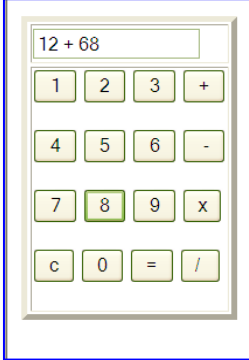




## U.T. 6.- Lenguaje Javascript - DOM. Parte Práctica

Alumn@: \_\_\_\_\_

<p>Ejercicio 1.- <i>Calculadora tabla formulario.</i> Realiza una página implemente una calculadora aritmética. Para ello haz uso de una tabla y un formulario. Indicar que la función que realiza la evaluación de una secuencia de caracteres que representa una expresión es <i>eval()</i>.</p>	
<p>Ejercicio 2.- <i>Fotoslider.</i> Realiza una página que muestre un proyector de fotografías. Tendremos una carpeta con n imágenes, cuyos nombres estarán almacenados en un vector. Necesitaremos una variable global que determine qué imagen es la que se está visualizando. Se definirá 3 funciones: <i>next()</i>, <i>previous()</i> y <i>first()</i></p>	
<p>Ejercicio 3.- <i>Dado.</i> Hacer una página que simule el lanzamiento de un dado. Cuando onclick el button se generará un número aleatorio entre 1 y 6, de forma que si obtengo un 4 aparecerá la imagen dado4.jpg.</p>	



## Ejercicio 4.- Compra de un coche

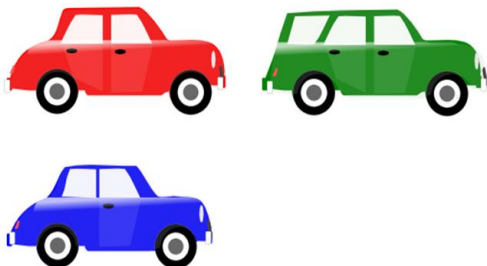
Tienes que desarrollar un formulario para la compra de un coche. Partimos de un precio base de 12.000€, que puede variar según las equipaciones que escoja cada persona:

- Motor: gasolina, diesel (+1500€), híbrido (+3000€)
- Modelo: 3 puertas - solo disponible para gasolina (-2000€), 5 puertas, familiar (+1500€).
- Color: rojo, verde, naranja, blanco (-500€), negro (-600€)
- Pintura metalizada: +750€ disponible solo para rojo, verde y naranja.
- Llantas de aleación: +200€
- Climatizador: +800€
- GPS: +340€
- DVD: +800€ (solo para el modelo familiar)

Desarrollo:

1. Escribe el código Javascript para obtener el precio final.
2. Añade un total, donde se refleja el valor del coche. Además las opciones deberán ser coherentes: si por ejemplo se selecciona el color blanco, el cuadro de pintura metalizada tiene que deshabilitarse.
3. Añade el dibujo del coche, que tiene que ser fiel al modelo y color.

### Imágenes



### Control SELECT:

Este otro objeto visual que podemos disponer en un FORM permite realizar la selección de un string de una lista y tener asociado al mismo un valor no visible. El objetivo fundamental en JavaScript es determinar qué elemento está seleccionado y qué valor tiene asociado. Esto lo hacemos cuando ocurre el evento onChange.

Para determinar la posición del índice seleccionado en la lista:

```
document.getElementById('select1').selectedIndex;
```

Considerando que el objeto SELECT se llama select1 accedemos a la propiedad selectedIndex (almacena la posición del string seleccionado de la lista, numerando a partir de cero).

Para determinar el string seleccionado:

```
document.getElementById('select1').options[document.getElementById('select1').selectedIndex].text;
```

Es decir que el objeto select1 tiene otra propiedad llamada options, a la que accedemos por medio de un subíndice, al string de una determinada posición.

Hay problemas en los que solamente necesitaremos el string almacenado en el objeto SELECT y no el valor asociado (no es obligatorio asociar un valor a cada string).

Y por último con esta expresión accedemos al valor asociado al string:

```
document.getElementById('select1').options[document.getElementById('select1').selectedIndex].value;
```