JavaScript for Developers

Callbacks | Asynchronous programming

Callbacks

- ✓ Function is a type
- ✓ It can act like a variable
- ✓ Callbacks
  = callback functions
  = function use as argument to call back later

Callbacks

- ✓ Use function as a parameter

```
const phepCong = (a,b) => (a+b);
const phepTru = (a,b) => (a-b);
const phepNhan = (a,b) => (a*b);
const phepChia = (a,b) => (a/b);

const tinhToan = (phepTinhFunc, a, b) => {
    const kq = phepTinhFunc(a, b);
    return kq;
}
```

Callbacks in action

✓ A function call later | Asynchronous

```
const sayHello = () => {
    console.log("Hello");
}

const sayGoodBye = (name1, name2) => {
    console.log("Good bye " + name1 + ", " + name2);
}

setTimeout(sayGoodBye, 3000, "Cong", "Cuong");

sayHello();
```

Callbacks in action

- ✓ If function only use in one place

```
const sayGoodBye = () => {
    console.log("Good bye!");
}

setTimeout(sayGoodBye, 3000);
```

```
setTimeout(() => {
    console.log("Good bye!");
}, 3000);
```

Callbacks in action

✓ Real world example

```
const fs = require("fs");

console.log("Ready to read file!");
fs.readFile("data.txt", 'utf-8', (err, data) => {
    if(err){
        console.log(err.message);
    }else{
        console.log("Data: " + data);
    }
})
console.log("--- THE END ---");
```

Refs

https://www.w3schools.com/js/js_callback.asp

https://www.w3schools.com/js/js_asynchronous.asp

https://nodejs.dev/learn/reading-files-with-nodejs

JavaScript for Developers

Callbacks | Asynchronous programming

The Brown Box