22-08 - Advanced Javascript Concept - 2

Monday, 22 August 2022 8:22 PM

① Higher order functions :- (HOF)

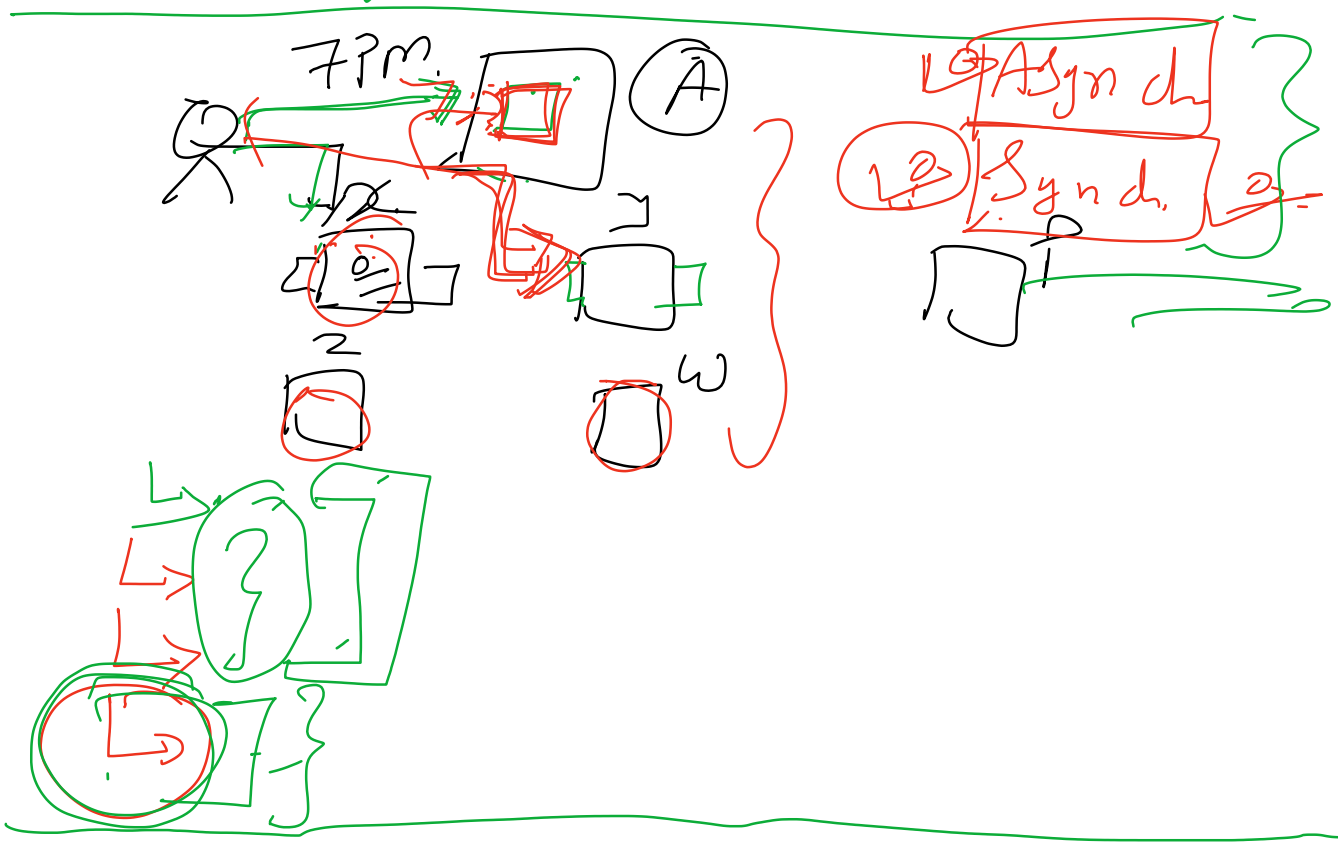
→ HOF are functions that return other functions as inputs as arguments or return functions as well.

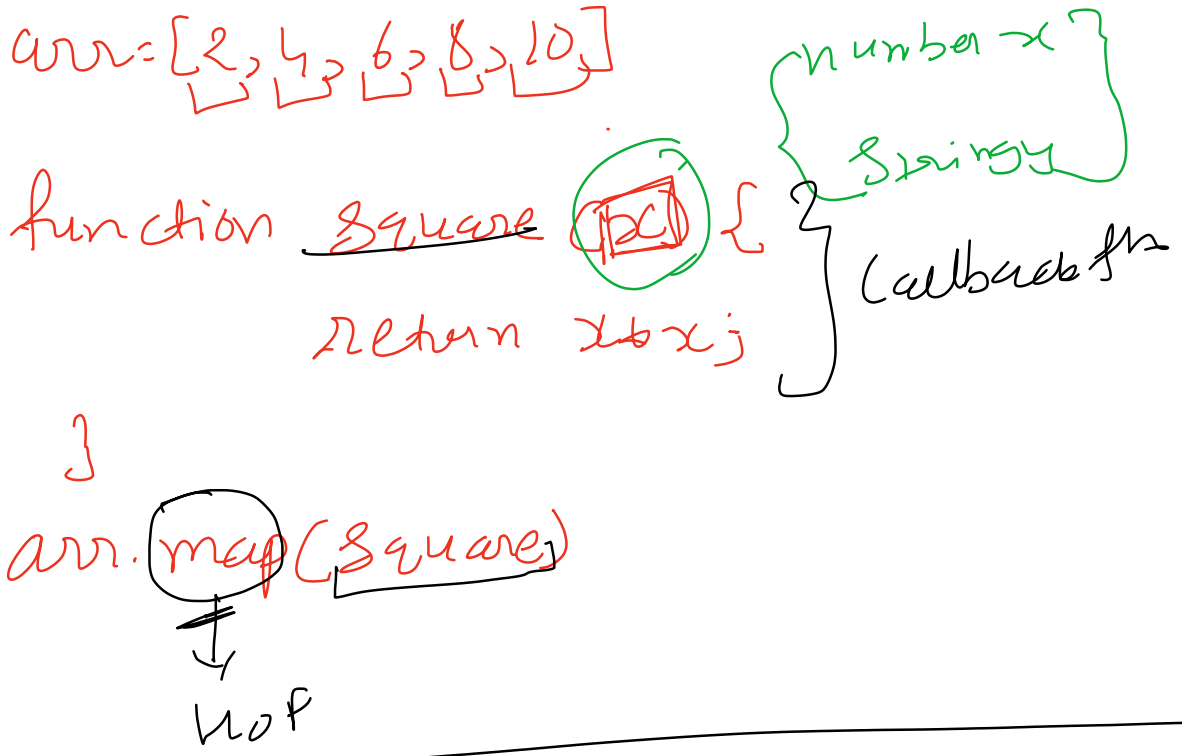
```
const res = arr.filter(function(item){
    return item > 3;
})
console.log(res)

// Map, filter, reduce, sort => they are acc
```

fn 2 → fn which is accepting another fn as an arg. } fn y

fn 2 → HOF }

fn y → callback fn :- }





② Composability :-

-> Process of combining multiple simple fns to build a more complicated one.

Const add = (a,b) => a+b;

Const mult = (a,b) => a*b;

Add(2,mult(3,5)) => 17

2 + 3 *5 +4 *7 *5 *3

Add(2,mult(3,5), mult(4,7),mult((5,3))
Add(2,15,28,15)

2 + 3*5 /7

```
Const add = (a,b)=>a+b;
Const mult = (a,b) => a*b;
Const divide = (a,b) => a/b;

Add(2, divide(mult(3,5),7))
```

Currying :-

Currying is a technique of evaluating function with multiple arguments, into a sequence of functions with single argument.

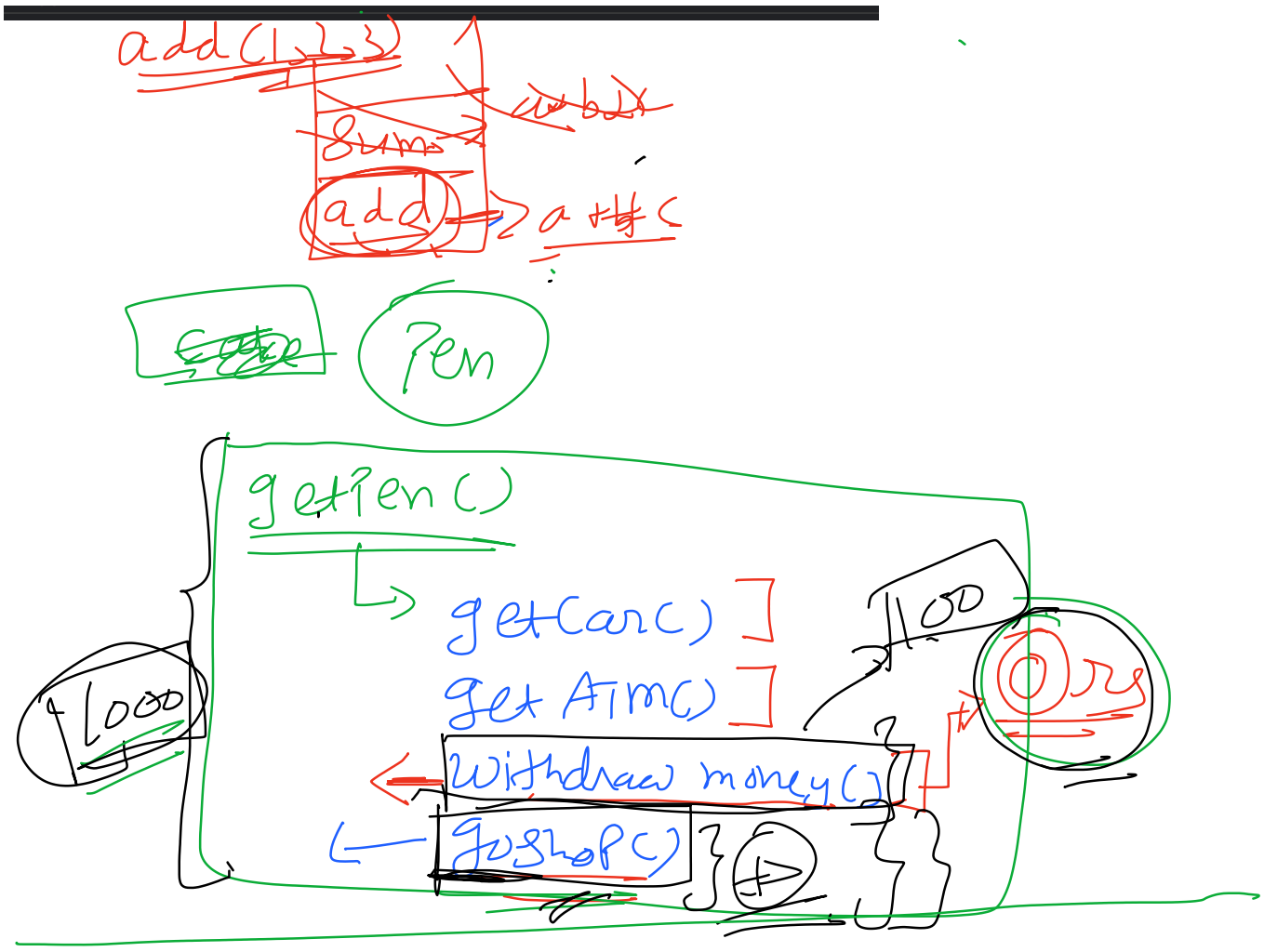
When a function, instead of taking all arguments at once, takes the first one and return a new function that takes a second one and returns a new function which takes third one and so on.

Uses :-

It helps to avoid passing same variable again and again

Ordinary function :-

```
debugger;
const add = (a,b,c) => {
    return a+b+c;
}
//add(1,2,3);

const curryAdd = (add) => {
    add = (a,b,c) => {
        return a*b*c;
    }
    return (a) => {
        return (b) => {
            return (c) => {
                return add(a,b,c);
            }
        }
    }
}
const addition = curryAdd(add);

console.log(addition(2)(2)(3))
// 12 ===== 6 ======= undefined ====== error
```



Anonymous function :- will not have any actual name

Arrow =>

Left => arg
Right => logic

```
(arg) => {
    //logic
}
```