



LA PROGRAMACIÓN: INICIATE EN UN MUNDO APASIONANTE

## 4. FUNCIONES Y PROGRAMAS

Hasta ahora los bloques de código que hemos escrito los ejecutamos sólo una vez, o varias veces seguidas si estamos en un bucle. En este módulo vamos a ver cómo podemos llamar al código de un bloque en distintos puntos de nuestro programa, sin que ello suponga tener que volver a escribir todas esas líneas. Para conseguir este objetivo tenemos las funciones.

## Funciones

Una función recibe unos parámetros de entrada, ejecuta una serie de instrucciones con esos parámetros y nos da un resultado.

Por ejemplo, dada una mano de cartas podemos construir una función `puntua` que calcule la puntuación obtenida en la mano de cartas. En este caso, recibimos las cartas como parámetro de la función. La lógica de la función itera (recorre) las cartas para calcular la puntuación, que es el resultado que devuelve nuestra función. Encapsulando este código en una función, cada vez que queramos calcular el valor de una mano de cartas, en vez de tener que volver a escribir todo el código anterior, hacemos una llamada a nuestra función `puntua` pasándole la mano a puntuar.

### Ejemplo 1: declaración de una función

```
function puntua (mano){  
    var total = 0;  
    for(var i=0; i<mano.length; i=i+1){  
        total = total + mano[i]  
    }  
    return total;  
}
```

Hemos declarado la función de nombre `puntua`, usando la palabra clave `function` para indicar que estamos declarando una función.



A continuación del nombre van entre paréntesis los parámetros, que pueden ser desde ninguno hasta cuantos queramos. En este caso es sólo uno: lo llamamos `mano` y corresponde a una colección de números con las puntuaciones de una mano de cartas. En el cuerpo de la función, que va entre llaves `{ }`, calculamos el valor total de la mano acumulando el valor de cada número de la colección en una variable `total`. Como puedes ver, dentro del cuerpo de la función accedemos a los parámetros mediante el nombre que le pusimos al comienzo. Para terminar la función devuelve usando la palabra `return` el valor acumulado en `total`. Esto es sólo la declaración de la función, y si lo ejecutamos en la consola no hará nada. Sólo hemos declarado la existencia de esta nueva función, pero para poner a funcionar el código que hemos creado, tendremos que invocar la función.

#### Ejemplo 2: llamada a una función

```
var mano = [10, 9, 8, 7, 8]
var resultado = puntua(mano);
var otra_mano = [1, 3, 5, 7, 8]
var otro_resultado = puntua(otra_mano);
```

Para ejercitar la función tendremos que invocarla. Esto se hace mediante el nombre de la función seguida de los parámetros que le pasamos entre paréntesis. En este caso, le pasamos una colección de números que corresponde a la mano de cartas, y la función calcula el resultado que almacenamos luego en una variable `resultado`. Podemos invocarla de nuevo con otros parámetros, en este caso con otra mano de cartas, y sigue calculando el resultado de forma adecuada.



Las funciones nos sirven además para definir el ámbito (en inglés *scope*) de las variables. Si declaramos una variable dentro de una función, no será accesible desde fuera del cuerpo de esa función. Lo mismo sucede con los parámetros.

### Ejemplo 3: ámbito de las variables

```
var carta = "1c";  
function imprimeCarta(miCarta){  
  var texto = "Mi carta es " + miCarta;  
  console.log(texto);  
}  
imprimeCarta(carta);
```

En este caso, la variable `texto` y el parámetro `miCarta` sólo serán accesibles desde dentro del cuerpo de la función. Si intentamos acceder a ellas desde fuera, obtendremos un error.



## Programas

Un programa se compone de la ejecución de un conjunto de instrucciones de código para desarrollar un trabajo concreto. Para estructurar el código de nuestro programa podemos utilizar funciones, siendo una de ellas la principal que orquesta la utilización de las demás.

### Ejemplo 4: un programa

```
function imprimePuntuacion(puntuacion){
  console.log("La puntuación es " + puntuacion);
}

function programa(){
  var mano = [10, 9, 8, 7, 8]
  var resultado = puntua(mano);
  imprimePuntuacion(resultado);
}

programa();
```

En este programa, reutilizamos la función `puntua` del ejercicio anterior y declaramos una nueva `imprimePuntuacion` que imprime por pantalla la puntuación total. Además, hemos creado una función principal `programa` que utiliza al resto de funciones para completar el trabajo de puntuar una mano de cartas e imprimir su resultado.

