

Date:18-06-24

JavaScript:-

1. Scope of JavaScript

- We can declare the variables inside any function ,then its scope remain under that function, then there is a local scope of Var.
- On the other hand, there is global scope, we can declare the variables anywhere & outside the function.
- Syntax-

```
    let a = "u";  
    {  
    let a = "u6";  
    console.log(a)  
    }  
    console.log(a)
```

2. Datatypes in JavaScript

- String
- Number
- BigInt
- Boolean
- Undefined
- Null
- Symbol
- Object

- The object data type can contain both **built-in objects**, and **user defined objects**
- Built-in object types can be:
objects, arrays, dates, maps, sets, intarrays, floatarrays, promises, and more.

3. Statements

- **If statement:-** the if statement is used to evalaute a particular condition. If the condition holds true, the associated code block is executed.
- **Syntax-**

```
    if ( condition ) {  
    // If the condition is met,  
    //code will get executed.}
```

- **If-else statement:-**The if-else statement will perform some action for a specific condition. Here we are using the else statement in which the else statement is written after the if statement and it has no condition in their code block.
- **Syntax-**

```

        if (condition1) {
            // Executes when condition1 is true
        }
        if (condition2) {
            // Executes when condition2 is true
        }
    }

```

- **Else-if statement:-** The else-if statement in javascript allows handling multiple possible conditions and outputs, evaluating more than two options based on whether the conditions are true or false.

- **Syntax-**

```

        if (1st condition) {
            // Code for 1st condition
        } else if (2nd condition) {
            // Code for 2nd condition
        } else if (3rd condition) {
            // Code for 3rd condition
        } else {
            // Code that will execute if all
            // above conditions are false
        }
    }

```

- **Switch statement:-** As the number of conditions increases, you can multiply else-if statements in javascript. But when we are dealing with many conditions, the switch statement may be a more preferred option.

- **Syntax-**

```

        switch (expression) {
            case value1:
                statement1;
                break;
            case value2:
                statement2;
                break;
            . . .
            case valueN:
                statementN;
                break;
            default:
                statementDefault;
        };
    }

```

- **Ternary operator (?:) :-** The conditional operator, also referred to as the ternary operator (?:), is a shortcut for expressing the conditional statements in javascript.

- **Syntax-**

```

        condition ? value if true : value if false
    }

```

- **Nested if-else statement:-** Nested if-else statements in javascripts allow us to create a complex conditional logic by checking the multiple conditions in a hierarchical manner. Each if statement can have an associated else block, and within each if or else block, you can nest another if-else statement. This nesting can continue to multiple levels, but it's important to maintain readability and avoid excessive complexity.

- **Syntax-**

```

        if (condition1) {
            // Code block 1
        }
        if (condition2) {

```

```
        // Code block 2
    } else {
        // Code block 3
    }
} else {
    // Code block 4
}
```