

ACÀMICA

# Agenda

---

Daily

Programo: Callback

Buenas prácticas

Break

Programamos: Armar estructura

Programan: Mi artista favorito

Cierre



# Callbacks

Los callbacks son funciones que llaman a otras funciones. Hoy veremos cómo utilizarlas y la importancia de trabajar con ellas.

# Daily



Daily



## Sincronizando...

### Bitácora



¿Cómo te ha ido?  
¿Obstáculos?  
¿Cómo seguimos?

### Challenge



¿Cómo te ha ido?  
¿Obstáculos?  
¿Cómo seguimos?

# Asincronía

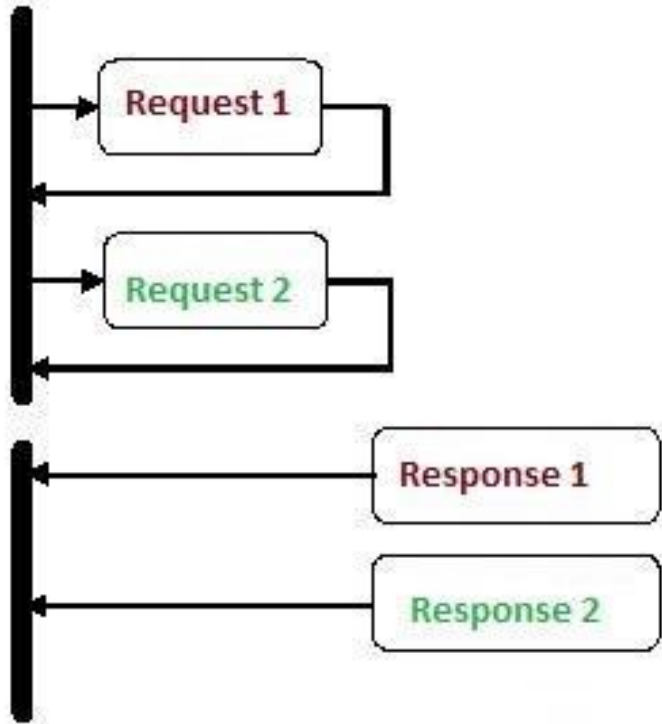


# Asincronía

Las instrucciones que ejecutamos con JavaScript por default son **asíncronas**, es decir, que que manda instrucciones una tras otra sin esperar a que concluya con un resultado.



## Asynchronous



Si pudiéramos ver cómo trabaja cada una de las instrucciones en una línea de tiempo se vería así.

Request 1 es ejecutado y vuelve al flujo principal; lo mismo con Request 2.

A medida que van llegando las respuestas, ingresan al flujo principal y quedan disponibles para el objetivo con el que habían sido programadas.



# Callback



# Callback

En Javascript, las funciones son objetos. Por esta razón, las funciones admiten funciones como argumentos y también son capaces de devolver otras funciones.

Entonces se debe denominar callback a la función que entra como parámetro de otra.



## ¿Por qué necesitamos Callbacks?

En JavaScript los callbacks son inmensamente necesarios para poder procesar la respuesta a una instrucción.



# Programo

mentores/as



# Callbacks

Veamos una demostración de cómo trabajar con callbacks.



# Buenas prácticas



## Validación

Valida mediante *typeof* que sea estrictamente una función antes de ejecutarla.

```
saludar(callback) {  
    if(callback === "function") {  
        callback();  
    }  
}
```



A close-up photograph of a white ceramic cup filled with a latte. The surface of the milk is decorated with intricate latte art, featuring a central heart shape surrounded by concentric, wavy lines. The cup is placed on a matching white saucer. In the background, a white napkin and a silver fork are visible, though they are out of focus. The overall lighting is soft and even.

**¡BREAK!**

---





# Programamos

todos/as



Javascript

# Programamos

Programemos la siguiente estructura:

Hemos realizado las acciones de sumar y restar mediante un callback.  
Es momento de añadir las funciones de multiplicar y dividir.  
Luego, añade el promedio de ambos utilizando una función inline.

# Programan

estudiantes



## **Continuamos con el E-commerce**

¡Prepárate para agregar algunas funcionalidades al MVP del encuentro pasado!

## Agregar / eliminar producto

Cuando un usuario añade o quita un producto a su carrito, muestra un mensaje personalizado informando el éxito de la acción realizada.

Utiliza un callback para mostrar los mensajes correspondientes.



## Ordena tu e-commerce

Agrega a tu listado de productos que los usuarios puedan ordenar ascendente / descendente por nombre o precio.

Utiliza un callback para ordenar tu array y mostrar los resultados como el usuario lo solicita.



# Retardo

Crea dos funciones

- **function one** imprime 1
- **function two** imprime 2

A la function one créale un retardo de 1 segundo con `setTimeout` para mostrar el mensaje.

En la function 2 imprime el mensaje sin ningún retardo.

En el hilo principal de tu aplicación ejecuta primero one y luego two.

Si todo es correcto, primero debería mostrarse el número 2.



# Para la próxima

---

- 1) Termina el ejercicio del encuentro de hoy.
- 2) Lee la bitácora 29 y carga las dudas que tengas al Trello.
- 3) Resuelve el challenge.

En el encuentro que viene uno/a de ustedes será seleccionado para mostrar el ejercicio de hoy y otro/a mostrará cómo resolvió el challenge de la bitácora. De esta manera, ¡aprendemos todos/as de (y con) todos/as, así que vengan preparados/as!



ACÀMICA