TechLift
KICKSTART YOUR CAREER

WEB ENGINEERING

.Net

P@SHA

PAKISTAN PSEB SOFTWARE EXPORT BOARD

Week 1 - Day 2 & 3

Learning Objectives

By the end of this session, the students will have reinforced to further developed an understanding of:

- What is JavaScript
- Inline, Internal and External JavaScript
- Display Output using JavaScript
- JavaScript Syntax
- Statements, Variables

: A Quick Review

This week is focused on everyone (students with technical/non technical background) starting from the same ground with HTML & JavaScript revisions.

What is Javascript?

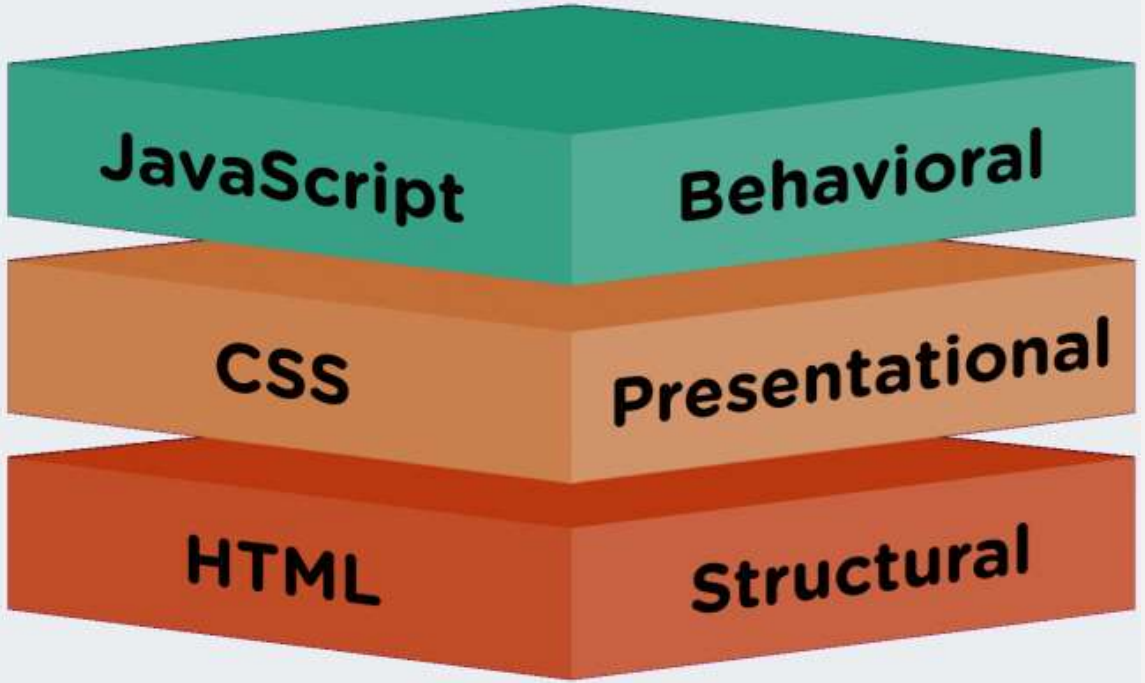
- Invented by Brendan Eich (1995)
  - Former CTO / CEO of mozilla
- A scripting language used for client-side web development
- Cross-platform
- Object based
- Dynamic
- Scripting language
- Currently ECMA-262 (ECMAScript 5)
- Web APIs (DOM, ..)
- JScript is the Microsoft version of JavaScript.

What is Javascript?

HTML for content

CSS for presentation

Javascript for interactivity



Language features - The good

- Everything is an Object
  - Including functions
- Objects are loosely typed
  - Every object can be assigned any value
- Objects are dynamic
  - Can change type @ runtime
  - Add members on the fly
- Object literals

Language features - The bad

- Easy to introduce globals = root of all evil
- Automatic line termination
  - ► Semicolons automatically inserted by interpreter
  - ► Can lead to quirky behavior
  - ► No warning
- Object equality and Object comparison
  - ► Error prone due to dynamically typed objects
  - ► Type coercion happens automagically (= auto casting)

What is Javascript?

Embedding javascript in a web page

```
<html>
    <head>
        <title>Hello Javascript</title>
    </head>
    <body>
        <script type="text/javascript">
            // your code should go here
        </script>
    </body>
</html>
```

What is Javascript?

Linking to a Javascript file

```
<html>
    <head>
        <title>Hello Javascript</title>
    </head>
    <body>
        <script type="text/javascript" src="app.js"></script>
    </body>
</html>
```

So, what can you do with JavaScript?

1. Mobile Apps
2. Front-end Web Development
3. Back-end Web Development
4. JavaScript as a Serverless Computing Programming Langauge
5. Desktop Applications
6. 2D & 3D Game Development
7. Artificial Intelligence
8. You can write plugins for popular applications
9. Data Visualization
10. Embedded Device Development and IoT

What can JavaScript do?

Click on each option below to view the embedded example:

1. Javascript can change HTML content
2. Javascript can change HTML attributes
3. Javascript can hide HTML elements
4. Javascript can shor hidden HTML elements

Types of Javascript

We can write the javascript code in three various ways, they are

1. Inline javascript
2. Internal Javascript
3. External Javascript

Inline Javascript

Inline javascript is written within the html tags.

Try this code

Example

```
<button onclick="alert('You clicked me');">Click Me</button>
```

Internal Javascript

Internal javascript is written inside the <script> and </script> tags.

Try this code:

Example

```
<script>
 function funCall()
 {
     alert("You clicked me");
 }
</script>
```

External Javascript

External javascript is written in a separate file and saved with .js extension.

External javascript file can be imported inside the html file using src attribute in <script> tag.

In the below example, a separate javascript file(ext.js) is created. In it we have defined **funCall** function. To view the javascript file Click //tutorialcodeplay.com/ext.js

Try this code:

Example

```
<script src="ext.js"></script>
```

We recommend you to define the javascript code above the </body> tag.

JavaScript Output

JavaScript Display Possibilities

(Click on the red links to view relevant examples)

JavaScript can "display" data in different ways:

Writing into

1. an HTML element, using innerHTML
2. the HTML output using document.write() &
3. Using document.write() after an HTML document is loaded, will delete all existing HTML: Example 2
   - The document.write() method should only be used for testing.

JavaScript Output

JavaScript Display Possibilities

JavaScript can "display" data in different ways:
Writing into

4. an alert box, using `window.alert()`
   - ➢ You can skip the window keyword.
   - ➢ In JavaScript, the window object is the global scope object, that means that variables, properties, and methods by default belong to the window object. This also means that specifying the window keyword is optional: Example 2

JavaScript Output

JavaScript can "display" data in different ways:

Writing into

5. the browser console, using `console.log()`
6. JavaScript Print: JavaScript does not have any print object or print methods.
   - You cannot access output devices from JavaScript.
   - The only exception is that you can call the `window.print()` method in the browser to print the content of the current window.

JavaScript Syntax

JavaScript syntax refers to the set of rules that determines how JavaScript programs are constructed. The JavaScript syntax is similar to C# and Java:

- Operators (+,*, =, !=, &&, ++, …)
- Variables (typeless)
- Conditional statements (if, else)
- Loops (for, while)
- Arrays (my_array[]) and associative arrays (my_array['abc'])
- Functions (can return value)
- Function variables (like the C# delegates)

JavaScript Syntax

JavaScript syntax refers to the set of rules that determines how JavaScript programs are constructed. The JavaScript syntax is similar to C# and Java:

- JavaScript can be implemented using JavaScript statements that are placed within the <script>... </script> HTML tags in a web page.
- You can place the <script> tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the <head> tags.
- The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows.

```
<script ...>
JavaScript code
</script>
```

JavaScript Syntax

- The script tag takes two important attributes −
  - **Language −** This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.
  - **Type** − This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

JavaScript Syntax

- Statements
- Numbers
- Strings
- Variables
- Operators
- Assignments
- Expressions (Using Constants, Strings and Variables
- Keywords
- Comments
- is case sensitive

JavaScript Values

The JavaScript syntax defines two types of values:

1. Fixed values
2. Variable values

Fixed values are called **Literals.**

Variable values are called **Variables.**

JavaScript Literals

The two most important syntax rules for fixed values are:

1. **Numbers** are written with or without decimals:

Try it yourself:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Numbers</h2>

<p>Number can be written with or without decimals.</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = 10.50;
</script>

</body>
</html>
```

Numbers - Output

JavaScript Literals

The two most important syntax rules for fixed values are:

2. <u>Strings</u> are text, written within double or single quotes:

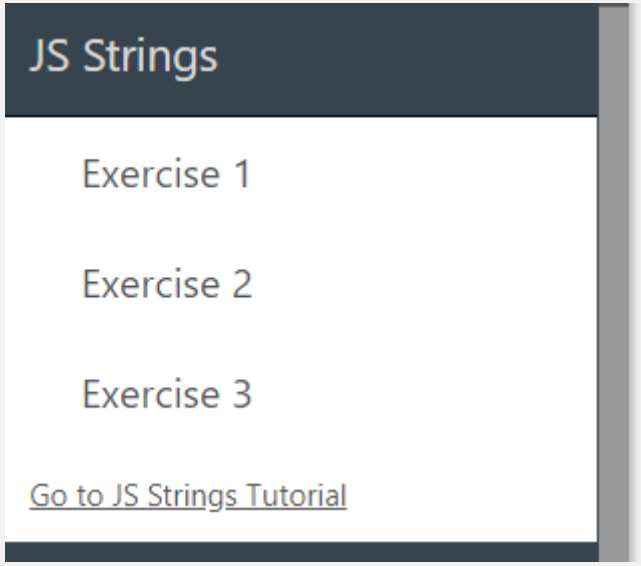
Practice Example:
Try it yourself:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Strings</h2>

<p>Strings can be written with double or single quotes.</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = 'John Doe';
</script>

</body>
</html>
```

Strings - Output

Strings - Exercises

Students will individually complete Strings exercises from this link:
https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables3



Quick Recap of Day 2

What are the following:

1. Javascript
2. Language features: The Good and The Bad
3. Types of JavaScript
4. JavaScript Input
5. JavaScript Syntax
6. Literals
7. Numbers
8. Strings

Day 3

Variables

In a programming language, **variables** are used to **store** data values.

JavaScript uses the keywords `var`, `let` and `const` to **declare** variables.

An **equal sign** is used to **assign values** to variables.

In this example, x is defined as a variable. Then, x is assigned (given) the value 6:

Variables - Practice Example

Try it yourself.

Then tally your answer with the one given on the next slide

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Variables</h2>

<p>In this example, x is defined as a variable.
Then, x is assigned the value of 6:</p>

<p id="demo"></p>

<script>
let x;
x = 6;
document.getElementById("demo").innerHTML = x;
</script>

</body>
</html>
```

Variables - Practice Example output

JavaScript Variables

In this example, x is defined as a variable. Then, x is assigned the value of 6:

6

Variables

EXAMPLES

```
var v1, v2, v3;     // Declare 3 variables
v1= 5;          // Assign the value 5 to v1
v2= 6;          // Assign the value 6 to v2
v3 = a + b;       // Assign the sum of v1and v2 to v3
```

```
a = 5; b = 6; c = a + b;
```

```
var  sname=  "Anju";
var  sname="Anju";
```

Declaring or Creating JavaScript Variables

Creating a variable in JavaScript is called "declaring" a variable.

Declare a JavaScript variable with the var keyword:

```
var Studname;
```

To assign a value to the variable, use the equal sign:

```
studname = "Anju";
```

We can assign a valu to the variable when you declare it:

```
var Studname = “Anju”;
```

Javascript Variables

4 Ways to Declare a JavaScript Variable:

- Using `var`
- Using `let`
- Using `const`
- Using nothing

Variables are containers for storing data (storing data values).

In this example, x, y, and z, are variables, declared with the var keyword:

```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Variables</h1>

<p>In this example, x, y, and z are variables.</p>

<p id="demo"></p>

<script>
var x = 5;
var y = 6;
var z = x + y;
document.getElementById("demo").innerHTML =
"The value of z is: " + z;
</script>

</body>
</html>
```

Javascript Variables

Example 2:

In this example, x, y, and z, are variables, declared with the let keyword:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Variables</h2>

<p>In this example, x, y, and z are variables.</p>

<p id="demo"></p>

<script>
let x = 5;
let y = 6;
let z = x + y;
document.getElementById("demo").innerHTML =
"The value of z is: " + z;
</script>

</body>
</html>
```

Javascript Variables

Example 3:

In this example, x, y, and z, are undeclared variables:

—----------

From all the examples above, you can guess:

- ➢ x stores the value 5
- ➢ y stores the value 6
- ➢ z stores the value 11

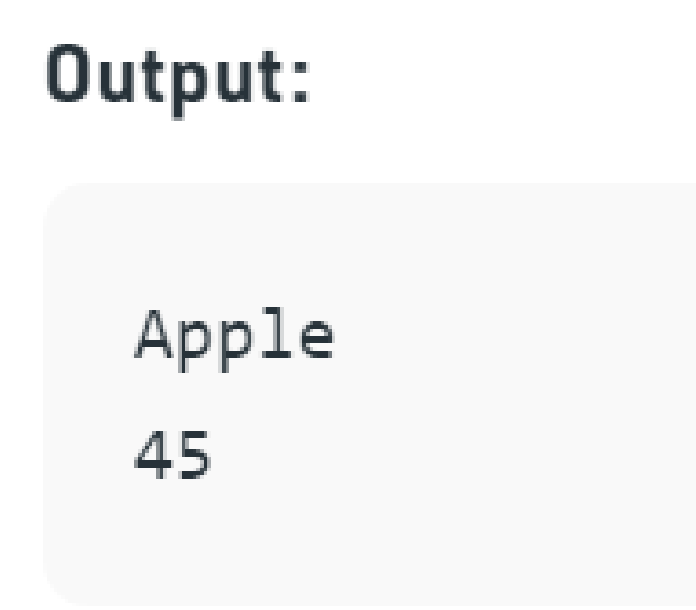
```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Variables</h1>

<p>In this example, x, y, and z are undeclared variables.</p>

<p id="demo"></p>

<script>
x = 5;
y = 6;
z = x + y;
document.getElementById("demo").innerHTML =
"The value of z is: " + z;
</script>

</body>
</html>
```

When to Use JavaScript `var`?

- Always declare JavaScript variables with var,let, orconst.
- The var keyword is used in all JavaScript code from 1995 to 2015.
- The let and const keywords were added to JavaScript in 2015.
- If you want your code to run in older browser, you must use var.

Practice Task:

Write down a JS syntax for the following output:



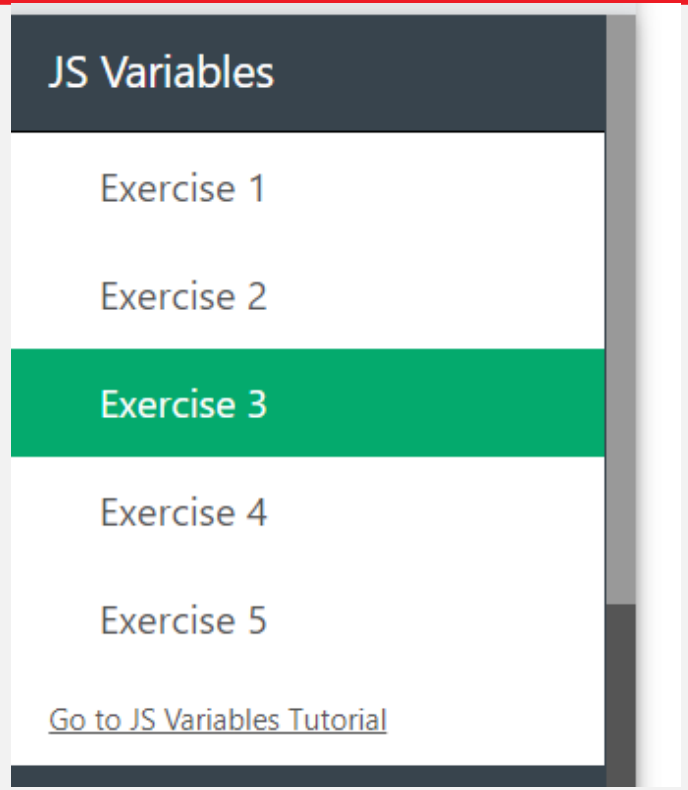
Check the code given on the next slide after you have completed your code.

```
<script>

// Declare a variable and initialize it
// Gloabal variable declaration
var Name="Apple";

// Function definition
function MyFunction() {

    // Local variable declaration
    var num = 45;

    // Display the value of Gloabal variable
    document.writeln(Name);

    // Display the value of local variable
    document.writeln("<br>" + num );
}

// Function call
MyFunction();

</script>
```

Task: Variables

Variable Exercises:

Students will individually complete **Variables** exercises from this link: https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables3



Declaring a Variable

```
var someNumber = 42; // The good way
someOtherNumber = 42 // The evil way
```

- The good way
  - Use var keyword
  - Variable always declared on current scope
- The evil way
  - No keyword
  - Variable always declared on global scope
  - Warning in strict mode

Variable Scope

- Global scope
  - Variable declared outside of any function
  - Accessible to any other code in the document
- Function scope
  - Variable declared inside of any function
  - Accessible to function and all inner functions
- Block scope
  - Variable declared within block is local to containing function

Statements

JavaScript statements are composed of:

- Values
- Operators
- Expressions
- Keywords, and
- Comments

Order of execution of Statements is the same as they are written.

Semicolons:

- Semicolons separate JavaScript statements.
- Semicolon marks the end of a statement in javascript

Statements

Statements also very similar to Java

Conditional	Loops	Exceptions
if(){}else{}	while(){};	throw
switch(){}	do{}while();	try{}catch(){};
	for(){};	
	continue;	
	break	

JavaScript - if...else Statement

- Trainer will demonstrate the code writing as example. Students will follow the trainer by writing the same code or any other exmaple given in the following link:

  https://www.tutorialspoint.com/javascript/javascript_ifelse.htm

- **JavaScript - Switch Case** will also be practiced in the same way

JavaScript - if...else Statement

- Trainer will demonstrate the code writing as example. Students will follow the trainer by writing the same code or any other example given in the following link:

  https://www.tutorialspoint.com/javascript/javascript_ifelse.htm

- **JavaScript - Switch Case** will also be practiced in the same way

Statements:

Practice Exercise

Write a statement for the following output. Then telly your code with the one given on the next slide:

Output:

Welcome

The value of c is 5.

```
<!DOCTYPE html>
<html>

<body>
    <h2>Welcome</h2>

    <p id="geek"></p>

    <script>
        // Statement 1
        var a, b, c;

        // Statement 2
        a = 2;

        // Statement 3
        b = 3;

        // Statement 4
        c = a + b;

        document.getElementById(
          "geek").innerHTML =
            "The value of c is " + c + ".";
    </script>

</body>

</html>
```

Example Statements

Expressions & Operators

Expressions and operators very similar to Java

- Precedence also similar

Assignment	Comparison	Arithmetic	Bitwise	Logical	String
+=	==	%	&	&&	+
-=	!=	++	|	||	+=
*=	===	--	^	!	
/=	!==	-	~		
%=	>		<<		
<<=	>=		>>		
>>=	<		>>>		
>>>=	<=				
&=					
^=					
|=					

JavaScript Operators

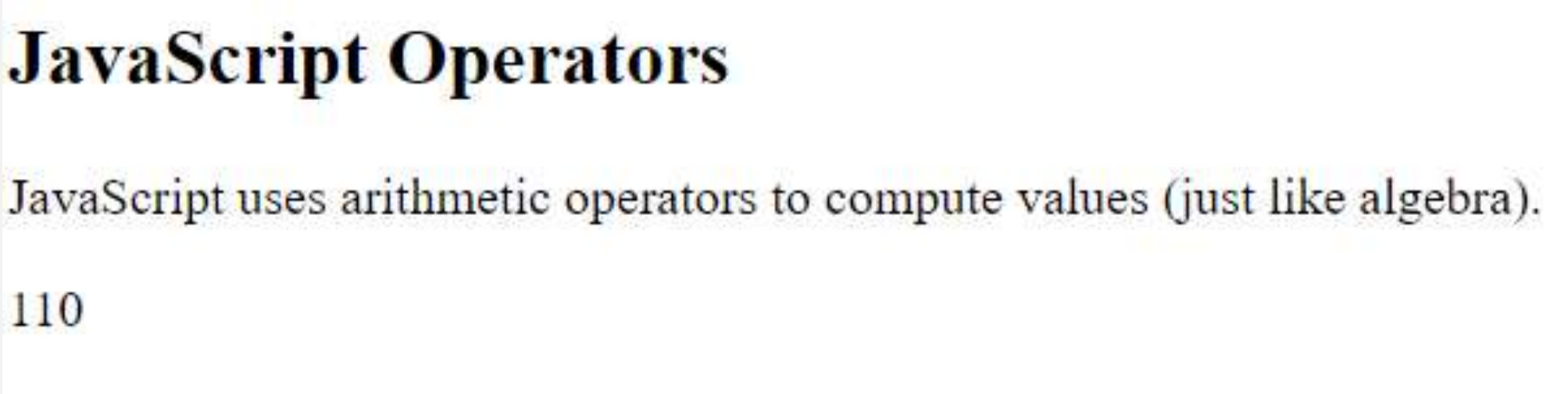
JavaScript uses **arithmetic operators** (+ - * /) to compute values:

Try it yourself.

Then tally your answer with the one given on the next slide

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Operators</h2>

<p>JavaScript uses arithmetic operators to compute values (just like algebra).</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = (5 + 6) * 10;
</script>

</body>
</html>
```

JavaScript Operators Example Output



JavaScript Operators

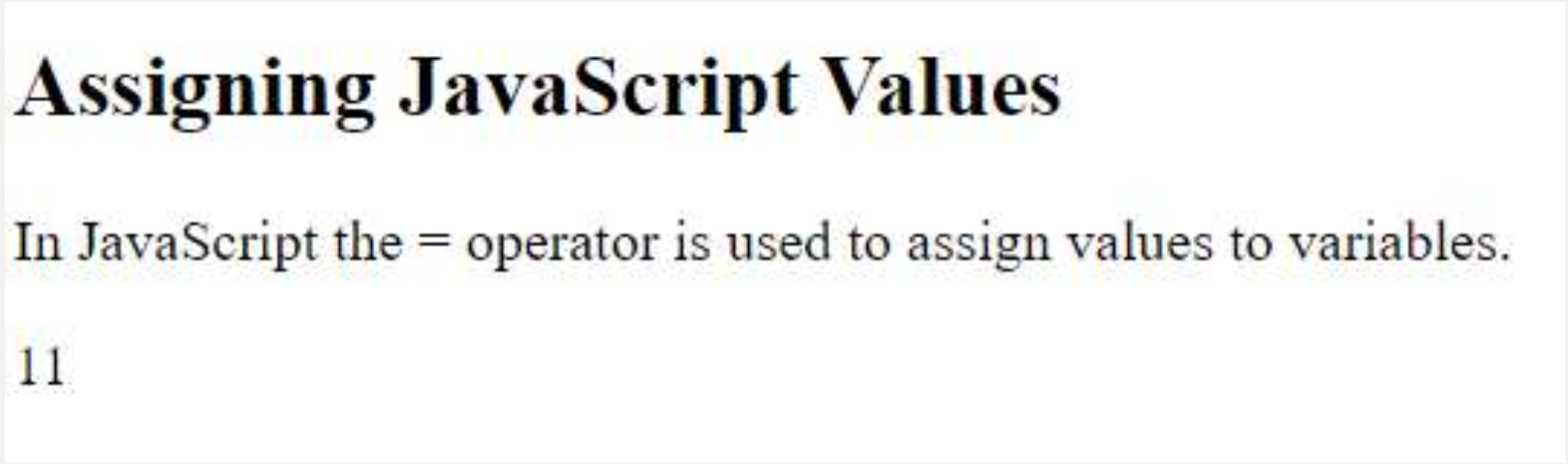
JavaScript uses an **assignment operator** (=) to assign values to variables:

Try it yourself.

Then tally your answer with the one given on the next slide

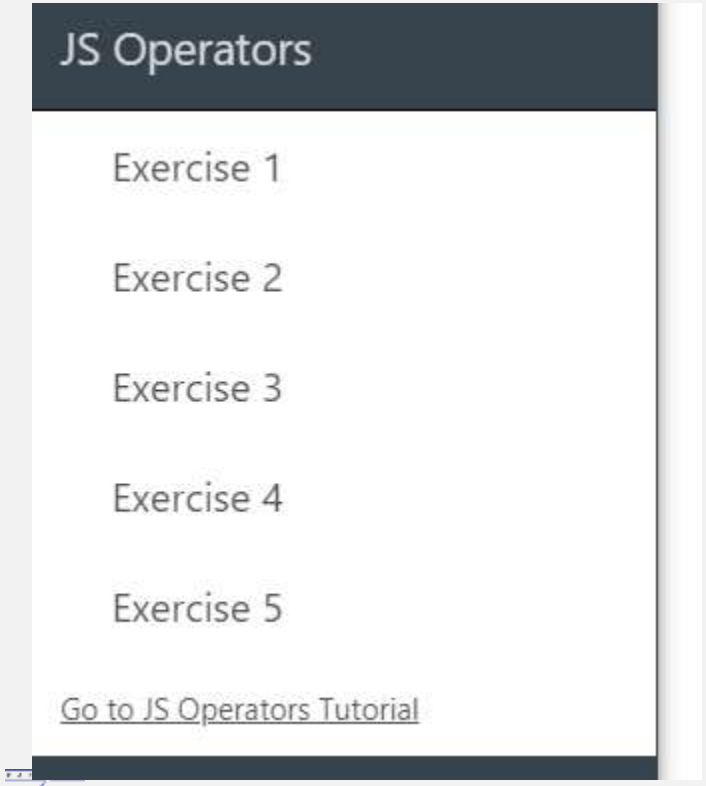
```
<!DOCTYPE html>
<html>
<body>

<h2>Assigning JavaScript Values</h2>

<p>In JavaScript the = operator is used to assign values to variables.</p>

<p id="demo"></p>

<script>
let x, y;
x = 5;
y = 6;
document.getElementById("demo").innerHTML = x + y;
</script>

</body>
</html>
```

JavaScript Operators Example Output



Operator Exercises:

Students will individually complete **Operators** exercises from this link: https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables3



JavaScript Expressions

- An expression is a combination of values, variables, and operators, which computes to a value.
- The computation is called an evaluation.
- For example, 5 * 10 evaluates to 50:
- Try writing an expressions with the output. Tally your code with the one given on the next slide:

JavaScript Expressions

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Expressions</h2>

<p>Expressions compute to values.</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = 5 * 10;
</script>

</body>
</html>
```

JavaScript Expressions

Practice each of the following examples in these steps:

Step 1:

Write the code on your computers

Step 2:

Tally your answer

- Javascript expressions (using constants)
- Javascript expressions (using strings):
  The values can be of various types, such as numbers and strings.
- Javascript expressions (using variables):
  Expressions can also contain variable values.

JavaScript Keywords

Practice each of the following examples in these steps:

Step 1:

Write the code on your computers

Step 2:

Tally your answer

- JavaScript keywords are used to identify actions to be performed.
- The let keyword tells the browser to create variables:

Keywords Example

Functions in JavaScript

Practice each of the following examples in these steps:

Step 1:

Write the code on your computers

Step 2:

Tally your answer

- JavaScript keywords are used to identify actions to be performed.
- The let keyword tells the browser to create variables:

Keywords Example

Javascript Functions

Follow the trainer on your computers as they demonstrate the following functions on their computer.

- Javascript | Arrow functions
- JavaScript | escape()
- JavaScript | unescape()
- JavaScript | Window print()
- Javascript | Window Blur() and Window Focus() Method
- JavaScript | console.log()
- JavaScript | parseFloat()
- JavaScript | uneval()
- JavaScript | parseInt()
- JavaScript | match()
- JavaScript | Date.parse()
- JavaScipt | Replace() Method
- JavaScript | Map.get( )
- JavaScript | Map.entries( )
- JavaScript | Map.clear( )
- JavaScript | Map.delete()
- JavaScript | Map.has( )

- Javascript variables
- Javascript variables as algebra
- Javascript numbers and strings
- Javascript var keyword
- Declaring many variables in one statement
- Declaring many variables multiline
- A variabel without a value returns the value undefined
- Re-declaring a variable will not destroy the value
- Adding Javascript numbers
- Adding Javascript strings
- Adding strings and numbers

Variables

Homework

Practice all the examples and come prepared with your queries tomorrow.

Javascript Statements

- Javascript statements are commands to the browser
- Javascript code is a sequence of statements
- Javascript statements are separated with semicolon
- Multiple statements on one line is allowed
- Javascript statements can be grouped together in code blocks
- You can break a code line after an operator or a comma

Homework

Practice all the examples and come prepared with your queries tomorrow.

Learning Objectives

By the end of this session, the students have practised

- What is JavaScript
- Inline, Internal and External JavaScript
- Display Output using JavaScript
- JavaScript Syntax
- Statements, Variables

Conclusion & Q/A

See you tomorrow!