

Funciones

Objetivo

Vamos a ejercitar lo visto sobre **funciones**, poniendo también en práctica temas de clases anteriores.

Recuerda que puedes consultar el material para ayudarte a realizar estos ejercicios. Además, recomendamos que amplíes los conceptos vistos en los sitios oficiales de las tecnologías que estamos aprendiendo: como la de [MDN Web Docs](https://developer.mozilla.org/es/docs/), [W3School](https://www.w3schools.com/) o foros reconocidos como [Stack Overflow](https://stackoverflow.com/).

Micro desafío 1:

1. En Visual Studio Code crear la carpeta de trabajo y dentro de ella inicializar un **repositorio local** y vincularlo con un **repositorio remoto** cuyo nombre debe tener la siguiente nomenclatura: **nombreApellido_tp12**
1. Crear un archivo llamado **rentalCar.js**
2. Crea una función llamada **rentalCar()** que permita calcular el monto a pagar por el **alquiler de un vehículo**, que reciba como parámetros: “*tipo de vehículo*”, “*Días de alquiler*” y “*silla para bebe*”.
3. Define y realiza los cálculos para *obtener el total a pagar*, teniendo en cuenta las siguientes consideraciones:
 - a. De acuerdo al tipo de vehículo, se le cobra al cliente un *monto por día*:
 - i. Compacto: \$14000
 - ii. Mediano: \$17000

- iii. Camioneta: \$28000
 - b. Si adicionalmente el cliente requiere la silla para niños, se le cobra un adicional por día de \$1200
4. Una vez creado el programa, ejecutalo. ¿**Recuerdas cómo?**
 5. Para probar las diferentes respuestas invoca a la función, asigna diferentes valores o argumentos. Usando el [console.log\(\)](#), muestra los resultados al usuario:

“Estimado cliente: en base al tipo de vehículo compacto alquilado, considerando los 3 días utilizados, el monto total a pagar es de \$42000 ” .

(Recuerda que si el cliente pidió la silla para niños debe también indicarlo en el mensaje final)

Luego de hacer las pruebas **todos los console.log() deben quedar comentados.**

Micro desafío 2 (opcional NO ENTREGABLE):

1. En la carpeta ya creada, crea un el archivo **fastFood.js**
2. El teach leader de la empresa nos pide construir un sistema para un local de **venta de comida rápida**. Los clientes eligen el tipo de hamburguesa base con un precio base y, por cada selección que el cliente agregue, se le suma el valor de su selección al precio.

Por ejemplo, si partiendo de una *hamburguesa base carne a la parrilla* con un precio **“a”**, el cliente les incorpora los adicionales: de queso **“b”** y *tomate* **“c”**, el *total a pagar* sería **a+b+c**. Así sucesivamente con el resto de los ingredientes.

Es importante tener presente que cada uno de estos adicionales estarán representados por valores booleanos, es decir, **true** o **false**, dependiendo de si aceptan o no cada uno de los adicionales. Así recibiremos un total de ocho

valores booleanos, uno por cada adicional.

3. Tu responsabilidad será la de **crear dos funciones**:

a. **PRIMERA FUNCIÓN - CALLBACK:** Tendrá la responsabilidad de *calcular el total a pagar por el cliente*, en base a un total de nueve parámetros que recibirá la función:

i. Un string para el **tipo de hamburguesa**, teniendo en cuenta los siguientes valores:

1. Carne a la parrilla: \$1800
2. Pollo: \$1500
3. Vegetariana: \$1200

ii. Y recibirá los ocho valores booleanos de los adicionales, teniendo en cuenta los siguientes valores:

1. Jamón = \$30
2. Queso = \$25
3. Salsa de tomate = \$5
4. Mayonesa = \$5
5. Mostaza = \$5
6. Tomate = \$15
7. Lechuga = \$10
8. Cebolla = \$10

b. **SEGUNDA FUNCIÓN:** Esta función debe llamarse **fastFood()**, la misma tendrá la responsabilidad de **devolver un mensaje al usuario** donde le indique **su nombre, su apellido y el total a pagar en base al tipo de hamburguesa seleccionado más los adicionales** indicados. Tenga en cuenta que esta función deberá recibir como parámetros: *“nombre del usuario”, “apellido del usuario”, “tipo de hamburguesa” y los ocho valores booleanos* en función a los adicionales que el cliente haya seleccionado.

4. La función callback debe retornar el total a pagar y la segunda función el siguiente

mensaje: ***Estimado Daniel Fuentes, el monto total a pagar es de: \$1905,***
donde **1905** es el valor que devuelve la función callback.

5. Para hacer las pruebas pueden hacerlo de la siguiente manera:

```
let mensaje = fastFood("carne",true, false, false, true, true, false, true, false)
console.log(mensaje)
```

6. Luego de hacer las pruebas todos los `console.log()` **deben quedar comentados**.

Desafíos (opcionales NO ENTREGABLES):

Si llegaste hasta acá, significa que vienes trabajando muy bien. **¡Felicitaciones!**

Para que no te quedes con las ganas y puedas seguir practicando, te proponemos que desarrolles las siguientes funciones. Verás que presentan una dificultad extra, por lo que tendrás que buscar algunos conceptos para poder resolverlos. Es recomendable que crees este hábito ya que es algo muy común que los programadores realizamos todos los días.

En la carpeta ya creada, crea un nuevo archivo llamado **extra.js**. Dentro del mismo;

1. Crear la **función upperConverter()** que recibe un string en minúscula, lo convierte en mayúsculas y lo retorna. Investiga qué hace el método de strings **.toUpperCase()**
2. Crear la **función dataType()** que recibe un parámetro y devuelve qué tipo de dato es ese parámetro. Investiga qué hace la palabra reservada **typeof()**.
3. Crear la **función dogAge()** que calcule la edad de los perros, considerando que 1 año para nosotros son 7 de ellos.
4. Crear la **función hourPrice()** que calcule el valor de una hora de trabajo, teniendo en cuenta que dicha función recibe tres parámetros: *El salario mensual, la cantidad de días trabajados y la cantidad de horas que regularmente trabajada por día.*
5. Pueden hacer las pruebas ejecutando las funciones dentro de un `console.log()`,

por ejemplo: `console.log(dogAge(4))` Luego **todos los `console.log()` deben quedar comentados.**

Al final del archivo debes exportar un objeto literal con todas las funciones que hayas creado. En el caso que las hayas creado todas debería quedar así:

```
module.exports = {upperConverter, dataType, dogAge, hourPrice}
```

Entrega

Por medio de este [formulario](#) hacer entrega de la URL del repositorio.

Cuando tengas alguna duda que te impida avanzar, puedes preguntarle a tus tutores.

¡ Éxitos !