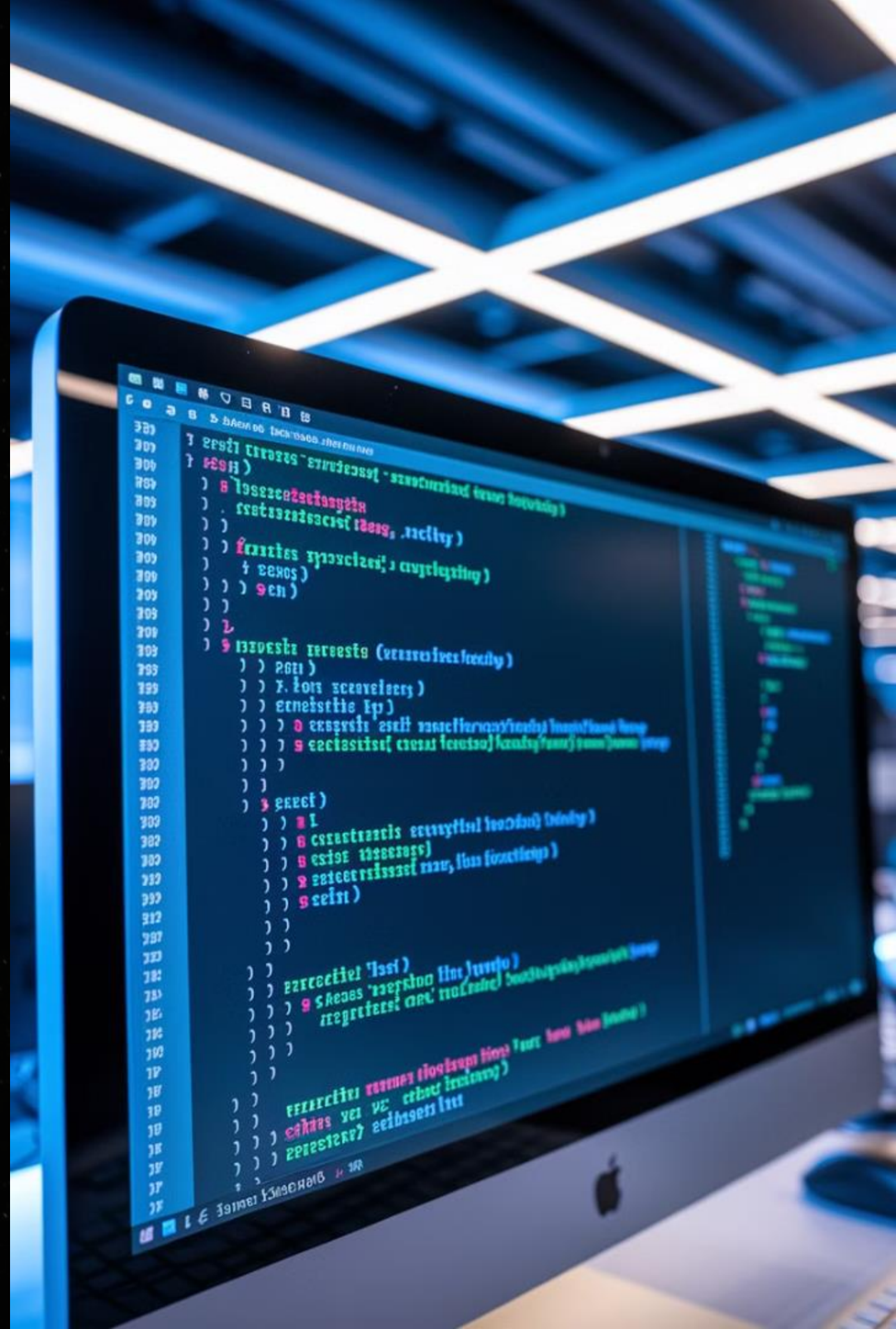
Functions, Arrays, and Objects in JavaScript

Asst. Lect. Ali Al-khawaja



Functions in JavaScript

What are Functions?

Definition: A function is a reusable block of code that performs a specific task.

Benefits of using functions:

- Code reusability
- Better organization
- Easier debugging

Example:

```
function greet() {
    console.log("Hello, World!");
}

greet(); // Calls the function
```

Function Syntax

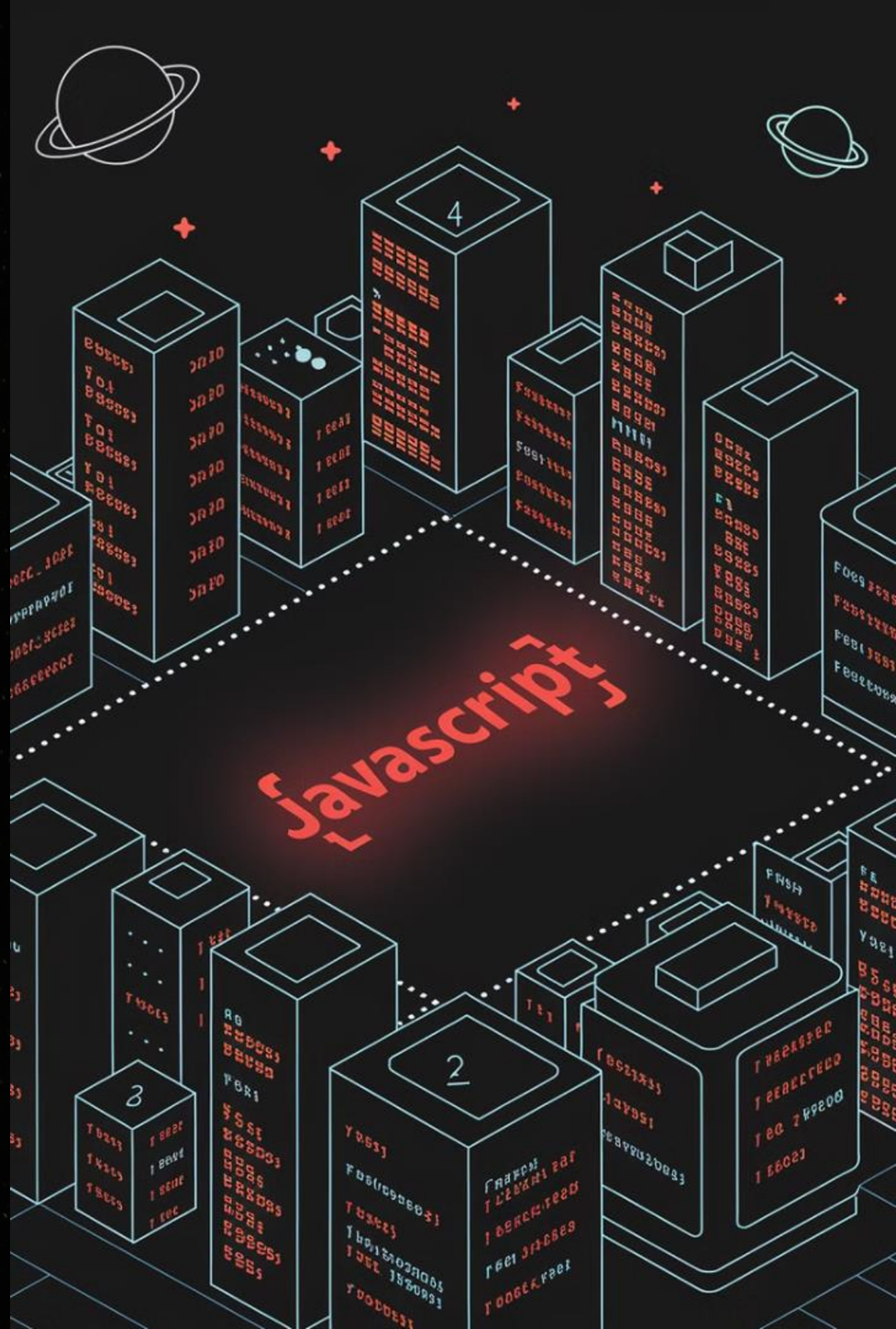
General Syntax:

```
function functionName(parameters) {
    // Code to execute
    return value; // (optional)
}
```

Example:

```
function add(a, b) {
    return a + b;
}

console.log(add(5, 3)); // Output: 8
```



Function with Default Parameters

1. Default parameters allow setting a default value if no argument is provided.

2. Example:

```
function greet(name = "Guest") {
    console.log("Hello, " + name);
}

greet();         // Output: Hello, Guest
greet("Ali");    // Output: Hello, Ali
```

Function Expressions & Arrow Functions

Function expressions store functions in variables.

Arrow functions provide a shorter syntax.

Example of Function Expression:

```
const multiply = function(a, b) {
    return a * b;
};

console.log(multiply(4, 2));
// Output: 8
```

Example of Arrow Function:

```
const divide = (a, b) => a / b;

console.log(divide(10, 2));
// Output: 5
```

Arrays in JavaScript

What are Arrays?

Definition: An array is a collection of multiple values stored in a single variable.

Key Features:

- Stores multiple values
- Indexed (starts from 0)
- Can hold different data types

Example:

```
let fruits = ["Apple",
"Banana", "Cherry"];

console.log(fruits[0]); //
Output: Apple
console.log(fruits.length); //
Output: 3
```

Array Methods

Method	Description	Example
push()	Adds element at the end	arr.push("Mango")
pop()	Removes last element	arr.pop()
shift()	Removes first element	arr.shift()
unshift()	Adds element at the beginning	arr.unshift("Grapes")
splice()	Adds/removes elements	arr.splice(1, 1, "Pear")
slice()	Extracts part of the array	arr.slice(1, 3)
forEach()	Loops through array	arr.forEach(fruit => console.log(fruit))

Example:

```
let numbers = [1, 2, 3, 4, 5];
numbers.push(6);
console.log(numbers); // Output: [1, 2, 3, 4, 5, 6]
numbers.pop();
console.log(numbers); // Output: [1, 2, 3, 4, 5]
```

Looping Through Arrays

Using for loop:

```
let colors = ["Red", "Green", "Blue"];
for (let i = 0; i < colors.length; i++)
{
    console.log(colors[i]);
}
```

Using forEach method:

```
colors.forEach(color => console.log(color));
```

Map, Filter, and Reduce

map():

Creates a new array by modifying elements.

```
let numbers = [1, 2, 3, 4];
let squares = numbers.map(num => num * num);
console.log(squares);

// Output: [1, 4, 9, 16]
```

filter():

Creates a new array with elements that match a condition.

```
let evenNumbers = numbers.filter(num =>
num % 2 === 0);
console.log(evenNumbers);

// Output: [2, 4]
```

reduce():

Reduces the array to a single value.

```
let sum = numbers.reduce((total, num) => total + num, 0);
console.log(sum); // Output: 10
```

Objects in JavaScript

What are Objects?

Definition: Objects store data in key-value pairs.

Objects are used to represent real-world entities.

Example:

```
let person = {
    name: "Ali",
    age: 25,
    city: "Baghdad"
};

console.log(person.name);
// Output: Ali
```

Adding & Modifying Object Properties

1

Adding a new property:

```
person.country = "Iraq";
console.log(person);
```

2

Modifying a property:

```
person.age = 26;
console.log(person);
```

3

Deleting a property:

```
delete person.city;
console.log(person);
```

Object Methods

1. Objects can also contain functions (methods).

2. Example:

```
let car = {
    brand: "Toyota",
    model: "Corolla",
    start: function() {
        console.log("Car is starting...");
    }
};

car.start(); // Output: Car is starting...
```

Looping Through Objects

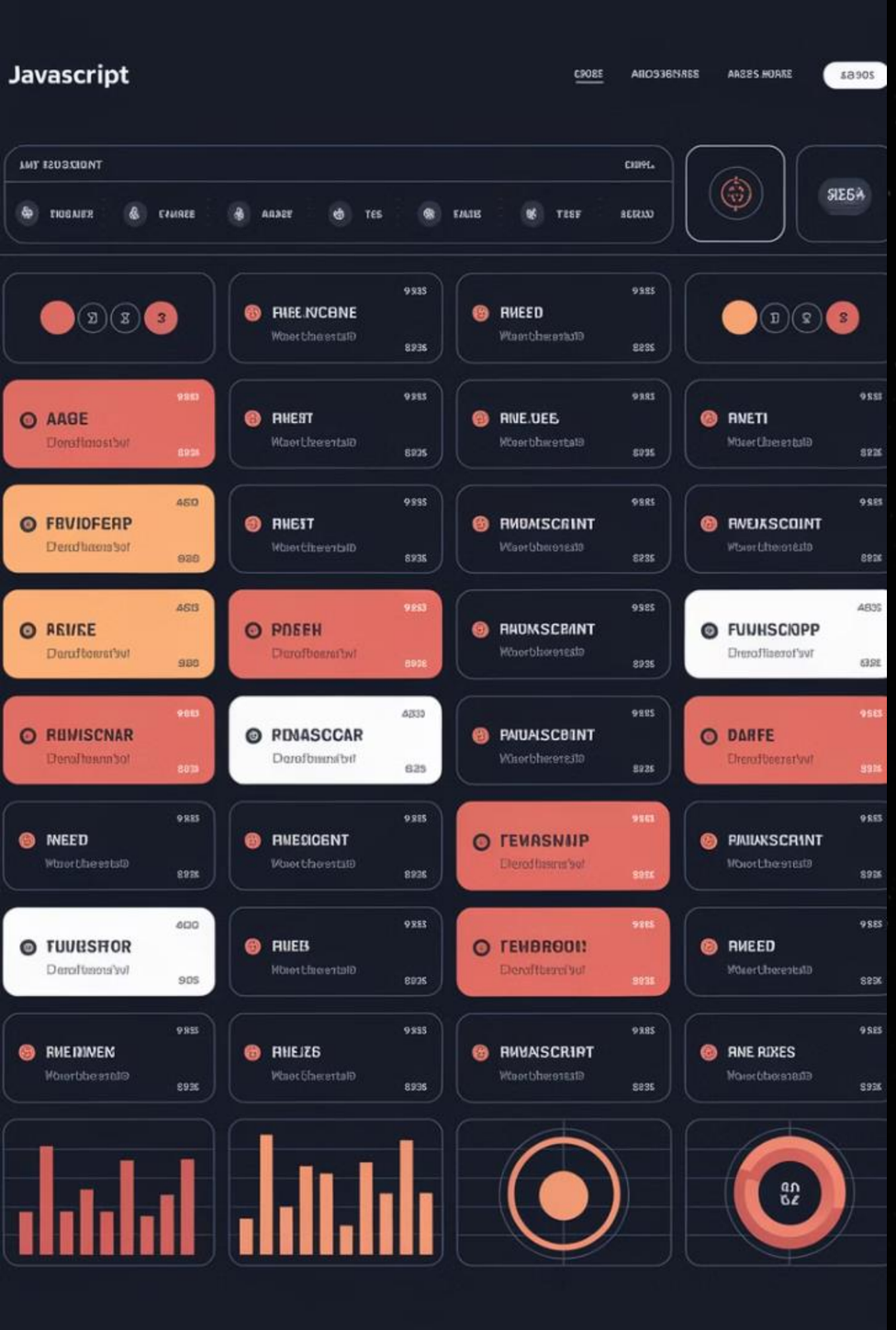
Using for...in loop to iterate through object properties:

```
let user = { name: "Ali", age: 30, country:
"Iraq" };

for (let key in user) {
    console.log(key + ": " + user[key]);
}
```

Combining Arrays, Objects, and Functions

This combination allows for powerful and flexible programming structures in JavaScript.



Array of Objects

Example:

```
let students = [
    { name: "Ali", grade: "A" },
    { name: "Sara", grade: "B" },
    { name: "Omar", grade: "A" }
];

console.log(students[1].name);
// Output: Sara
```

Homework 1

1. Write a function that takes a name as an argument and returns "Hello, Name!".

Homework 2

1. Create an array of numbers and use map() to create a new array of their squares.


Any Questions? Let's Code and Discuss!