Certified Tech Developer
The Ultimate Degree

Arrow Function y Callbacks

Estos conceptos son tediosos pero pueden tomar un papel importante durante nuestra carrera como programador, para ello realizaremos unos ejercicios para así poder sentar buenas bases sobre los mismos.

Arrow function.

Tal vez los siguientes enunciados resulten familiares, solo que esta vez la realizaremos utilizando **arrow function** en vez de la forma tradicional

1. Crear una función que convierta pulgadas en centímetros.
   Recibe por parámetro pulgadas y retorna su equivalente en centímetros.
2. Crear una función que recibe un string y lo convierte en una una URL.
   ej: “pepito” es devuelto como “http://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.
5. Crear una función que calcule el valor de tu hora de trabajo, introduciendo tu sueldo mensual como parámetro.
   PD: considera que tu mes de trabajo tiene 40 horas.

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

Callbacks

En pocas palabras es una función que es utilizada como parámetro en otra función, para comprender mejor realizamos los siguientes ejercicios:

- A partir del siguiente array de nombres, crear una función que reciba un parámetro string, para realizar la búsqueda de elementos que coincidan con dicho parámetro, y retorna un mensaje en caso de encontrar .

```
//Array de nombres

const nombres = ['Martin','Homero','Cosme','Steven','Adam'];

function buscarNombre(nombre){

      //Escriba aqui su codigo

}

//Ejemplo de invocacion

buscarNombre('Martin'); //El nombre Martin fue encontrado
```

- Para manejar el error en caso de no encontrar coincidencias crearemos una función llamada **mostrarResultado** que reciba un parámetro, la misma deberá consultar si el parámetro es un string vacío mostrar un mensaje de “Nombre no encontrado”, en caso contrario mostrar el mensaje “El nombre fue encontrado”.
- Ahora editaremos la función **buscarNombre** para que la misma reciba un *callback* como parámetro y sea invocada una vez recorrido todo el array de nombre, enviando como parámetro un string vacío en caso de no encontrar coincidencias, caso contrario enviar el elemento coincidente (función desarrollada en el punto anterior)

```
function buscarNombre(nombre, callback){

    //Su código

}

function mostrarResultado(res){

     //Su código

}

//Ejemplo de invocación con callback como argumento

buscarNombre('Martin',mostrarResultado)
```