){
        alert("No puede pedir altavoces sin pedir el RadioCD");
        error=true;
    }
    return error;
}
function calcular(){
    let precio=0;
    if (hay_errores()){
        return false;
    }
    /* Si llegamos aquí es que

```

```

no hay errores y el precio se puede calcular*/
if (esta_checked("gasolina")){
    precio=precio+7000;
}
if (esta_checked("diesel")){
    precio=precio+8200;
}
if (esta_checked("1100")){
    precio=precio+800;
}
if (esta_checked("1800")){
    precio=precio+1900;
}
if (esta_checked("2300")){
    precio=precio+2500;
}
if (esta_checked("normal")){
    precio=precio+750;
}
if (esta_checked("metalizada")){
    precio=precio+1580;
}
/*Obsérvese que el precio
no ha cambiado dependiendo del color*/
if (esta_checked("aleron")){
    precio=precio+190;
}
if (esta_checked("radiocd")){
    precio=precio+230;
}
if (esta_checked("altavoces")){
    precio=precio+320;
}
if (esta_checked("gps")){
    precio=precio+520;
}
alert("Su precio es:"+precio);
}

```

## Dinamismo con Google Maps

Google Maps ofrece un servicio de mapas con una limitación de 25.000 peticiones diarias. El código básico sería así:

### HTML de GMaps

```

<!DOCTYPE html>

<html>
<head>
    <!--En móviles poner la escala inicial a 1-->
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!--Cargamos los estilos y los efectos de Bootstrap-->
    <link
        rel="stylesheet"
        type="text/css" href="bootstrap/css/bootstrap.css"/>
    <script src="bootstrap/js/bootstrap.js"></script>
    <script src="js/jquery.js"></script>
    <script
        src="http://maps.googleapis.com/maps/api/js?key=AIzaSyDpv9zCj9szIIu--LuNmDsry2fZCRrc
            </script>
    <style>
        #mapa{

```

```

        width: 500px;
        height: 500px;
        float: right;
        background-color: rgb(180,190,240);
    }
    #controles{
        float: left;
    }

```

</style>

<script src="js/programa.js"></script>

<title>Plantilla JQuery</title>

</head>

<body>

<div class="container">

<h1>Javascript con mapas</h1>

<div id="controles">

    Introduzca latitud:<input type="text" id="latitud">

    <br/>

    Introduzca longitud:<input type="text" id="longitud">

    <br/>

    <input type="submit" id="mover" value="¡Viajar!">

    <select id="ciudades">

        <option value="CR">Ir a Ciudad Real</option>

        <option value="BA">Ir a Barcelona</option>

        <option value="PO">Ir a Pontevedra</option>

    </select>

    <br/>

    Calculador de distancias desde CR a otras ciudades

    <select id="ciudades">

        <option value="CR">Ir a Ciudad Real</option>

        <option value="BA">Ir a Barcelona</option>

        <option value="PO">Ir a Pontevedra</option>

    </select>

</div>

<div id="mapa">

</div>

</div>

</body>

</html>

## Javascript de GMaps

```

var latitud=38.59
var longitud=-3.55
var mi_nivel_de_zoom=8
var obj_mapa
function inicio(){
    var div_mapa=document.getElementById("mapa")
    var obj_coordenadas=new google.maps.LatLng(latitud,longitud)
    var obj_opciones={
        center:obj_coordenadas,
        zoom:mi_nivel_de_zoom
    }
    obj_mapa=new google.maps.Map(div_mapa, obj_opciones)

    $("#mover").click (mover_el_mapa)
}

function mover_el_mapa() {
    var obj_latitud=$("#latitud")
    var valor_latitud=obj_latitud.val()
}

```

```

var valor_longitud=$("#longitud").val()
var nuevas_coordenadas=new google.maps.LatLng(
    valor_latitud, valor_longitud)
obj_mapa.panTo(nuevas_coordenadas)
}

```

## Ejercicio

Ampliar el programa con una lista de ciudades del mundo. Cuando el usuario elija una de ellas, nuestro programa nos dirá la distancia desde Ciudad Real a dichas ciudades. Considerar las siguientes coordenadas en formato (latitud, longitud):

- Ciudad Real: (38.59, -3.55)
- Nueva York: (40.73, -73.87)
- Sidney: (-33.90, 151.13)
- Berlin: (52.31, 13.39)
- París: (48.85, 2.35)

Para poder conseguir esto, hay que modificar la URL de carga de GoogleMaps para solicitar que se cargue una biblioteca que nos ayudará a resolver este punto. En concreto, ahora pasaremos un parámetro `libraries` con el valor `geometry` que nos permitirá utilizar la biblioteca en concreto. Ahora el HTML es así:

```

<script src="http://maps.googleapis.com/maps/api/js?key=AIzaSyDpv9zCj9szIIu--LuNmDsry2fZ
</script>

```

Ahora una función que nos calcula la distancia sería algo como esto:

```

/* Nos da La distancia en metros entre CR
 * y el punto (latitud_destino, longitud_destino)
 * (abreviados lat_dest y lng_dest)*/
function distancia(lat_dest, lng_dest)
{
    let latitud_cr=38.59
    let longitud_cr=-3.55
    let coords_origen=new google.maps.LatLng(
        latitud_cr, longitud_cr)
    let coords_destino=new google.maps.LatLng(
        lat_dest, lng_dest)
    let distancia=google.maps.geometry.spherical.computeDistanceBetween(
        coords_origen, coords_destino
    )
    return distancia
}

```

## Ejercicio

La empresa Automobile Creation for Millenium Enterprise (ACME) planea lanzar una página web en la que se permita al usuario configurar los coches a su medida, ofreciendo las distintas opciones en pantalla para que el usuario las elija. Sin embargo no todas las combinaciones se permiten en fábrica por lo que deberán tenerse en cuenta las siguientes

## Especificaciones

- Hay dos motores: gasolina (5000) y diésel (6800)
- Hay dos carrocerías: monovolumen (4500) y berlina (3700)
- Hay tres accesorios: radio-cd con MP3 (180), alerones deportivos (220) y llantas de aleación (200)

Por diversos problemas, no es posible combinar las siguientes opciones:

- No se pueden tener berlinas de gasolina.

- No se puede integrar el alerón en los monovolúmenes.
- No se puede poner el radio-cd a los monovolúmenes.

Cuando se marque cualquiera de estas opciones, hay que limpiar todo el configurados y avisar de que no se puede hacer eso. No se pueden usar alerts

```

let vector_ids=["#gasolina", "#diesel", "#monovolumen",
                "#berlina", "#radiocd", "#alerones", "#llantas"]
let precios=[5000, 6800, 4500,
            3700, 180, 220, 200]

function inicio(){

    for (let pos in vector_ids){
        let el_id=vector_ids[pos]
        $(el_id).click ( calcularPrecio )
    }
}

function cocheEsFabricable() {
    //Caso 1: nada de berlinas de gasolina
    let marcada_la_berlina=$("#berlina").prop("checked")
    let marcada_la_gasolina=$("#gasolina").prop("checked")
    if (marcada_la_berlina && marcada_la_gasolina) {
        alert ("No se pueden fabricar berlinas de gasolina")
        return false
    }

    let marcado_alaron=$("#alerones").prop("checked")
    let marcado_monovolumen=$("#monovolumen").prop("checked")
    if (marcado_alaron && marcado_monovolumen ) {
        alert ("No podemos integrar los alerones en monovolúmenes")
        return false
    }

    let marcado_radiocd=$("#radiocd").prop("checked")
    if (marcado_radiocd && marcado_monovolumen) {
        alert ("No podemos fabricar un monovol. con radio-cd")
        return false
    }
    return true
}
/* Calcula el precio del coche en función de lo que esté marcado
 * y lo que no.*/
function calcularPrecio() {
    let todo_bien=cocheEsFabricable()
    if (todo_bien!=true) {
        return
    }
    let precioCoche=0
    for (var pos in vector_ids){
        let el_id=vector_ids[pos]
        if ($(el_id).prop("checked")) {
            let precio_accesorio=precios[pos]
            precioCoche=precioCoche+precio_accesorio
        }
    }
    alert ("El precio es:"+precioCoche)
}

```

## Ampliación

Se desea que el usuario puede elegir entre los siguientes colores con los siguientes precios:

- Blanco: 700 euros
- Rojo, Verde y Azul básicos: 950

- Gris, Negro y Naranja: 1400 euros por ser colores metalizados

Además, se desea ver una muestra de color en algún punto de la página. Para lograrlo se necesitará utilizar un método que proporciona JQuery y que se llama `addClass`

## Solución HTML

```
<head>
  <style>
    .muestrarrojo{
      background-color:red;
    }
    .muestraverde{
      background-color: green;
    }
    .muestraazul{
      background-color: blue;
    }
    .muestragris{
      background-color: grey;
    }
    .muestrablanco{
      background-color: white;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Motores</h1>
    <input type="radio" id="gasolina" name="motor">Motor Gasolina
    <br/>
    <input type="radio" id="diesel" name="motor">Motor Diésel
    <br/>
    <h1>Carrocerías</h1>
    <input type="radio" id="monovolumen"
    name="carroceria">Monovolumen
    <br/>
    <input type="radio" id="berlina" name="carroceria">Berlina
    <br/>
    <h1>Accesorios</h1>
    <input type="checkbox" name="accesorios[]"
    id="radiocd">Radio-CD
    <br/>
    <input type="checkbox" name="accesorios[]"
    id="alerones">Alerones
    <br/>
    <input type="checkbox" name="accesorios[]"
    id="llantas">Llantas
    <br/>
    <h1>Colores</h1>
    <!--Los colores irán
    en una columna y La muestra en otra-->
    <div class="row">
      <div class="col-md-3">
        <input type="radio"
        name="colores"
        id="blanco">Blanco
        <br/>
        <input type="radio"
        name="colores"
        id="rojo">Rojo
        <br/>
        <input type="radio"
        name="colores" id="gris">Gris
      </div>
      <div class="col-md-9 center-block">
        <h2>Muestra de color</h2>
      </div>
    </div>
  </div>
</body>
```

```

        </div>
    </div><!--Fin de la fila-->
</div>
</body>

```

## Solución JS

Añadiremos este código a nuestro programa anterior.

```

let obj_documento = $(document)
obj_documento.ready(inicio)

let vector_ids=["#gasolina", "#diesel", "#monovolumen",
                "#berlina", "#radiocd", "#alerones", "#llantas"]
let precios=[5000, 6800, 4500,
            3700, 180, 220, 200]

function inicio(){

    for (let pos in vector_ids){
        let el_id=vector_ids[pos]
        $(el_id).click ( calcularPrecio )
    }
    $("#blanco").click ( ponerColorBlanco )
    $("#rojo").click ( ponerColorRojo )
    $("#gris").click ( ponerGris )
}
function limpiarColores(){
    let clases=["muestrarrojo", "muestrablanco",
                "muestragris"]
    for (let pos in clases) {
        $("#muestra").removeClass( clases[pos] )
    }
}
function ponerGris() {
    limpiarColores()
    $("#muestra").addClass("muestragris")
}
function ponerColorRojo(){
    limpiarColores()
    $("#muestra").addClass ("muestrarrojo")
}
function ponerColorBlanco(){
    limpiarColores()
    $("#muestra").addClass ("muestrablanco")
}

```

## Otro configurador de coches

Si no se utiliza JQuery se desea crear un configurador de coches en JS que responda a las siguientes premisas:

- Hay dos modelos a elegir: el modelo A cuesta 7000 euros y el modelo B cuesta 9000.
- Se pueden elegir dos tipos de motor. El motor de gasolina cuesta 2000 y el diésel 5000 euros.
- Se pueden elegir 0, 1, muchos o todos los extras siguientes: Pintura metalizada por 1000 euros más, pack de sonido por 500 euros más y pack de seguridad por 1000 euros más

## Configurador en HTML

```

<form>
    <h3>Elija un modelo</h3>
    <input type="radio" name="modelo"
           id="modelo_a">Modelo A<br/>

```

```

<input type="radio" name="modelo"
       id="modelo_b">Modelo B<br/>
<h3>Elija un tipo de motor</h3>
<input type="radio" name="motor"
       id="gasolina">Gasolina<br/>
<input type="radio" name="motor"
       id="diesel">Diésel<br/>
<h3>Elija extras</h3>
<input type="checkbox" name="extras"
       id="metalizada">Pintura metalizada<br/>
<input type="checkbox" name="extras"
       id="sonido">Extra sonido<br/>
<input type="checkbox" name="extras"
       id="seguridad">Extra seguridad<br/>
<input type="submit" value="Calcular"
       onclick="calcular();return false;">
</form>
<div id="zonaresultados"></div>

```

## Configurador en JS

```

function esta_checked(id){
    let control;
    control=document.getElementById(id);
    if (control.checked){
        return true;
    } else {
        return false;
    }
}

function calcular(){
    let preciototal =0;
    let preciomodelo =0;
    let preciomotor =0;
    let precioextras = 0;
    if ( (!esta_checked("modelo_a"))
        && (!esta_checked("modelo_b")) ){
        alert("Debe marcar un modelo");
        return ;
    }
    if (esta_checked("modelo_a")){
        preciomodelo=7000;
    }
    if (esta_checked("modelo_b")){
        preciomodelo=9000;
    }
    if (esta_checked("gasolina")){
        preciomotor=2000;
    }
    if (esta_checked("diesel")){
        preciomotor=4000;
    }
    if (esta_checked("metalizada")){
        precioextras=1000;
    }
    if (esta_checked("sonido")){
        precioextras=precioextras+500;
    }
    if (esta_checked("seguridad")){
        precioextras=precioextras+1000;
    }
    preciototal=preciomodelo+preciomotor+precioextras;

    let zonaresultados;
    zonaresultados=document.getElementById(
        "zonaresultados");
}

```

```

        zonaresultados.innerHTML="Precio:"+preciototal;
    }
}

```

## Calculo de impuestos

Se desea crear una pequeña aplicación web que permita calcular los impuestos que se deben pagar a partir de unos datos, en concreto:

- Se debe introducir el salario anual en un campo de tipo `number`.
- Hay dos radios que permiten indicar si el contribuyente tiene hijos o no.
- Hay dos checkboxes que permiten saber si el contribuyente tiene derecho a algunas modificaciones llamadas B1 y B2.

Las reglas para el calculo son:

- Si el salario es menor de 20000 los impuestos son 0.
- Si el salario está entre 20001 y 30000 los impuestos son el 10%.
- Si el salario está entre 30001 y 50000 los impuestos son el 20%.
- Si el salario es mayor de 50000 se paga el 38%
- Si se tienen hijos los impuestos se reducen en 180 euros.
- Si se tiene la bonificación B1 los impuestos se reducen en 355 euros.
- Si se tiene la bonificación B2 los impuestos se reducen en 560 euros.
- Se recuerda que las bonificaciones B1 y B2 son compatibles (de hecho hemos dicho que están en checkboxes).
- La cantidad de impuestos puede ser negativa (que significa que le sale «a devolver»)

## HTML calculador

```

<!DOCTYPE html>

<html>
<head>
    <meta charset="utf-8">
    <link type="text/css" href="css/estilo.css"
          rel="stylesheet">
    <title>Plantilla de web</title>
    <script type="text/javascript"
           src="js/jquery.js"></script>
    <script type="text/javascript" src="js/programa.js"></script>
</head>

<body>
<!--Toda nuestra página irá aquí dentro-->
<div id="contenido">

    <h1>Plantilla de proyecto web</h1>
    <form>
        Salario:
        <input type="number" id="salario" min="0" max="1000000" value="0"> <br/>
        <input type="radio" id="con_hijos" name="hijos">Con hijos a cargo <br/>
        <input type="radio" id="sin_hijos" name="hijos" checked="checked">
            Sin hijos a cargo <br/>
        <input type="checkbox" id="bonificacion_b1"> Con derecho a bonif. B1 <br/>
        <input type="checkbox" id="bonificacion_b2"> Con derecho a bonif. B2 <br/>
        <button id="calcular">Calcular impuestos</button>
    </form>
    <div id="mensajes">Aquí aparecerán sus resultados</div>
</body>
</html>

```

## JS Calculador (con JQuery (DAM))

```

function getFloat ( identificador ){
    let objeto_control=$(identificador);
    let cadena_control=objeto_control.val();
    let cantidad=parseFloat(cadena_control);
    return cantidad;
}
function isChecked ( identificador ){
    let objeto_control=$(identificador);
    /*Comprobamos si el control
    tiene la propiedad checked*/
    if (objeto_control.prop("checked")){
        return true;
    }
    /* Si no tiene la propiedad checked
    es que no está marcado*/
    return false;
}
function calcular(){
    let salario_anual=getFloat("#salario");
    let impuestos=0;

    if ( (salario_anual>=20000) && (salario_anual<=30000) ){
        impuestos=salario_anual * 0.1;
    }

    if ( (salario_anual>=30001) && (salario_anual<=50000) ){
        impuestos=salario_anual * 0.2;
    }
    if (salario_anual>=50001){
        impuestos=salario_anual * 0.38;
    }
    if (isChecked("#con_hijos")){
        impuestos = impuestos - 180;
    }
    if (isChecked("#bonificacion_b1")){
        impuestos = impuestos - 245;
    }
    if (isChecked("#bonificacion_b2")){
        impuestos = impuestos - 535;
    }

    let div_mensajes=$("#mensajes");
    div_mensajes.html("Sus <b>impuestos</b> son:"+impuestos);
    return false;
}
function main(){
    let boton=$("#calcular");
    boton.click ( calcular );
}

let objeto_documento=$(document);
objeto_documento.ready(main);

```

## JS Calculador (sin JQuery (ASIR))

```

function limpiar_erroros() {
    let objeto_diesel=document.getElementById("diesel");
    let objeto_pot2300=document.getElementById("pot2300");
    if ((objeto_diesel.checked) && (objeto_pot2300.checked)) {
        alert("Incompatibilidad diesel+2300");
        objeto_pot2300.checked=false;
    }
}

```

```

let objeto_pintura_normal=document.getElementById("normal");
let objeto_color_gris =document.getElementById("gris");
let objeto_color_azul =document.getElementById("azul");
let objeto_color_verde=document.getElementById("verde");

if ( (objeto_pintura_normal.checked)
&& (objeto_color_gris.checked) ){
    alert("Incompatibilidad normal+gris");
    objeto_color_gris.checked=false;
}
if ( (objeto_pintura_normal.checked)
&& (objeto_color_azul.checked) ){
    alert("Incompatibilidad normal+azul");
    objeto_color_azul.checked=false;
}
if ( (objeto_pintura_normal.checked)
&& (objeto_color_verde.checked) ){
    alert("Incompatibilidad normal+verde");
    objeto_color_verde.checked=false;
}

}

function calcular() {

    limpiar_errores();

    let precio=0;

    let objeto_gasolina=document.getElementById("gasolina");
    if (objeto_gasolina.checked) {
        precio=precio + 7000;
    }
    let objeto_diesel=document.getElementById("diesel");
    if (objeto_diesel.checked) {
        precio=precio + 8200;
    }

    let objeto_pot1100=document.getElementById("pot1100");
    if (objeto_pot1100.checked) {
        precio=precio + 800;
    }
    let objeto_pot1800=document.getElementById("pot1800");
    if (objeto_pot1800.checked) {
        precio=precio + 1900;
    }
    let objeto_pot2300=document.getElementById("pot2300");
    if (objeto_pot2300.checked) {
        precio=precio + 2500;
    }

    //Mostramos el precio final
    let objeto_mensajes=document.getElementById("mensajes");
    objeto_mensajes.innerHTML="Precio <u>final</u>"+precio;
}

```

## Comparador de telefonía

Se desea crear una aplicación que permita al usuario saber qué compañía de telefonía le conviene más partiendo de los siguientes datos

- La empresa A ofrece una tarifa que cuesta 20 euros al mes con los 1000 primeros minutos gratis y después cada minuto cuesta 8 céntimos.
- La empresa B ofrece una tarifa que cuesta 10 euros al mes con los 500 primeros minutos gratis y después cada llamada cuesta 12 céntimos.

Hacer una aplicación que permita al usuario introducir la cantidad de minutos que llamará para tres meses distintos que llamaremos «Mes 1», «Mes 2» y «Mes 3» y que le diga qué compañía le interesa más.

## HTML del comparador

```
<form>
    Mes 1<input type="number" value="800"
                min="0" max="5000" id="mes1"> <br/>
    Mes 2<input type="number" value="2000"
                min="0" max="5000" id="mes2"> <br/>
    Mes 3<input type="number" value="600"
                min="0" max="5000" id="mes3"> <br/>
    <input type="submit" value="¿Qué me conviene?"
                onclick="calcular();return false">
</form>
```

## JS del comparador

```
function extraer_numero(id){
    let control;
    control=document.getElementById(id);
    let valor=control.value;
    let numero=parseInt(valor);
    return numero;
}

function precio_mes_empresa_a(minutos){
    //Primero vemos si se pasa
    if (minutos<1000){
        //Si no se pasa, el precio es 20
        return 20;
    }
    //Si llegamos a este punto, es porque se pasa

    //Calculamos cuantos minutos se pasa
    let exceso=minutos-1000;
    //Y ese exceso es a 8 centimos/minutos
    let precio_exceso=exceso*0.08;
    //El precio de ese mes es
    //20 euros más lo que cueste el exceso
    let precio_mes=20+precio_exceso;
    //Devolvemos el calculo a quien lo pidiese
    return precio_mes;
}

function precio_mes_empresa_b(minutos){
    //Primero vemos si se pasa
    if (minutos<500){
        //Si no se pasa, el precio es 10
        return 10;
    }
    //Si llegamos a este punto, es porque se pasa

    //Calculamos cuantos minutos se pasa
    let exceso=minutos-500;
    //Y ese exceso es a 12 centimos/minutos
    let precio_exceso=exceso*0.12;
    //El precio de ese mes es
    //10 euros más lo que cueste el exceso
    let precio_mes=10+precio_exceso;
    //Devolvemos el calculo a quien lo pidiese
    return precio_mes;
}

function calcular(){}
```

```

let minutos_mes1;
minutos_mes1=extraer_numero("mes1");
let minutos_mes2;
minutos_mes2=extraer_numero("mes2");
let minutos_mes3;
minutos_mes3=extraer_numero("mes3");

let coste_mes1_empresa_a;
coste_mes1_empresa_a=precio_mes_empresa_a(
    minutos_mes1);

coste_mes2_empresa_a=precio_mes_empresa_a(
    minutos_mes2);
coste_mes3_empresa_a=precio_mes_empresa_a(
    minutos_mes3);

//Calculamos el precio que costaría
//esos minutos en la empresa A
let coste_total_a;
coste_total_a=coste_mes1_empresa_a+coste_mes2_empresa_a+coste_mes3_empresa_a;

//Repetimos el proceso para la empresa B
let coste_mes1_empresa_b;
coste_mes1_empresa_b=precio_mes_empresa_b(
    minutos_mes1);
coste_mes2_empresa_b=precio_mes_empresa_b(
    minutos_mes2);
coste_mes3_empresa_b=precio_mes_empresa_b(
    minutos_mes3);
//Calculamos ahora el precio que costaría
//esos minutos en la empresa B
let coste_total_b;
coste_total_b=coste_mes1_empresa_b+coste_mes2_empresa_b+coste_mes3_empresa_b;
let zonaresultados;
zonaresultados=document.getElementById("zonaresultados");
let texto;
texto="Con la A el precio es:" + coste_total_a;
texto+=<br/>";
texto+= "Con la B el precio es:" + coste_total_b;
zonaresultados.innerHTML=texto
}

```

## Calculadora en Javascript

Se desea crear una calculadora en Javascript que tenga el siguiente interfaz

Num 1	<input type="text" value="12"/>
Num 2	<input type="text" value="23"/>
<input type="radio"/> Sumar <input type="radio"/> Restar <input type="radio"/> Multiplicación <input type="radio"/> División	

Ejemplo de interfaz de la calculadora en JS

El usuario puede insertar números usando decimales hasta las centésimas y podrá seleccionar el cálculo que quiere hacer usando los elementos radio . El resultado deberá aparecer en algún elemento div sin utilizar por tanto elementos alert ni distraer la atención del usuario.

## Interfaz HTML para la calculadora

En el fichero siguiente se muestra un ejemplo de interfaz

```
<!DOCTYPE html>
<html>
  <head>
    <title>Calculadora JS</title>
    <meta charset="utf-8">
  </head>
  <body>
    <form>
      Num 1:
      <input type="number" step="0.01"
        id="num1" value="2" onchange="calcular();"> <br/>
      Num 2:
      <input type="number" step="0.01"
        id="num2" value="3" onchange="calcular();"> <br/>
      <input type="radio" name="calculo"
        id="suma" onclick="calcular();">
      Sumar<br/>
      <input type="radio" name="calculo"
        id="resta" onclick="calcular();">
      Restar<br/>
      <input type="radio" name="calculo" id="multiplicacion" onclick="calcular();">
      Multiplicación<br/>
      <input type="radio" name="calculo"
        id="division" onclick="calcular();">
      División<br/>
    </form>
    <div id="resultado"></div>
    <script type="text/javascript" src="calculadora.js">
    </script>
  </body>
</html>
```

## Programa JS para la calculadora

El siguiente fichero muestra una posible solución:

```
/*Esta función se ocupa de sacar un control, sacar el texto que hay dentro (recuérdese que es un string)*/
function sacar_numero(id){
  let control=document.getElementById(id);
  /*Algunos controles tienen un "campo" value en el cual
  está el valor de lo que nos interesa, pero ¡está en forma de texto!*/
  let texto=control.value;
  /* Convirtamos ese texto a número*/
  let numero=parseFloat(texto);
  /*Y devolvemos ese número a quien nos ejecutó*/
  return numero;
}

/*Dado un identificador está función devuelve "true" si ese elemento está marcado y "false" si no*/
function esta_checked(id){
  let control=document.getElementById(id);
  if (control.checked){
    return true;
  } else {
    return false;
  }
}

/*Esta función toma un texto y lo inserta dentro del
div resultado*/
function escribir_resultado(texto){
  let divresultado=document.getElementById("resultado");
  divresultado.innerHTML="Resultado:"+texto;
}
```

```

/*Esta es la función que se ejecuta cuando alguien "cambia" un número o cuando alguien "hace
function calcular(){
    let num1=sacar_numero("num1");
    let num2=sacar_numero("num2");
    let resultado=0;
    if (esta_checked("suma")){
        resultado=num1+num2;
    }
    if (esta_checked("resta")){
        resultado=num1-num2;
    }
    if (esta_checked("multiplicacion")){
        resultado=num1*num2;
    }
    if (esta_checked("division")){
        resultado=num1/num2;
    }
    escribir_resultado(resultado)
}

```

## Póliza de seguros

Se pide crear una aplicación para el cálculo de una póliza de seguros en base a distintas situaciones.

- El precio base es 175 euros.
- El usuario puede tener el carnet menos de 6 años. En ese caso el precio se incrementa en 280 euros.
- El usuario puede tener el carnet 6 años o más. En ese caso la póliza se reduce en 25 euros.
- En una lista desplegable se puede elegir el tipo de vehículo:
  - Familiar: el precio no se modifica.
  - Comercial: el precio se incrementa en 165 euros.
  - Deportivo: el precio se incrementa en 280 euros.
- Por último hay dos checkboxes para marcar distintas situaciones:
  - Si se tienen hijos a cargo el precio se reduce en un 5%.
  - Si se tienen padres a cargo el precio se reduce en un 3%
  - Se pueden marcar ninguna, una de ellas o las dos.

## Interfaz de la póliza de seguros

Debería quedar algo así:

The screenshot shows a user interface for calculating insurance premiums. At the top, there is a horizontal slider labeled 'Antigüedad' (Age) with arrows for adjustment. Below it is a dropdown menu labeled 'Tipo de vehículo' (Vehicle Type) with 'Familiar' selected. Underneath the dropdown is a section labeled 'Situación familiar' (Family Situation) with two checkboxes: 'Con hijos a cargo' (With children) and 'Con padres a cargo' (With parents).

Ejemplo de interfaz de la póliza en JS

```

<!DOCTYPE html>
<html>
  <head>
    <title>Pólizas de seguros</title>
    <meta charset="utf-8">

```

```

</head>
<body>
    <form>
        Antigüedad<br/>
        <input type="radio"
            name="antiguedad" id="menos"
            onclick="calcular();">
        &lt; de 6 <br/>
        <input type="radio"
            name="antiguedad" id="mas"
            onclick="calcular();">
        &gt; de 6 <br/>
        Tipo de vehículo<br/>
        <select id="vehiculo" onclick="calcular();">
            <option value="familiar"> Familiar </option>
            <option value="comercial">Comercial</option>
            <option value="deportivo">Deportivo</option>
        </select>
        <br/>
        Situación familiar <br/>
        <input type="checkbox"
            id="conhijos" onclick="calcular();">
        Con hijos a cargo <br/>
        <input type="checkbox"
            id="copadres" onclick="calcular();">
        Con padres a cargo
    </form>
    <div id="resultado"></div>
    <script type="text/javascript" src="polizaseguros.js">
    </script>
</body>
</html>

```

## JS de la póliza de seguros

```

/*Dado un identificador está función devuelve "true" si ese elemento está marcado y "false"
function esta_checked(id){
    let control=document.getElementById(id);
    if (control.checked){
        return true;
    } else {
        return false;
    }
}
function calcular(){
    let precio=0;

    let divresultado=document.getElementById("resultado");
    divresultado.innerHTML="Precio:"+precio;
}

```

## Matrícula

Un centro de estudios desea ofrecer a sus alumnos la posibilidad de cursar algunas materias de forma flexible.

- El usuario puede cursar las horas semanales que desee. Cada hora de clase cuesta 5 euros al mes.
- Un usuario puede ser antiguo alumno. En ese caso se le hace un descuento del 12% sobre el precio final.
- El usuario puede elegir las materias que desee: en concreto puede elegir de primero las asignaturas de «Matemáticas I» e «Historia» y de segundo «Matemáticas II» y «Física».
- Está permitido matricularse de «Matemáticas I» y de «Matemáticas II» a la vez pero **no está permitido elegir «Matemáticas II» sin elegir «Matemáticas I»**. Si se hace esto **no debe verse el**

**precio, sino alguna clase de mensaje de error**

- El precio final mensual debe aparecer en un div .

## Interfaz de la matrícula

El resultado debe ser similar a esto:

**Generales**

Horas semanales:

Antiguo alumno  
 Nuevo alumno

**Primero**

Matemáticas I  
 Historia

**Segundo**

Matemáticas II  
 Física

Ejemplo de interfaz de la matrícula

```
<!DOCTYPE html>
<html>
  <head>
    <title>Matrículas</title>
    <meta charset="utf-8">
  </head>
  <body>
    <form>
      <h1>Generales</h1>
      Horas semanales:
      <input type="number" id="horas"
        min="1" max="30" value="10"><br/>
      <input type="radio" name="antalumno"
        id="si_antiguo">
      Antiguo alumno <br/>
      <input type="radio" name="antalumno"
        id="no_antiguo">
      Nuevo alumno <br/>

      <h1>Primero</h1>
      <input type="checkbox" id="mat1">
      Matemáticas I <br/>
      <input type="checkbox" id="his">
      Historia <br/>
      <h1>Segundo</h1>
      <input type="checkbox" id="mat2">
      Matemáticas II <br/>
      <input type="checkbox" id="fis">
      Física
      <br/>
      <button onclick="calcular();">
        Calcular precio
      </button>
    </form>
    <div id="resultado"></div>
  </body>
</html>
```

## JS de la matrícula

```
function calcular(){  
}
```