Day 8 - Callback Functions

Problem statement : If we want 4 functions to execute in a flow , we are dependent on the user to call them in sequence.

E.g searchItem > add to cart > Checkout > Payment.

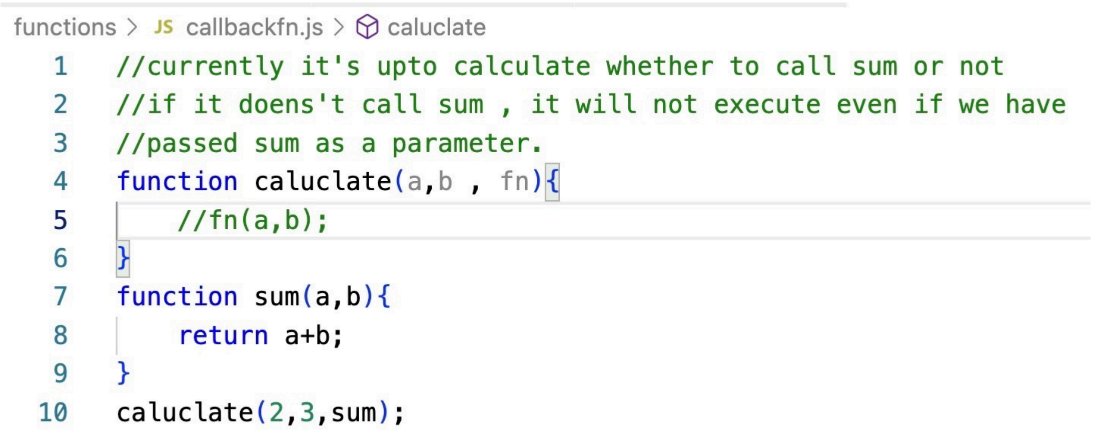
It’s user who is controlling the flow and not the system.
How can we achieve system control , user should click one button and everything should be executed as per flow.

Callback in laymen term

- A called B but B is busy , he said I will callback (as per my will)
- When B is free (but not clearly defined) , he will callback A.

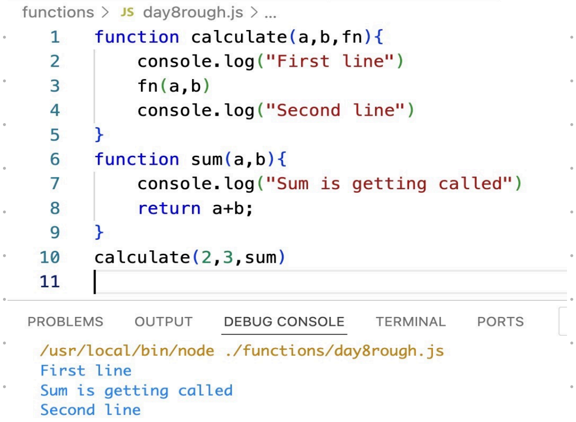
In JavaScript term

- We have already learnt : Passing a function as an argument.
- Below pic , we are leave the invocation of sum function on the mercy of calculate function. so sum is a call back function
- It’s up to calculate , whether to call sum or not
- CALLBACK function
  “ Those function which are passed as a parameter to another function , the fn who is recieving it will call them as per their will.”



```
//currently it's upto calculate whether to call sum or not
//if it doens't call sum , it will not execute even if we have
//passed sum as a parameter.
function caluclate(a,b , fn){
    //fn(a,b);
}
function sum(a,b){
    return a+b;
}
caluclate(2,3,sum);
```

Calculate can decide to call/not call / call multiple times according to their will.
This is actually a good thing , using this we can solve our initial problem of cart.



```
function calculate(a,b,fn){
    console.log("First line")
    fn(a,b)
    console.log("Second line")
}
function sum(a,b){
    console.log("Sum is getting called")
    return a+b;
}
calculate(2,3,sum)
```

```
/usr/local/bin/node ./functions/day8rough.js
First line
Sum is getting called
Second line
```