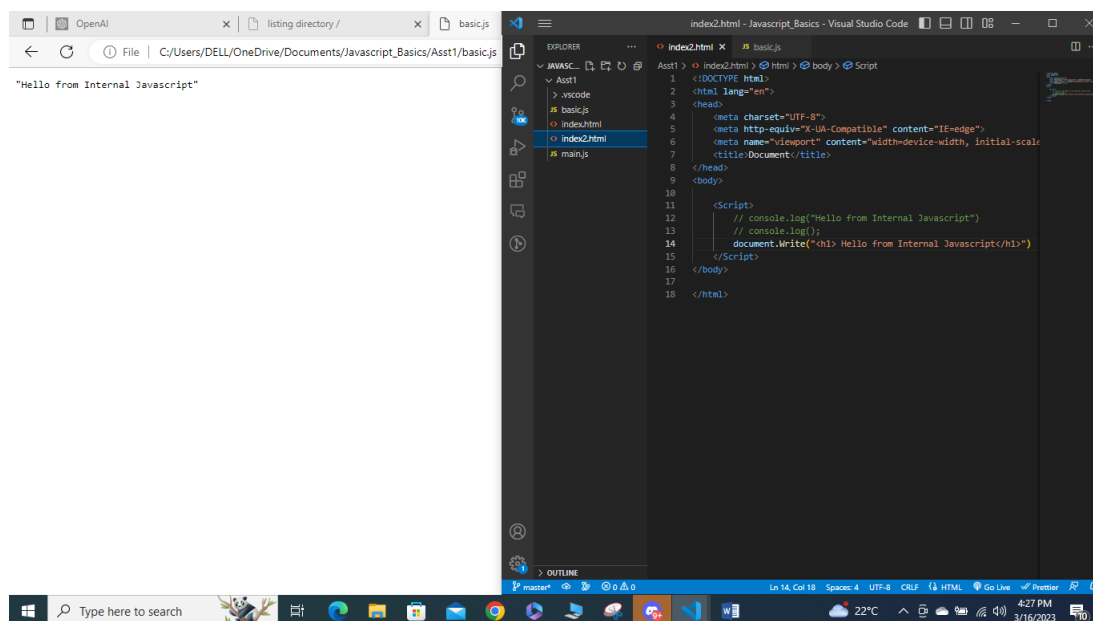
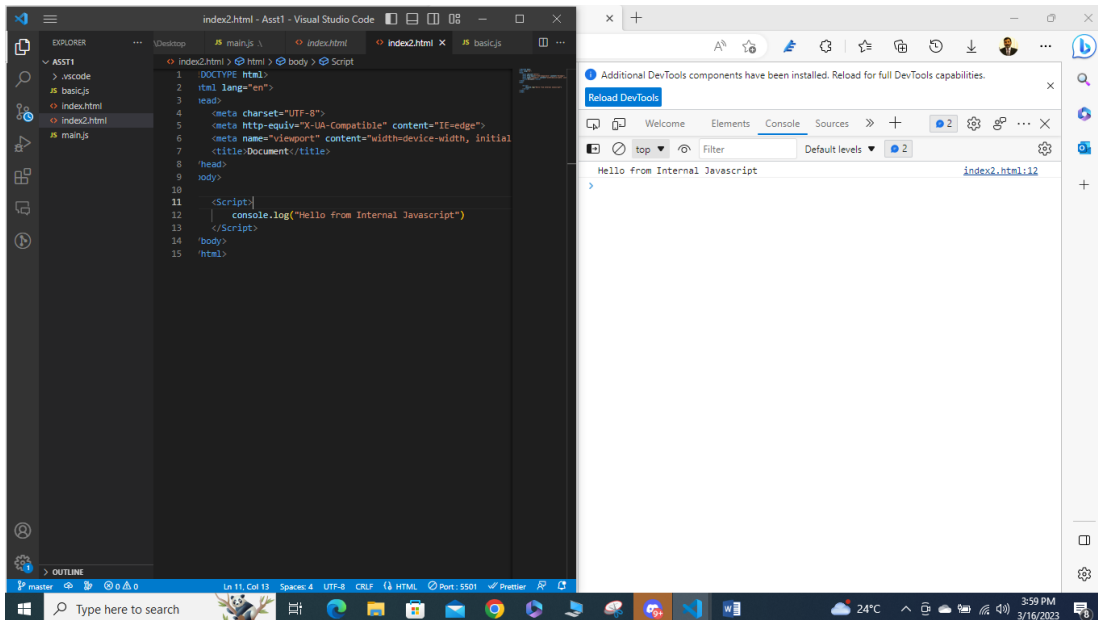


# Assignments of Session 3

By: Dr. Ghulam Shabbir

## Assignment No. 1:

*Create an index.html file and use JavaScript in it to output "Hello World" using either console or document.write.*



*Create a script.js file and link it with HTML and print the same output as the above assignment.*

## Assignment No. 2:

*Write about the difference between errors and bugs and describe the types of errors and bugs. Define them.*

Errors and bugs are both common in programming, but they have different meanings. Errors are mistakes made by the programmer when writing code, while bugs are errors in the program that cause it to behave unexpectedly. Understanding the types of errors and bugs can help programmers diagnose and fix issues more efficiently.

- In programming, the terms "**error**" and "**bug**" are often used interchangeably, but they actually have different meanings.
- An error is a mistake made by a programmer when writing code.
- These can be
  - **Syntax errors**, where the code is written incorrectly and the program can't run,
  - **Examples of syntax errors** include typos, missing or extra punctuation, and misspelled keywords.
  - **Logical errors**, where the code runs but doesn't produce the expected result.
  - Logical errors are more difficult to find because the program runs without any issues, but produces incorrect results. This can be due to faulty logic or incorrect assumptions made by the programmer.
  - **Bug** is an error in the program that causes it to behave in unexpected ways or to crash.
  - Bugs can be caused by errors made by the programmer, but they can also be caused by issues with the hardware or software environment.
  - **Examples of bugs** include program crashes, incorrect outputs, and unexpected behavior.

## **Types of Errors and Bugs:**

### Syntax errors:

These are errors caused by incorrect syntax in the code, such as

- missing semicolons
- incorrect brackets
- misspelled keywords

These errors prevent the program from running.

### Runtime errors:

These are errors that occur

- while the program is running, often caused by invalid inputs or
- unexpected conditions.

Common examples include division by zero, accessing an out-of-bounds array index, or attempting to read a file that doesn't exist.

### Logical errors:

These are errors caused by flawed logic in the program. They can be difficult to detect because the program may run without any issues, but produce incorrect results.

### Integration bugs:

- *These are bugs* that occur when different parts of a program don't work together correctly.
- This can be due to
  - incompatible software versions
  - data format issues
  - communication problems etc.

### Performance bugs:

- These are bugs that affect the speed or efficiency of the program.
- Common examples include memory leaks, inefficient algorithms, and unnecessary computations.

## Assignment No. 3:

*Explain what statements and expressions are in JavaScript and highlight the difference between them.*

The main difference between a statement and an expression is that a statement is used to perform an action or execute a code block, whereas an expression is used to produce a value.

### Javascript Statement

- In JavaScript, a statement is a piece of code that performs an action.
- It can include
  - variables
  - functions
  - loops
  - keywords, such as *if*, *for*, *while*, and *switch*.
- A statement can be a single line of code or a block of code enclosed in *curly braces {}*

Example: the following is a statement that assigns the value 10 to the variable x:

```
let x; x = 10;
```

### Javascript Expression

- It is a piece of code that evaluates to a value.
- It can include variables, operators, and function calls.
- An expression can be as simple as a single variable or value, or it can be more complex with multiple operators and function calls.

#### Example:

- Following is an expression that evaluates to the value 20:

`let x = 10; let y = x * 2; // y is now 20`

- Expressions can also be used within statements. For instance, an if statement uses an expression to determine whether to execute the following code block:

`let x = 10; if (x === 10) { console.log("x is equal to 10"); }`

In this example, the expression `x === 10` evaluates to a boolean value of true, which then executes the code block within the if statement.

#### Assignment No. 4:

*Define syntax in JavaScript or any other programming language, and also explain how to use comments in JavaScript and the different types of comments available.*

Comments are useful for adding notes to code, explaining complex code blocks, and making the code more readable and maintainable. However, it is important to use comments judiciously and avoid over-commenting as it can make the code harder to read.

- Syntax in programming languages refers to the set of rules that dictate the structure, format, and organization of statements, expressions, and other elements in the language.
- A programming language follows a specific syntax to ensure that the code is consistent, readable, and easy to understand.
- Comments in JavaScript are used to add notes and explanations to the code. These comments are ignored by the browser and do not affect the code's execution.
- JavaScript supports two types of comments:
  1. **Single-line comments:** These are used to add a comment that spans a single line of code. In JavaScript, single-line comments are denoted using two forward slashes (//) at the beginning of the comment line. For example:
 

```
// This is a single-line comment in JavaScript var x = 10;
```

```
// This line declares and initializes a variable x with a value of 10
```
  2. **Multi-line comments:** These are used to add a comment that spans multiple lines of code. In JavaScript, multi-line comments are denoted using forward slash and asterisk (/\*) at the beginning of the comment block and asterisk and forward slash (\*/) at the end of the comment block. For example:

```
/* This is a multi-line comment in JavaScript. It can span multiple lines of code. */
```

```
var y = 20;
```

```
/* This line declares and initializes a variable y with a value of 20 */
```