D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: – Scripting Concepts **Class: TY (A, B) (C.S.E.)-II**

Assignment No.: 1.A

S/W Requirement – notepad++,VScode, nodejs.

Title: A. Study of JavaScript basics

JavaScript is a widely used scripting language that helps in making web pages interactive. It runs on the client side and is mainly used to control webpage behavior, validate user input, handle events, and manipulate HTML elements dynamically. JavaScript is easy to learn and plays an important role in modern web development. JavaScript is a client-side scripting language used to make web pages dynamic and interactive. It is embedded in HTML using the <script> tag. JavaScript does not require compilation and runs directly in the browser.

JavaScript was created in 1995 by Brendan Eich.It was developed at Netscape Communications Corporation.Brendan Eich designed JavaScript in just 10 days.Initially, JavaScript was named Mocha.Later, it was renamed LiveScript and Finally, it was named JavaScript to gain popularity due to the success of the Java programming language (though Java and JavaScript are different).

In 1997**,** JavaScript was standardized by ECMA International**.** The standardized version of JavaScript is called ECMAScript**.** ECMA stands for European Computer Manufacturers Association**.** JavaScript has reached the version ES14.ECMA(European Computer Manufacturers Association).ECMAScript 2025 (ES2025) is the latest (16th) edition available now.

Tags in JavaScript

JavaScript uses the **<**script> tag in HTML to write and run JavaScript code.

<script> Tag

- Used to write JavaScript code inside an HTML page.
- It can be placed in the <head> or <body> section.

Example –

```
<!DOCTYPE html>
<html>
<head>
   <title>Basic JavaScript Program</title>
</head>
<body>

<h2>JavaScript Example</h2>
```

```
<script>
   document.write("Hello, Welcome to JavaScript!");
</script>

</body>
</html>
```

Here:

<!DOCTYPE html>→ Defines the document as an HTML5 document.

<html>→ Root element of the HTML page.

<head>→ Contains page information like the title.

<title>→ Displays the title in the browser tab.

<body>→ Contains the visible content of the webpage.

<h2>→ Displays a heading on the webpage.

<script>→ Used to write JavaScript code.

document.write("Hello, Welcome to JavaScript!");→ Displays the message on the webpage.

</script>→ Ends the JavaScript code block.

Difference between console.log(), alert() and document.write()

1. console.log()

Example:

```
console.log("Hello JavaScript");
```

Explanation:

- Prints output in the console
- Mainly used for debugging
- Does not disturb the webpage
- **Best for developers**

2. alert()

Example:

```
alert("Welcome to JavaScript");
```

Explanation:

- Shows a popup message
- Stops execution until user clicks OK
- Useful for important messages
- Not user-friendly for frequent use

3. document.write()

Example:

document.write("Hello World");

Explanation:

- Writes content directly on the webpage
- Can erase existing content if used after page load
- Mostly used for learning purposes
- Not recommended in real projects

JavaScript can be added to HTML file in two ways:

- **Internal JS:**
    - We can add JavaScript directly to our HTML file by writing the code inside the <script> tag.
    - The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.
- **External JS:**
    - We can write JavaScript code in another files having an extension.js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

Internal javascript Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Internal JS</title>
</head>
<body>
```

```
<h2>Internal JavaScript Example</h2>

<script>
  document.write("This is Internal JavaScript");
</script>

</body>
</html>
```

External JavaScript Example:

Step 1: JavaScript code written in a separate .js file.

Create JS File (script.js)

```
document.write("This is External JavaScript");
```

Step 2: Link JS File in HTML (1.html)

```
<!DOCTYPE html>
<html><head>
 <body>
<title>External JS</title>
    <script src="script.js"></script>
</head>
<h2>External JavaScript Example</h2>
</body>
</html>
```

Results –

1. Students should be able to study Basic of Javascript

Observations –

1. Student should able to execute the code by using the internal and external javascript.

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: – Scripting Concepts **Class: TY (A, B) (C.S.E.)-II**

Assignment No.: 1.B

S/W Requirement – notepad++,VScode, nodejs.

Title: B. Study of JavaScript basics

Variables

- A JavaScript variable is simply a name of storage location.
- Variables in JavaScript can be declared using var, let, or const. JavaScript is dynamically typed, so variable types are determined at runtime without explicit type definitions.
- Rules for Naming JavaScript Variables
    - Variable names must start with a letter, an underscore _, or the dollar sign $.
    - Variables cannot start with numbers.
    - Variable names are case-sensitive. So age and Age are different variables.
    - Variable names cannot be keywords (special words reserved for specific purposes in JavaScript such as if, else, let, var, etc.).

```
<html>
<head>
  <title> Variables in JavaScript </title>
</head>
<body>
  <script>
    var abc = "HIIIIIIIIIIIIIIII";
    var _abc = "Hi!";
    var $abc = "Hello!";
    //  var 9abc = "Bye!";  // This is invalid
    document.write(abc);
    document.write("_abc = " + _abc + "<br>");
    document.write("$abc = " + $abc + "<br>");
```

```
        let age = 23;
        let Age = 20;
        /*invalid  let new = 5;  Error! new is a keyword*/
        document.write(age);
        document.write("Age="+Age+"<br>");

        const num = 5;
        document.write("num = " +num+ "<br>")

    </script>
</body>
</html>
```

Control Statements

Control statements are used to control the flow of execution of a program.
They decide which statement will execute, how many times, and when to stop.

Types of Control Statements in JavaScript

1. Decision Making Statements
    - if
    - if–else
    - else–if ladder
    - switch
2. Looping Statements
    - for
    - while
    - do–while

1. if Statement

Meaning

Executes a block of code only if the condition is true.

Syntax

```
if(condition)
{
  statements;
}
```

Example

Problem Statement: Check whether a number is positive.

2. if–else Statement

Meaning

Executes one block if condition is true, otherwise executes else block.

Syntax

```
if(condition)
 {
  statements;
}
 Else
 {
  statements;
}
```

Example

Problem Statement: Check whether a person is eligible to vote.

3. else–if Ladder

Meaning

Used to check multiple conditions one by one.

Syntax

```
if(condition1)
{
  statements;
}
else if(condition2)
{
```

```
  statements;
}
 else
{
  statements;
}
```

Example

Problem Statement: Display grade based on marks.

4. switch Statement

Meaning

Used when there are multiple fixed choices.

Syntax

```
switch(expression)
 {
  case value1:
    statements;
    break;
  case value2:
    statements;
    break;
  default:
    statements;
}
```

Example

Problem Statement: Display day name using day number.

Looping Statements

5. for Loop

Meaning

Executes a block of code fixed number of times.

Syntax

```
for(initialization; condition; increment/decrement) {
```

```
  statements;
}
```

Example

Problem Statement: Print numbers from 1 to 5.

```
<script>
```

6. while Loop

Meaning

Executes code while the condition is true.

Syntax

```
while(condition)
{
  statements;
}
```

Example

Problem Statement: Print numbers from 1 to 5.

7. do–while Loop

Meaning

Executes code at least once, then checks condition.

Syntax

```
do
{
  statements;
} while(condition);
```

Example

Problem Statement: Print numbers from 1 to 5.

Results –

1. Students should be able to study Basic syntax of javascript

Observations –

1. Student should able to execute the code by using the control statements.

Prepared By:

Prof. K.A. Bhosale, Prof. S. N. Patil