



Programación Imperativa

Trabajando con funciones

Vamos a ejercitar sobre nuestro conocimiento en funciones y poner en práctica todo lo aprendido hasta el momento. Como siempre, recordá que todo el material que hayas visto hasta el momento podés consultarlo. No te asustes si ves algo que no hemos visto todavía, con el correr del tiempo la práctica de buscar conceptos nuevos y cómo utilizarlos va a ser cada vez más sencilla.

¿Qué devuelve cada función?

En este ejercicio deberás pensar qué devuelve cada función sin ejecutarla en la consola.

Micro Desafío 1

```
1  function test1(x, y) {  
2      |    return y - x  
3      |  
4      |  
5      |  
6  }
```

Pista: No importa el orden en que pasemos los argumentos, sino el de los parámetros en el **return**.

Respuesta: La función devuelve el número 30. Si queremos que el resultado se muestre en la consola, deberíamos implementar el

`console.log()`.

Micro Desafío 2

```
1  function test2(x, y) {  
2      return x * 2  
3      console.log(x)  
4      return x / 2  
5  }  
6  
7  
8  test2(10)
```

Pista: Cuando JavaScript encuentra la palabra clave **return**, devuelve el valor de la expresión que tiene a su derecha y termina la ejecución del bloque.

Respuesta: El valor que devuelve la función es 20 ya que utiliza un **return**. Las líneas de código debajo del **return** no se ejecutan.

Funciones simples

1. Crear una función que convierta pulgadas en centímetros. Es decir, recibe por parámetro pulgadas y retorna su equivalente en centímetros.

```
JS ejercicio1_c5s.js X
C5S > JS ejercicio1_c5s.js > ...
1  let pulgadas;
2
3  function pasaraCentimetros(pulgadas){
4  return (pulgadas * 2.54)
5  }
6
7  console.log (pasaraCentimetros(10) + " cm")
8  console.log (pasaraCentimetros(20) + " cm")
9  console.log (pasaraCentimetros(30) + " cm")
10
11

PROBLEMAS  SALIDA  DEBUG CONSOLE  TERMINAL

jaire@LAPTOP-82MLN8CG MINGW64 ~/Downloads/Digital House/Programación imperativa/C5S
$ node ejercicio1_c5s.js
25.4 cm
50.8 cm
76.2 cm

jaire@LAPTOP-82MLN8CG MINGW64 ~/Downloads/Digital House/Programación imperativa/C5S
$
```

2. Crear una función que recibe un string y lo convierte en una URL.
Ejemplo: "pepito" es devuelto como "http://www.pepito.com".

```
JS ejercicio2_c5s.js X
C5S > JS ejercicio2_c5s.js > ...
1  let nombre;
2
3  function web(nombre){
4  |   const extension = "www.";
5  |   const dominio = ".com";
6  return(extension + nombre + dominio)
7  }
8
9  console.log(web("pepito"));
10

PROBLEMAS  SALIDA  DEBUG CONSOLE  TERMINAL

$ node ejercicio2_c5s.js
www.pepito.com
```

3. Crear una función que recibe un string y devuelve la misma frase, pero con



admiración.

4. Crear una función que calcule la edad de los perros, considerando que 1 año para nosotros son 7 de ellos.

```
function edadPerros(edad){  
  return "Tu perro tiene " + edad * 7 + " años";  
}
```

5. Crear una función que calcule el valor de tu hora de trabajo, introduciendo tu sueldo mensual como parámetro. Considera que tu mes de trabajo tiene 40 horas.

```
function valorHora(sueldo) {  
  return sueldo / 40;  
}  
  
const sueldo = 4000
```

6. Crear la función calculadorIMC() que reciba la altura en metros y el peso en kilogramos y calcule el IMC de una persona. Luego, ejecutar la función probando diferentes valores.

```
C5S > JS ejercicio6_c5s.js > ...  
1  let altura;  
2  let peso;  
3  
4  function IMC(peso, altura){  
5    const alturaAlCuadrado = (altura * altura)  
6    return (peso / alturaAlCuadrado)  
7  }  
8  
9  console.log(IMC(70, 1.80));  
10 console.log(IMC(100, 2));  
11 console.log(IMC(50, 1.25));
```

PROBLEMAS SALIDA DEBUG CONSOLE TERMINAL

```
$ node ejercicio6_c5s.js  
21.604938271604937  
25  
32
```

Si llegamos hasta este punto estamos más que satisfechos. Ahora te dejamos unos ejercicios con una dificultad extra. Vas a tener que buscar algunos conceptos por tu cuenta para poder resolver estos ejercicios. Esta es una práctica que los programadores realizamos todos los días. Como siempre te decimos, una parte importante del aprendizaje en programación es ignorar la complejidad, e ir estrictamente a lo que necesitamos. Sabemos que no es una práctica fácil de realizar, pero con el tiempo va a ir haciéndose más fácil y divertida.

1. Crear una función que recibe un string en minúscula, lo convierta a mayúsculas y lo retorne. Pista: investigá qué hace el método de strings: **toUpperCase()**
2. Crear una función que recibe un parámetro y devuelve qué tipo de dato es ese parámetro. Pista: te servirá revisar qué hace la palabra reservada **typeof**.
3. Crear una función a la que le pasamos el radio de un círculo y nos devuelve la circunferencia. Pista: investigá si el objeto *Math* tiene entre sus propiedades el número pi.