academeez

Closure

How Javascript looks for the variables

Closure

academeez

- After understanding closure we will understand how Javascript is looking for variables
- We will understand common patterns like Factory and private variables

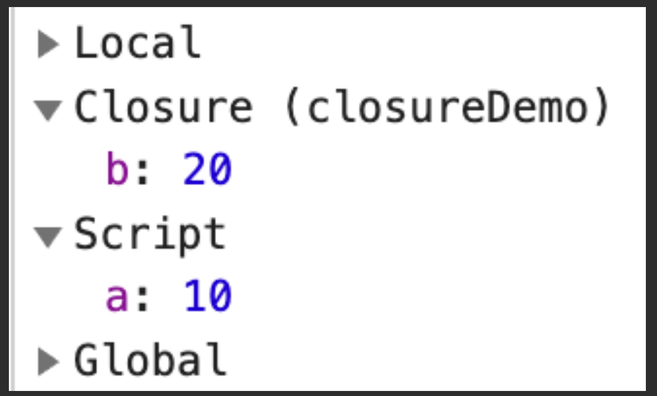
What is closure

academeez

- When you create a function, a closure is created for that function
- Javascript will use the closure to find variables that are outside you function scope
- First we search in the near function than we step out to the external function and then we step out again and again.

```
const a = 10;

function closureDemo() {
    const b = 20;
    return function add() {
        return a + b;
    }
}
```



First example:

academeez

- What will be printed?

```
let a = 10;

setTimeout(() => {
    console.log(a);
    a+=10;
}, 1000)

setTimeout(() => {
    console.log(a);
}, 2000)
```

Common pattern - Factory - Express example

academeez

- Express is used as an example.
- We are using closure to create a middleware function with certain configurations
- The internal function can do a similar action for different configured use cases

```
function greetings(name) {
    return (req, res) => {
        res.send(`Hello ${name}`);
    }
}

app.get('/piglet', greetings('Piglet'));
app.get('/sweetness', greetings('Sweetness'));
```

Closure example - React

- In React we create UI components using functions
- This means that closure will play a large role here and understanding how it works can save you a lot of debugging time
- In this example do you think counter will change or will the value be fixed?

```
function Login() {
    // this function will be called multiple times
    const counter = Math.random();

    useEffect(() => {
        // the function here is called once
        setInterval(() => {
            // this is called every second
            console.log(counter)
        }, 1000)
    }, [])
}
```

Closure pattern - private variables

- We can use closure to create private variables that can only be accessed from inside the object

```
function Person() {
    const privateNumber = 42

    const privateMethod = function() {
        return 10;
    }

    this.hello = function() {
        console.log(privateNumber + privateMethod());
    }
}

const me = new Person()
me.hello()
```

Closure pattern - private variables

academeez

- Private variable will soon be available in the class syntax as well, currently on Stage3

```
class Person {
    #sayHello() {
        console.log('say hello');
    }

    publicHello() {
        this.sayHello();
    }
}

const me = new Person();
me.sayHello();  // error
me.publicHello(); //ok
```

Summary

academeez

- It’s important to understand where Javascript will look for variables in functions
- Understanding this concept will later let you understand different patterns we can use.
- Javascript will look for the variable inside the function than it will start looking at the external functions and than at the global scope

academeez

Thank You

Next Lesson: Class

Create class and instances