## 🔷 JavaScript Functions

### 1️⃣ Function Introduction

**What is a Function?**

A **function** is a block of code that **does a task** and runs **only when called**.

**Why use functions?**

- Reuse code
- Reduce repetition
- Make code clean & readable

---

**Simple Function Example**

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

greet(); // function call
```

📌 **Use:** Prints message when needed.

---

### 2️⃣ Function Parameters

**What are Parameters?**

Parameters are **values passed to a function** to make it flexible.

---

**Example**

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

```
let result = add(5, 3);
console.log(result); // 8
```

**Use:** Same function works for different values.

---

**Function with Default Parameter**

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

greet();       // Hello User
greet("Aman");  // Hello Aman
```

---

### 3️⃣ Function Expression

**What is Function Expression?**

A function **stored in a variable**.

---

**Example**

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

console.log(multiply(4, 5)); // 20
```

**Use:**

- Used in callbacks

- More control over scope

---

## 🔷 Arrow Functions (ES6)

### What is Arrow Function?

Shorter syntax for writing functions.

---

### Normal vs Arrow

```
// Normal
function square(x) {
  return x * x;
}

// Arrow
const square = x => x * x;
```

---

### Arrow Function with Block

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

### Use:

- Cleaner code
- Mostly used in modern JS, React, etc.

---

## 🔷 JavaScript Objects

### 5️⃣ Objects Introduction

**What is an Object?**

An object stores **data in key–value pairs**.

---

**Object Example**

```
let student = {
  name: "Ravi",
  age: 20,
  course: "BCA"
};
```

**Use:** Represents real-world entities.

---

### 6️⃣ Object Properties

**Properties = Variables inside object**

---

**Access Properties**

```
console.log(student.name);
console.log(student["age"]);
```

---

**Modify Property**

```
student.age = 21;
```

---

**Add New Property**

```
student.city = "Delhi";
```

---

## 7️⃣ Object Methods

**Methods = Functions inside object**

---

**Example**

```
let user = {
  name: "Amit",
  greet: function() {
    return "Hello " + this.name;
  }
};

console.log(user.greet());
```

this refers to **current object**.

---

**Object with Arrow Method (not recommended)**

```
let obj = {
  name: "Test",
  say: () => {
    console.log(this.name);
  }
};
```

---

## 8️⃣ Object Display

**How to display object data?**

---

### 1️⃣ Using Property Access

```
console.log(student.name + " " + student.course);
```

---

### 2️⃣ Using Loop

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

---

**JavaScript Dates**

### 1️⃣ JS Date Introduction

### 👉 What is Date in JavaScript?

JavaScript uses the **Date object** to work with **date and time**.

📌 Date stores:

- Year
- Month
- Day
- Hour, Minute, Second

---

**Create Current Date**

```
let today = new Date();
console.log(today);
```

📌 **Use:** Get current date & time from system.

---

## 2️⃣ JS Date Formats

**JavaScript accepts different date formats**

**JavaScript Date Formats (Correct Explanation)**

JavaScript stores dates internally as **milliseconds** since **January 1, 1970 (Unix Time)**.
To create a date, JavaScript provides different **Date formats**.

---

### 1. ISO Date Format (Most Recommended)

The **ISO format** is the **best and safest** way to create dates in JavaScript.

let date = new Date("2025-01-15");

**Format:**

YYYY-MM-DD

✓ Works in all browsers
✓ No ambiguity
✓ Best for exams, projects, and reports

---

**ISO Date with Time**

let date = new Date("2025-01-15T10:30:00");

**Used when date and time are both required** (logins, bookings, timestamps).

---

### 2. Date Using Numeric Values (Very Reliable)

let date = new Date(2025, 0, 15);

**Syntax:**

new Date(year, month, day)

Month starts from **0**

- 0 = January
- 11 = December

✓ Avoids format issues
✓ Good for logic-based questions

---

**3. Date with Full Parameters**

let date = new Date(2025, 0, 15, 10, 30, 0);

**Format:**

new Date(year, month, day, hour, minute, second)

✓ Provides complete control over date and time

---

**4. Date Using Text Format (Not Recommended)**

let date = new Date("January 15, 2025");

Depends on browser language
Not reliable for exams or production

✓ Only for basic understanding

---

**5. Date Using Timestamp**

let date = new Date(0);

This represents:

January 1, 1970, 00:00:00

✓ Used for calculations and comparisons
✓ Advanced concept

---

## 3️⃣ JS Date Get Methods

**Get methods read values from a Date object**

---

### Common Get Methods

```
let d = new Date();


d.getFullYear();   // Year (2025)
d.getMonth();     // Month (0–11)
d.getDate();      // Day (1–31)
d.getDay();       // Weekday (0–6)
d.getHours();     // Hours
d.getMinutes();   // Minutes
d.getSeconds();   // Seconds
d.getMilliseconds();
d.getTime();      // Milliseconds since 1970
```

---

### Example

```
let d = new Date();


console.log(d.getDate() + "-" + (d.getMonth()+1) + "-" + d.getFullYear());
```

**Use:** Show formatted date to user.

---

## 4️⃣ JS Date Set Methods

**Set methods change date values**

---

**Common Set Methods**

```
let d = new Date();


d.setFullYear(2026);
d.setMonth(5);     // June
d.setDate(20);
d.setHours(10);
d.setMinutes(45);
d.setSeconds(30);
```

---

**Example**

```
let d = new Date();
d.setDate(d.getDate() + 7); // add 7 days
console.log(d);
```

**Use:** Calculate future/past dates.

---

**5️⃣ JS Date Methods (Important)**

**Commonly used Date methods**

---

**Convert Date to String**

```
let d = new Date();


d.toDateString();    // Wed Jan 15 2025
```

```
d.toTimeString();    // 10:30:00 GMT...
d.toISOString();     // 2025-01-15T05:00:00.000Z
d.toLocaleDateString(); // Based on locale
d.toLocaleTimeString();
```

---

### 6️⃣ Date Comparison

**Compare Dates**

```
let d1 = new Date("2025-01-10");
let d2 = new Date("2025-01-15");

if (d1 < d2) {
  console.log("d1 is earlier");
}
```

---

### 7️⃣ Real-Life Use Example

**Age Calculator**

```
function calculateAge(birthYear) {
  let currentYear = new Date().getFullYear();
  return currentYear - birthYear;
}

console.log(calculateAge(2002));
```

---