

Exercici: Funcions callback en JS

Primera part: Llegeix el document annex a aquesta activitat i fes un esquema representant el contingut del mateix.

Segona part: Assegura't d'haver fet la primera part de l'exercici abans de començar amb aquesta. Fes una pàgina que contingui un mini-programa JavaScript. Aquest programa ha de constar de tres funcions:

- La funció `primera()`
- La funció `segona()`
- La funció `tercera()`
- La funció `cuarta()`

D'aquest programa en farem tres versions que cridarem clicant sobre tres botons:

- **Botó 1:** La versió clàssica()
 - La versió clàssica farà una execució seqüencial de les funcions.
 - El programa anirà informant a la consola d'on està i que està fent. (per escriure a la consola feu servir la comanda: `console.log("text a mostrar per consola")`)
 - Quan executem hauriem d'obtenir alguna cosa semblant a:

```
primera() {
    (...)
    segona();
```

```
tercera();
```

```
(...)
```

```
segona() {
    (...)
    BucleLlarg()
    (...)
}
```

```
tercera() {
    (...)
    BucleLlarg()
    (...)
}
```

```
} |
```

LOG:

```
Empieza la ejecución de primera()
Llamo a segunda() desde principal()
  Empieza la ejecución de segunda()
  Entro al bucle llarg()
  Surto del bucle llarg()
  Acaba la ejecución de segunda()
Acabo de llamar a segunda desde principal()
Llamo a tercera() desde principal()
  Empieza la ejecución de tercera()
  Entro al bucle llarg()
  Surto del bucle llarg()
  Acaba la ejecución de tercera()
Acabo de llamar a tercera desde principal()
Acaba la ejecución de primera()
```

- **Botó 2: La versió amb Callback(funcioAexecutar)**
- La versió amb Callback(funcioAexecutar) serà una funció que recull com a paràmetre una altra funció. De manera aleatòria executarà la funció tercera o la funció segona
- El programa anirà informant a la consola d'on està i que està fent. (per escriure a la consola feu servir la comanda: `console.log("text a mostrar per consola")`)
- Quan executem hauriem d'obtenir alguna cosa semblant a:

```
primera(){
  (...)
  Aleatoriament
```

```
Aleatoriament
```

```
Segona() o tercera(){
  (...)
  BucleLlarg()
  (...)
}
```

```
Tercera() o segona(){
```

```

    (...)
    BucleLlarg()
    (...)
  }

  (...)
}
```

LOG:

```

Empieza la ejecución de primera()
Llamo a segunda() o tercera () desde principal()
  Empieza la ejecución de ?()
  Entro al bucle llarg()
  Surto del bucle llarg()
  Acaba la ejecución de ?()
Acabo de llamar a ? desde principal()
Acaba la ejecución de primera()
```

■ Botó 3: La versió assíncrona (funcioAexecutar)

Tot i que JS no ens deixa treballar de manera assíncrona real sí que ho podem simular i de vegades ens serà molt pràctic.

- Feu un tercer botó que executi una funció que passarem per callback. El treball d'aquesta funció assíncrona serà llençar un alert i escriure alguna cosa a la consola 5 segons després d'haver acabat l'execució de la funció que la va crear. Es a dir una execució d'aquest estil:

```

primera() {
  (...)
  segona(callback);

  (...)
}

Segona() {
  (...)
  setTimeout(5 segons,
cuarta)
  (...)
}
```

5 Segons

```
cuarta();
```

```
cuarta() {  
    (...)  
    Alert()  
    Console log  
    (...)  
}
```

LOG:

```
Empieza la ejecución de primera()  
Llamo a segunda() desde principal()  
    Empieza la ejecución de segunda()  
    Preparo el llançament de quarta()  
    Acaba la ejecución de segunda()  
Acabo de llamar a segunda desde principal()  
Acaba la ejecución de primera()
```

Pasan 5 segundos y

```
Empieza la ejecución de cuarta()  
    Se muestra el alert  
Acaba la ejecución de cuarta()
```

Pista: Us faran falta les funcions `setTimeout()` i `alert()`

Full de ruta:

- Llegeix el document adjunt a aquest exercici que trobaràs a l'aula virtual (aquest mateix .pdf).
- Lliura una pàgina web amb la solució que has programat.
- La pàgina web ha d'incloure un apartat de conclusions i reflexions.
 - **Idees:** Com ha anat? S'ha comportat el programa com esperàves? Sí/No? Reflexiona sobre quins avantatges o aplicació pràctica podem trobar per les funcions amb callback.
 - **Proves opcionals:** Puc cridar a dues funcions callback dins d'una altra funció? Puc cridar a una funció callback amb paràmetres?