

LESSON 7

CLOSURE

[[SCOPE]]

- ▶ Inner
- ▶ Outer
- ▶ Global

LEXICAL ENVIRONMENT

- ▶ Hidden object
 - ▶ Environment Record
 - ▶ Parent scope

EXAMPLE

```
window.global = 0;  
  
let a = 1;  
  
function fn() {  
  let b = 2;  
  console.log(a + b);  
}
```

fn замкнута на a

NESTED FUNCTIONS

```
function sayHi(firstName, lastName) {  
  function getFullName() {  
    return firstName + " " + lastName;  
  }  
  alert("Hello, " + getFullName());  
}
```

CLOSURE

- ▶ Function + scope;
- ▶ return function;

EXAMPLE

```
let count = 0;

function counter() {
  count++;

  return count;
}

console.log(counter()) // 1
console.log(counter()) // 2
console.log(counter()) // 3
console.log(counter()) // 4
```

EXAMPLE

```
function getCounter() {  
  let count = 0;  
  
  return function () {  
    return ++count;  
  };  
}  
  
const counter = getCounter();  
  
console.log(counter()); // 1  
console.log(counter()); // 2  
  
const counter2 = getCounter();  
  
console.log(counter2()); // 1  
console.log(counter2()); // 2
```

LESSON 7

CONTEXT

GLOBAL OBJECT

- ▶ Global context (window)
- ▶ Functional

THIS

- ▶ Returns a context

```
function someFn() {  
  console.log(this); // window  
  console.log(this.date); // undefined  
}
```

```
someFn();
```

```
function someFn() {  
  console.log(this.date);  
}  
  
var date = "28 September 2020";  
  
someFn();
```

```
function someFn() {  
  console.log(this.date); // undefined  
}  
  
let date = "28 September 2020";  
  
someFn();
```

METHODS

```
const user = {  
  name: "Kirk Hammett",  
  sayHello: function () {  
    console.log("Hello, " + this.name);  
  },  
};  
  
user.sayHello() // "Hello, Kirk Hammett"
```

CALL, APPLY

```
const user = {
  name: "Kirk Hammett",
  sayHello: function (title) {
    console.log("Hello, " + title + this.name + "!");
  },
};

const user2 = {
  name: "Lady Gaga",
};

user.sayHello("Mr."); // "Hello, Mr.Kirk Hammett!"
user.sayHello.call(user2, "Mrs."); // "Hello, Mrs.Lady Gaga!"
user.sayHello.apply(user2, ["Mrs."]); // "Hello, Mrs.Lady Gaga!"
```

BIND

```
const user = {
  name: "Kirk Hammett",
  sayHello: function () {
    console.log("Hello, " + this.name + "!");
  },
};

const community = {
  name: "Young Engineer",
};

const greetings = user.sayHello.bind(community);

greetings(); // "Hello, Young Engineer!"
```

CONSTRUCTOR WITH PARAMETERS

```
function Person() {  
  this.name = "Ivan";  
  this.age = 19;  
}  
  
const worker = new Person();  
  
console.log(worker);  
console.log(worker.name);  
console.log(worker.age);  
console.log(worker instanceof Person);  
console.log(worker instanceof Object);
```

CONSTRUCTOR WITH PARAMETERS

```
function Person(name, lName, age) {  
  this.firstName = name;  
  this.lastName = lName;  
  this.age = age;  
}  
  
const student1 = new Person("John", "Smith", 20);  
const student2 = new Person("Kevin", "Smith", 21);  
  
console.log(student1);  
console.log(student2);
```