Basics in JavaScript

Index

Regular Expression in JavaScript

Array in JavaScript

1. forEach, Map, Filter

2. Destructuring Array

3. Rest Operator

Array: Destructuring Array

```
const numbers = [10,20,30];

// Destructuring
const [a,b,c] = numbers;

console.log(a); // 10
console.log(b); // 20
console.log(c); // 30
```

Array: Destructuring Array (2)

```javascript
const numbers = [10,20,30];

// Destructuring
const [first, , third] = numbers;

console.log(first); // 10
console.log(third); // 30
```

Array: Destructuring Array (3)

```
const number = [5];
const [x,y = 99] = number;

console.log(x); // 5
console.log(y); // 99
```

Array: Rest Operator

```
const [first, ...rest] = [10, 20, 30, 40];

console.log(first); // 10
console.log(rest); // [20,30,40]
```

Object in JavaScript

1. Basic feature of an object

2. Creating an Object

3. Destructuring an Object

Class and OOP in JavaScript

1. Class

2. Prototype

3. Constructors

4. Types

5. Subclasses

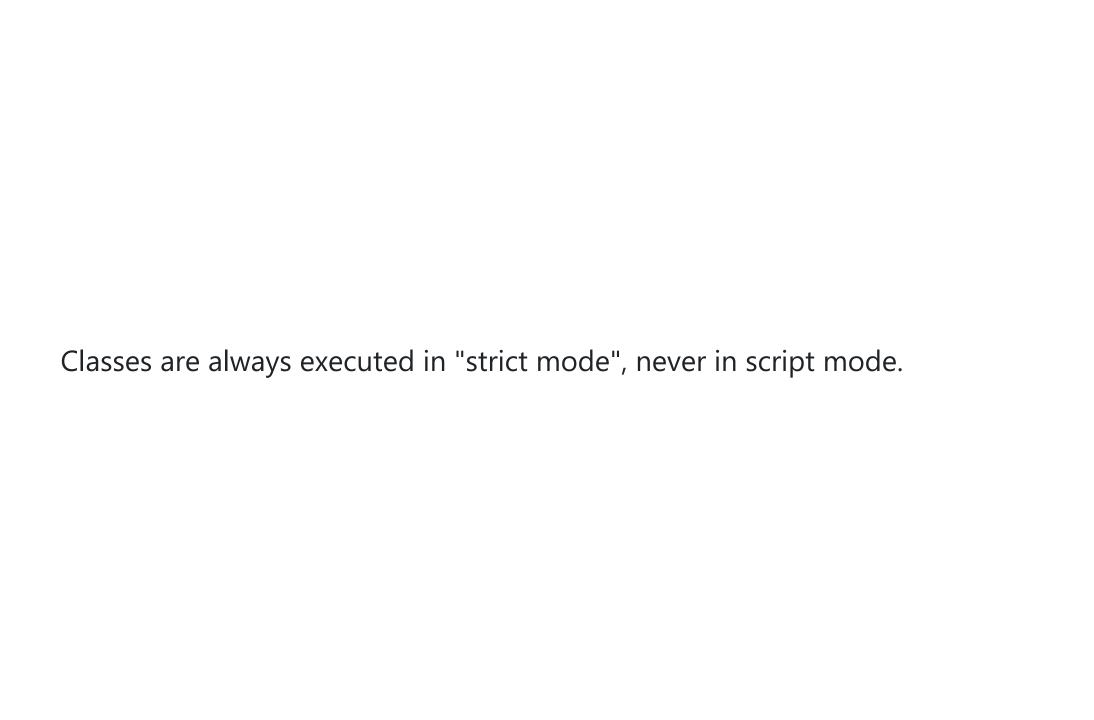6. Modules

Class and OOP: Class

- Class is not an object, it is a template for object
- Constructors work as in Java

Abstract Example of JavaScript Class

```
class ClassName {
    constructor() {...}
    method_1() {...}
    method_2() {...}
    method_3() {...}
}
```

Concrete Example of JavScript Class

```
class Car {
    constructor(name, year) {
        this.name = name;
        this.year = year;
    }

    age(x) {
        return x - this.year;
    }
}

const myCar = new Car("Ford", 2014);
```

Classes are always executed in "strict mode", never in script mode.

Function

1. Recursion

2. Scope

3. Closure

4. Callbacks

5. IIFE

6. Functions are First-Class Citizens

7. High-Order Function

8. Currying

9. Memoization

Asynchronous Programming

Asynchronous Programming: Topics

1. Asynchronous Programming by Events

2. Promise - Then, Catch, Finally

3. Async - Await

Testing

1. Basics

2. Jasmine (Jest, Mocha)

External Libraries

External Libraries: Topics

1. Ajax

2. jQuery

3. Fetch

4. Axios

5. Superagent

6. Prototype

7. Node HTTP