Functions in JavaScript

🔷 What is a Function?

A **function** is a **block of code** designed to perform a specific task.
It helps you **reuse code** instead of writing the same thing multiple times.

Example:

```
function greet() {
  console.log("Hello, Raj!");
}
greet(); // Function call
```

Output:

```
Hello, Raj!
```

🔷 Function Syntax

```
function functionName(parameters) {
  // code to execute
}
functionName(arguments);
```

Example:

```
function add(a, b) {
  console.log(a + b);
}
add(5, 10); // Output: 15
```

🔷 Function with Return Value

A function can **return** a result using the `return` keyword.

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

let result = multiply(4, 5);
console.log("Result:", result); // Output: 20
```

🔷 Function with Default Parameters

You can give parameters default values.

```
function greet(name = "Guest") {
```

```
  console.log("Hello, " + name + "!");
}
greet();          // Hello, Guest!
greet("Raj");     // Hello, Raj!
```

🔹 Function Expressions

You can also assign a function to a variable.

```
const square = function(num) {
  return num * num;
};
console.log(square(5)); // Output: 25
```

🔹 Arrow Functions (ES6 Feature)

A shorter way to write functions.

```
const add = (a, b) => a + b;
console.log(add(10, 20)); // Output: 30
```

If the function has multiple lines:

```
const greet = (name) => {
  console.log("Hello, " + name);
};
greet("Raj");
```

🔹 Function Scope

Variables declared inside a function are **local** to that function.

```
function test() {
  let x = 10; // local variable
  console.log(x);
}
test();
console.log(x); // ❌ Error: x is not defined
```

🔹 Nested Functions

A function can be declared **inside another function**.

```
function outer() {
  function inner() {
    console.log("Hello from inner function");
  }
  inner();
}
```

```
outer();
```

✅ Mini Assignment (Practice)

1. Create a function `greetUser(name)` that prints “Hello, [name]!”.
2. Create a function `square(num)` that returns the square of a number.
3. Use an **arrow function** to multiply two numbers.
4. Create a function `calculateAge(birthYear)` that returns your current age.