Taula de continguts:

[Introducció a JS](#h.51l5az7r2uyr)

[Activitat 1.4.2: Funcions callback en JS](#h.ctasximulhit)

[Conclusions](#h.vmxrp2frz9o8)

# Introducció a JS

## Activitat 1.4.2: Ús de funcions callback en JS

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 principal()
* 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 classica()
  + 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 sevir la comanda: console.log(”text a mostrar per consola”)
  + Quan executem hauriem d'obtenir alguna cosa semblant a:

LOG:

Empieza la ejecución de classica()

Llamo a segunda()desde classica()

Empieza la ejecución de segunda()

Entro al bucle llarg()

Surto del bucle llarg()

Acaba la ejecución de segunda()

Acabo de llamar a segunda desde classica()

Llamo a tercera()desde classica()

Empieza la ejecución de tercera()

Entro al bucle llarg()

Surto del bucle llarg()

Acaba la ejecución de tercera()

Acabo de llamar a tercera desde classica()

Acaba la ejecución de classica()

* **Botó 2:** La versió ambCallback(funcioAexecutar)
  + La versió ambCallback(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 sevir la comanda: console.log(”text a mostrar per consola”)
  + Quan executem hauríem d'obtenir alguna cosa semblant a:

LOG:

Empieza la ejecución de ambCallback()

Llamo a segunda() o tercera () desde primera()

Empieza la ejecución de ?()

Entro al bucle llarg()

Surto del bucle llarg()

Acaba la ejecución de ?()

Acabo de llamar a ? desde primera()

Acaba la ejecución de ambCallback()

* **Botó 3:** La versió assincrona(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 desprès d'haver acabat l'execució de la funció que la va crear. Es a dir una execució d'aquest estil:

LOG:

Empieza la ejecución de assincrona()

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 assincrona()

**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()

## 

## Conclusions

La pàgina web ha d'incloure un apartat de conclusions i reflexions. Aquí teniu algunes preguntes que us podeu fer:

* **Idees:**
* Com ha anat?
* S'ha comportat el programa com esperàves? Sí/No?
* Reflexiona sobre quins avantatges o aplicació pràtica 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?
* Mentre s’executa una funció callback amb molt pes de treball, es poden fer altres coses amb la pàgina web? (pots fer un bucle infinit per simular una funció amb molta càrrega de treball)
* Mentre s’executa una funció assíncrona amb molt pes de treball, es poden fer altres coses amb la pàgina web? (pots fer un bucle infinit per simular una funció amb molta càrrega de treball)
* Mentre s’executa una funció clàssica amb molt pes de treball, es poden fer altres coses amb la pàgina web? (pots fer un bucle infinit per simular una funció amb molta càrrega de treball)