



Student Identity

Name : Lovie Jechonia Tonimba

NIM : 244107060101

Class : 2G

Major : D-IV Business Information System

Topic

1. Introduction to Basic Concepts of JavaScript
2. Data Types, Operators, and Functions in JavaScript
3. JavaScript in HTML

Objectives

Students are expected to:

1. Understanding the concept of Javascript
2. Understanding Data types, operators and functions in javascript
3. Students are able to run javascript in HTML files

Introduction

JavaScript is a client-side programming language used for web development. A client-side programming language means that the processing is done on the client's side. The client application in this case refers to web browsers such as Google Chrome and Mozilla Firefox. Client-side programming languages differ from server-side programming languages like PHP, where all the program code is executed on the server side.

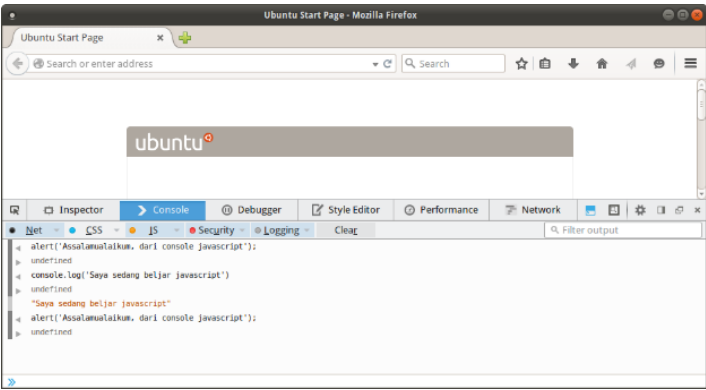
To run JavaScript, you only need a text editor and a web browser. JavaScript offers features such as being a high-level programming language, client-side, loosely typed, and object-oriented. Initially, JavaScript was developed to make interactions between users and websites faster without having to wait for processing on the web server. Before JavaScript, every interaction from the user had to be processed by the web server.

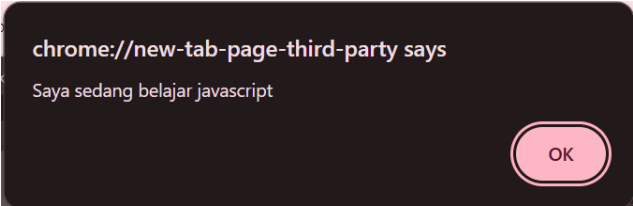

Imagine when you fill out a registration form on a website, click the submit button, wait about 10 seconds for the website to process the form, and then receive a page stating that some form fields were left unfilled. This is the kind of situation for which JavaScript was developed. The processing to check whether all form fields have been filled can be transferred from the web server to the web browser.

As JavaScript evolved, it became useful not only for form validation but also for many modern purposes. Various animations to beautify web pages, chat features, modern effects, games—all of these can be created using JavaScript. There are 3 main ways to write JavaScript tags:

1. Writing the tag with `<script type="text/javascript">` at the start and ending with `</script>`. The attribute informs the browser that the script within the tag is JavaScript in text format.
2. Writing the tag with `<script language="javascript">` at the start and ending with `</script>`. This attribute is used to specify the version of JavaScript being used. For example, `<script language="javascript1.2">` indicates that the version of JavaScript used is 1.2.
3. Writing the tag with `<script language="javascript" type="text/javascript">` at the start and ending with `</script>`. This mixed method combines the old and new ways of writing, allowing compatibility for web browsers that support JavaScript but may not yet support HTML fully.

Practical Section 1: Learning Javascript

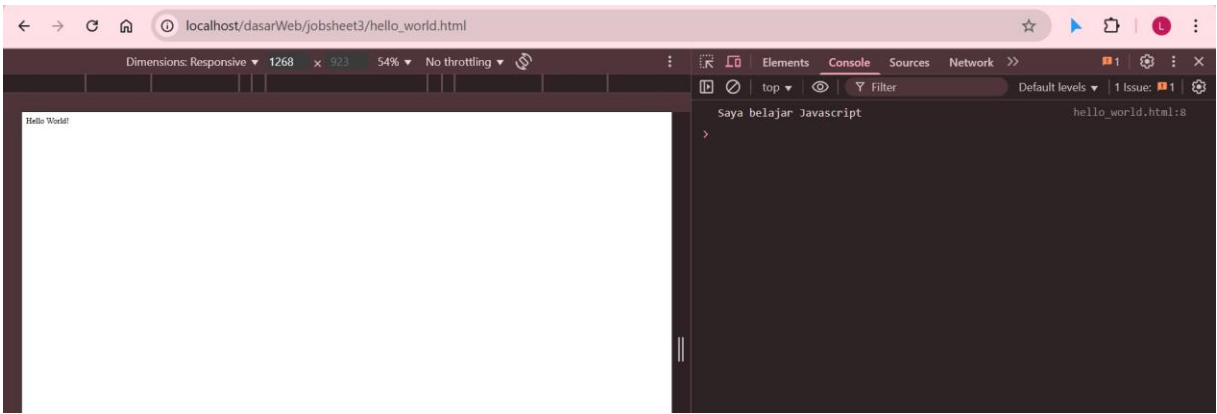
Step	Description
1	<p>We can open the JavaScript console through Inspect Element -> Console.</p>  <p>In the console, we can write functions or JavaScript code, and the results will be displayed immediately.</p>
2	<p>For example, let's try the following code:</p> <pre>console.log("Hi apa kabar!"); alert("Saya sedang belajar javascript");</pre> <p>Observe what appears on the console, then record your observations!.</p> 

	 <p>The console.log(“Hi, how are you?”) function will print the string “Hi, how are you?” directly to the console. Meanwhile, the alert(“I am learning JavaScript”) function will display a pop-up dialog on the browser screen with the message “I am learning JavaScript”. This pop-up will block user interaction with the web page until the pop-up is closed.</p>
3	<p>If you are using Nodejs, then the way to access <i>the console</i> is to type the node’s command in the Terminal.</p> 
4	<p>Observe what happens, then record your observations. What can be concluded after trying <i>the Javascript</i> console?</p> <p>(Question No.1)</p> <p>The console.log(“Learning Nodejs”) command successfully printed the string “Learning Nodejs” to the console. Second, the variable declaration var name = “Petanikode” was successfully executed without producing any visual output, because it was only a declaration and not a print command. Finally, console.log(“My name” + name) successfully concatenates the string “My name” with the value of the variable name, which contains ‘Petanikode’, and prints “My namePetanikode” to the console. This shows that the console can be used to execute JavaScript code line by line, view the output directly, and test variables.</p>

Practical Section 2: Creating the First Javascript Program

Step	Description
------	-------------

1	Please open a text editor, then create a new file named <code>hello_world.html</code>
2	<p>Type the program code below:</p> <pre> <!DOCTYPE html> <html> <head> <title>Hello World Javascript</title> </head> <body> <script> console.log("Saya belajar Javascript"); document.write("Hello World!"); </script> </body> </html> </pre> 
3	<p>Save it as <code>hello_world.html</code>, then open the file with a web browser.</p> 
4	Observe what happens in the browser, then record your observations (Question No.2)


	When this HTML file is opened in a browser, two things will happen. First, the line <code>document.write("Hello World!")</code> directly displays the text "Hello World!" on the web page.
5	<p>Now try to open the javascript console, right click page in the browser, then choose Inspect Elements > Console</p> 
6	<p>Observe what happens in the Console tab, then record your results!</p> <p>(Question No. 3)</p> <p>Second the line <code>console.log("I am learning JavaScript")</code> will not be visible on the web page, but the text "I am learning JavaScript" will be printed in the JavaScript Console</p>
7	<p>Earlier, we wrote the command:</p> <pre>console.log("Saya belajar Javascript");</pre> <p>Why do you think the command is not displayed?</p> <p>(Question No.4)</p> <p>The code <code>console.log("Saya belajar Javascript")</code> won't be visible on a web page because it's designed to print output to the developer console, not the document body. The console is a tool primarily used by developers for debugging and testing code.</p>

Practical Section 3: How to Write Javascript Code in HTML

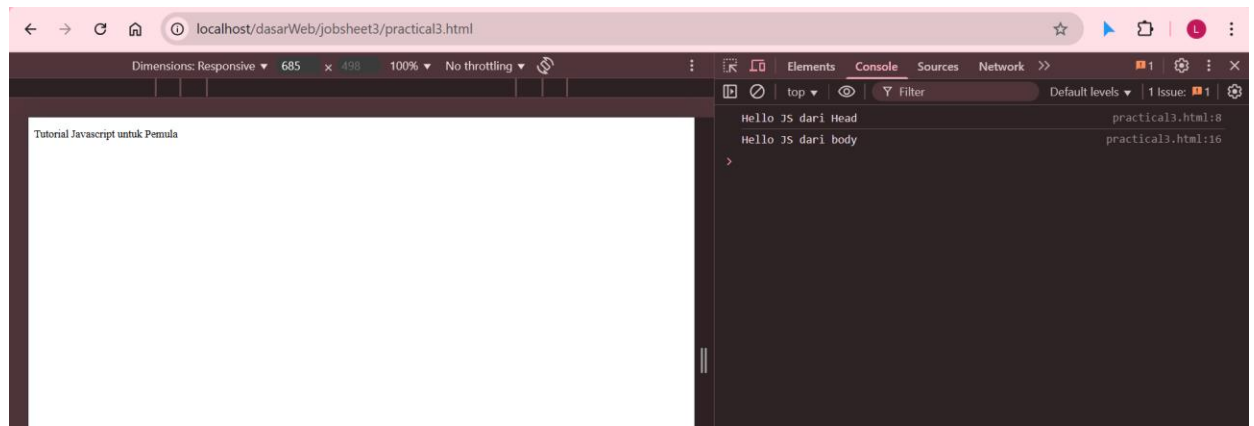
In practicum 2 we have written javascript code in HTML, this method is an embedded writing method. Some other ways that we need to know include:

1. **Embed** (Javascript code pasted directly into HTML)
2. **Inline** (Javascript code written on HTML attributes)
3. **External** (Javascript code is written separately from the HTML file)

1. Writing Javascript Code with Embed	
Step	Description

1	In this way, we use the <code><script></code> tag to embed <i>the</i> Javascript code in the HTML. These tags can be written in the <code><head></code> and <code><body></code> tags
2	<p>Type the program code below:</p> <pre> <!DOCTYPE html> <html> <head> <title>Belajar Javascript dari Nol</title> <script> // ini adalah penulisan kode javascript // di dalam tag <head> console.log("Hello JS dari Head"); </script> </head> <body> <p>Tutorial Javascript untuk Pemula</p> <script> // ini adalah penulisan kode javascript // di dalam tag <body> console.log("Hello JS dari body"); </script> </body> </html> </pre> 
3	Observe what happens to the browser? Record your observations

(Question No. 5)



Two things are recorded in the browser console. First, the line `console.log("Hello JS from Head")` inside the `<head>` tag is executed, and the result immediately appears in the console. Second, the line `console.log("Hello JS from body")` inside the `<body>` tag is also executed, and the output appears right below the first message in the console. This shows that JavaScript can be executed from both the `<head>` and `<body>` sections—and messages from both locations will be printed to the console. You can also see that the text in the `<p>` tag is displayed in the browser.

4

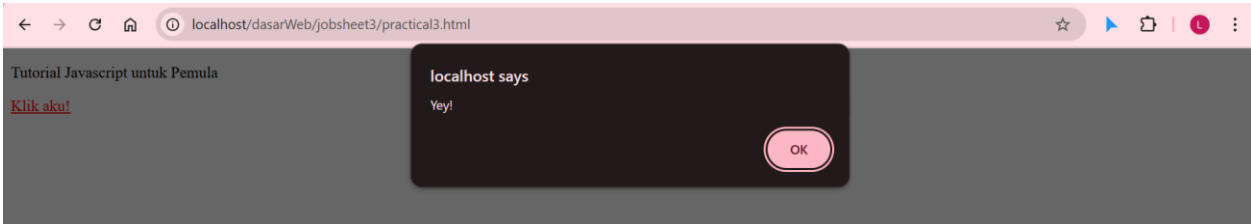
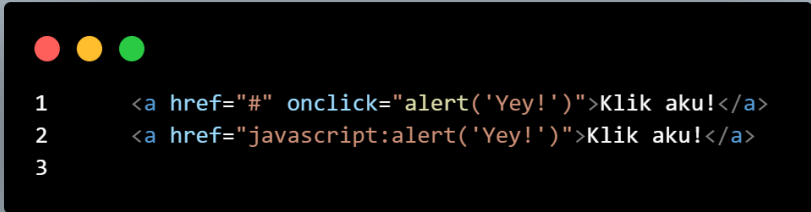
Which do you think is better, written in the `<head>` or `<body>` tag?

(Question No. 6)


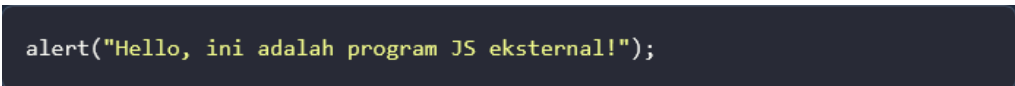
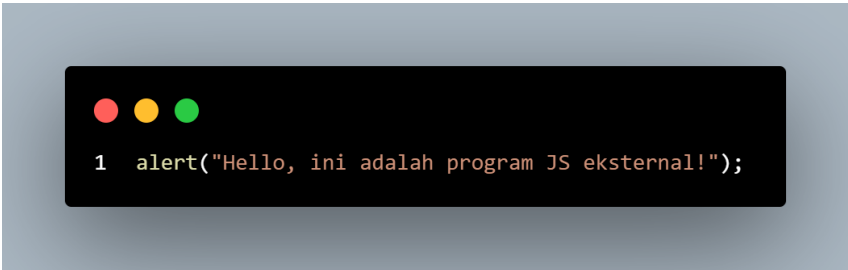

Placing JavaScript scripts within the `<body>` tag is, in my opinion, better than placing them within the `<head>` tag. When placed in `<head>`, the browser will execute the script before loading other HTML content, which can delay page rendering, especially if the script is large and complex. This can cause users to wait longer to view the page content. Conversely, placing the script at the bottom of the `<body>` tag allows the browser to render the entire page content first. That way, users can interact with the page faster, while the script is executed later.


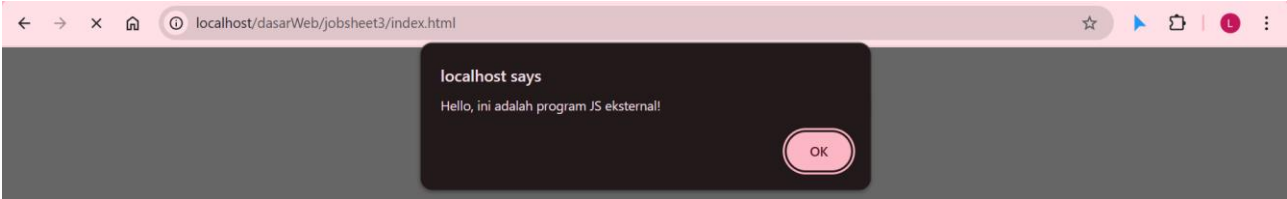
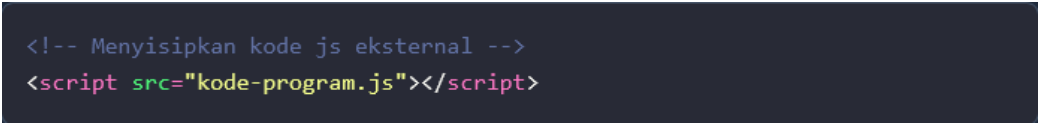
2. Inline Javascript Code Writing

Step	Description
1	In this way, we'll write the javascript code inside the HTML attribute. This method is usually used to call a function on a specific event. One example is when clicked.
2	Type the program code below: <pre>Klik aku!</pre> Or it can also be like this: <pre>Klik aku!</pre>

	 <pre> 1 <body> 2 <p>Tutorial Javascript untuk Pemula</p> 3 Klik aku! 4 <script> 5 //Ini adalah penulisan kode javascript 6 //di dalam tag <body> 7 console.log("Hello JS dari body") 8 </script> 9 </body> 10 </pre>
3	<p>Observe what happens to the browser! Record your observations (Question No. 7)</p>  <p>The web page will display a link that says “Click me!”. When the user clicks on the “Click me!” link, a pop-up dialog will appear with the message “Yey!”.</p>
4	<p>What is the difference between the two program codes (Question No. 8)</p>  <pre> 1 Klik aku! 2 Klik aku! 3 </pre> <p>Both lines of code produce the same effect, which is to display a pop-up alert when the link is clicked, but they use different methods, namely:</p> <ol style="list-style-type: none"> 1. First line: uses the onclick attribute, which is a standard HTML event handler. When the user clicks this element, the JavaScript code inside the quotation marks (alert(‘Yey!’)) will be executed. 2. Second line: uses the JavaScript pseudo-protocol within the href attribute. This is an older method and is not recommended. When this link is clicked, the browser will execute the JavaScript code. Although it works, this method is considered not in line with best practices.

3. External Javascript Code Writing

Step	Description
1	<p>In this way, we'll write the javascript code separately from the HTML file.</p> <p>In this method, we will write JavaScript code separately from the HTML file. This approach is usually recommended for large projects, as it makes project code management easier.</p>
2	<p>Let's try, create two files, namely HTML and Javascript files.</p> 
3	<p>Contents of the <code>kode-program.js</code> file :</p>  
4	<p>Contents of the <code>index.html</code> file :</p> 

	 <pre> 1 <!DOCTYPE html> 2 <html lang="en"> 3 <head> 4 <title>Belajar Javascript dari Nol</title> 5 </head> 6 <body> 7 <p>Tutorial Javascript untuk Pemula</p> 8 9 <!-- Menyisipkan kode js eksternal --> 10 <script src="kode-program.js"></script> 11 </body> 12 </html> </pre>
5	<p>Observe what happens to the browser! Record your observations</p> <p>(Question No. 9)</p>  <p>When index.html is opened in a browser, the external script kode-program.js will be loaded and executed, and the browser will display a pop-up dialog with a message as shown in the image. Before the pop-up is closed, the webpage will continue to load until we close the pop-out that appears. After the pop-up is closed, the browser will continue loading the rest of the page, which displays the text “JavaScript Tutorial for Beginners” inside the `<p>` tag. So, the first thing you see is the pop-up, then the text on the page.</p>
6	<p>In the experiment, we wrote separate javascript code with HTML code.</p> <p>Then in the HTML code we insert the src attribute in the <script> tag</p>  <pre> <!-- Menyisipkan kode js eksternal --> <script src="kode-program.js"></script> </pre> <p>Then anything in kode-program.js file will be readable from index.html file</p>
7	<p>What would happen if the javascript file was in a different folder?</p> <p>Observe and record your observations</p> <p>(Question No. 10)</p> <p>If the JS file is located in a different folder, the pop-out display will not appear in the browser because the external js file is not detected. The <script src="..."> tag instructs the browser to retrieve the script file from the specified location. If the browser cannot find the file at that location, no code will be</p>


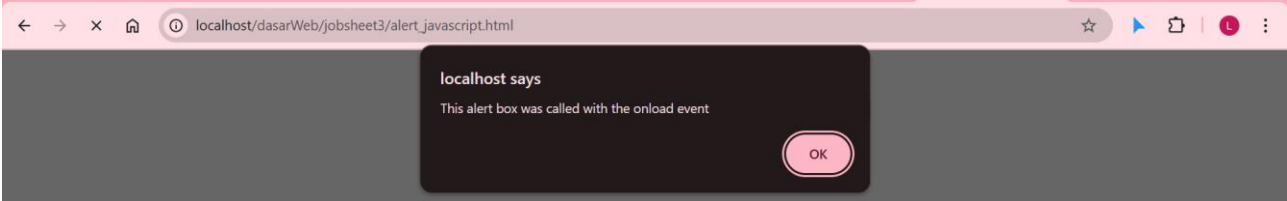
	executed. As a result, the <code>alert()</code> command in the file will never be executed, so the pop-up will not appear.
8	<p>Suppose we have a folder structure like this:</p> <pre> belajar-js/ ├── js/ │ ├── kode-program.js │ └── index.html </pre> <p>So to insert the <code>kode-program.js</code> file into the HTML, we can write the following code:</p> <pre> <script src="js/kode-program.js"></script> </pre> <p>Because the <code>kode-program.js</code> file is in the <code>js</code> directory.</p> <p>We can also insert javascript that exists on the internet by providing the full URL address.</p> <p>Example:</p> <pre> <script src="https://www.petanikode.com/js/kode.js"></script> </pre>

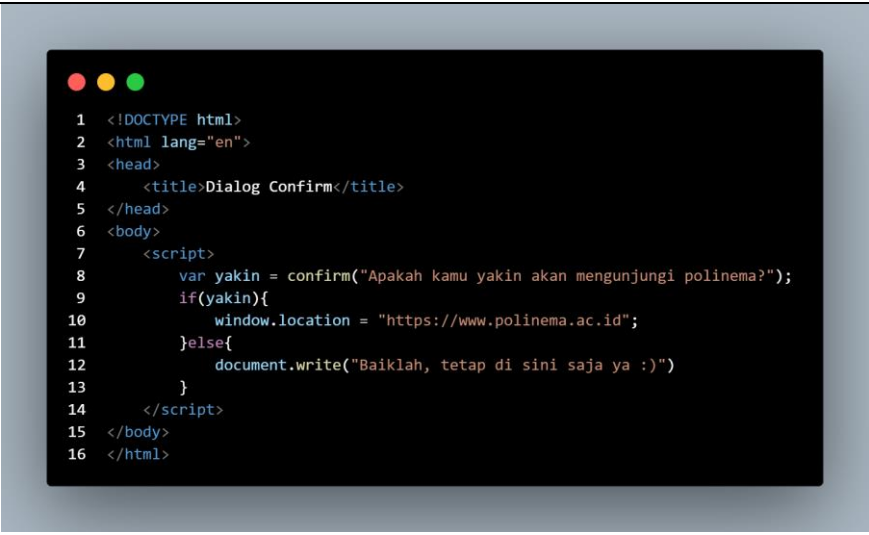


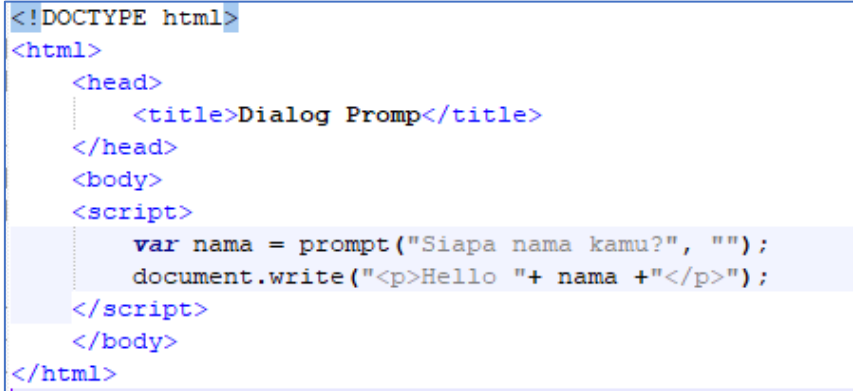
Practical Section 4: Dialogue Window

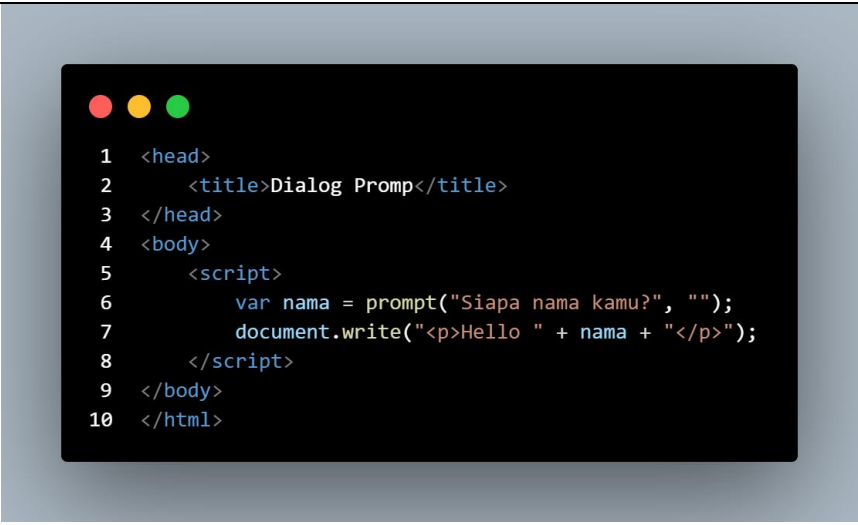
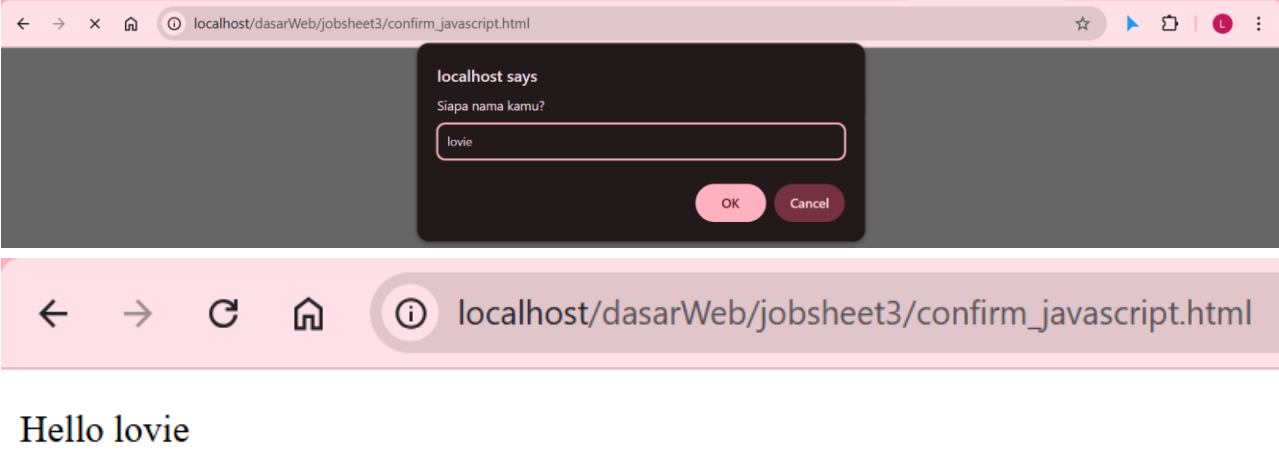
A dialog window is a window used to interact with users. There are three types of dialog windows in Javascript:

1. The `alert()` dialog window;
2. The `confirm()` dialog window;
3. The prompt dialog `window()` ;

Step	Description
1	Create a new file <code>alert_javascript.html</code> and save it in the project folder.
2	Type the program code below
	<pre> <html> <head> <script type="text/javascript"> function message() { alert("This alert box was called with the onload event") } </script> </head> <body onload="message()"> </body> </html> </pre>

	 <pre> 1 <!DOCTYPE html> 2 <html lang="en"> 3 <head> 4 <script type="text/javascript"> 5 function message(){ 6 alert('This alert box was called with the onload event') 7 } 8 </script> 9 </head> 10 <body onload="message()"> 11 </body> 12 </html> </pre>
3	<p>Observe what appears on the browser</p> 
4	<p>Record your observations (Question No. 11)</p> <p>A pop-out window appears as specified in the message code in the HTML file containing the alert script. After the pop-up window is closed, the browser display is blank because there is no body content in the HTML file.</p>
5	<p>Create a new file named <code>confirm_javascript.html</code> and save it in the project folder</p>
6	<p>Type the program code below</p> <pre> <!DOCTYPE html> <html> <head> <title>Dialog Confirm</title> </head> <body> <script> var yakin = confirm("Apakah kamu yakin akan mengunjungi polinema?"); if (yakin) { window.location = "https://www.polinema.ac.id"; } else { document.write("Baiklah, tetap di sini saja ya :)"); } </script> </body> </html> </pre>

	 <pre> 1 <!DOCTYPE html> 2 <html lang="en"> 3 <head> 4 <title>Dialog Confirm</title> 5 </head> 6 <body> 7 <script> 8 var yakin = confirm("Apakah kamu yakin akan mengunjungi polinema?"); 9 if(yakin){ 10 window.location = "https://www.polinema.ac.id"; 11 }else{ 12 document.write("Baiklah, tetap di sini saja ya :)") 13 } 14 </script> 15 </body> 16 </html> </pre>
7	<p>Observe what appears on the browser</p> <p>If we visit polinema website</p>  <p>If we close the pop-out</p> 
8	<p>Record your observations</p> <p>(Question No. 12)</p> <p>If the user clicks "OK" then the JavaScript if condition is met. The browser will then be redirected to the URL https://www.polinema.ac.id, navigating the user to the Polinema website. If the user clicks "Cancel" The else block of the code will execute. Instead of redirecting, the browser will display the text "Baiklah, tetap di sini saja ya :)" on the page.</p>
9	<p>Type the program code below</p>  <pre> <!DOCTYPE html> <html> <head> <title>Dialog Promp</title> </head> <body> <script> var nama = prompt("Siapa nama kamu?", ""); document.write("<p>Hello " + nama + "</p>"); </script> </body> </html> </pre>

	 <pre> 1 <head> 2 <title>Dialog Promp</title> 3 </head> 4 <body> 5 <script> 6 var nama = prompt("Siapa nama kamu?", ""); 7 document.write("<p>Hello " + nama + "</p>"); 8 </script> 9 </body> 10 </html> </pre>
10	<p>Observe what appears on the browser</p>  <p>localhost says Siapa nama kamu? lovie OK Cancel</p> <p>← → ↻ 🏠 ⓘ localhost/dasarWeb/jobsheet3/confirm_javascript.html</p> <p>Hello lovie</p>
11	<p>Record your observations (Question No. 13)</p> <p>When this HTML file is opened, a prompt dialog box will appear, asking, "Siapa nama kamu?" It also includes an input field for the user to type their name. If the user types a name and clicks "OK", the variable nama will store the text that was entered. The document.write() command will then display "Hello" followed by the entered name on the web page. For example, if the user types "Lovie" the page will show "Hello Lovie". If the user clicks "Cancel" or leaves the input field empty and clicks "OK", the nama variable will be set to null or an empty string. The page will then display "Hello" with no name.</p>

Practical Section 5: Variables


The way to create a variable that is commonly used in javascript is to use the var keyword followed by the name of the variable and its value.

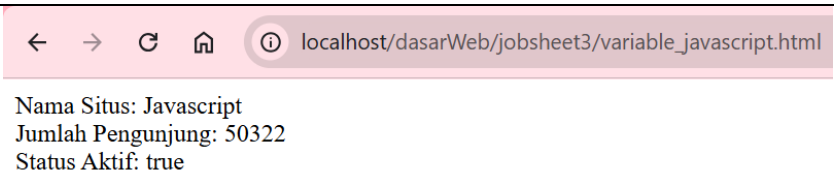
Example: **var title = "Learn Javascript Programming";**

Displaying the contents of a Variable

To display the contents of the variables, we can utilize functions to display outputs such as:

- The `console.log()` function returns the output to the javascript console;
- The `document.write()` function returns the output to an HTML document;
- and the `alert()` function returns the output to the dialog window.

Step	Description
1	Create a new file <code>variable_javascript.html</code> and save it in the project folder.
2	Type the program code below
	<pre><!DOCTYPE html> <html lang="en"> <head> <title>Belajar Variabel dalam Javascript</title> <script> // membuat variabel var name = "Javascript"; var visitorCount = 50322; var isActive = true; // menampilkan variabel ke jendela dialog (alert) alert("Selamat datang di " + name); // menampilkan variabel ke dalam HTML document.write("Nama Situs: " + name + "
"); document.write("Jumlah Pengunjung: " + visitorCount + "
"); document.write("Status Aktif: " + isActive + "
"); </script> </head> <body> </body> </html></pre> 
3	Observe what appears on the browser

	
4	<p>Record your observations</p> <p>(Question No. 14)</p> <p>A dialog box with the message "Selamat datang di Javascript" will first appear. After the user clicks "OK", the web page will display three lines of text. This happens because the JavaScript code declares and initializes three variables, a string (name), a number (visitorCount), and a boolean (isActive). It then uses document.write() to output the label and the value of each variable directly to the web page, with
 tags ensuring each piece of information appears on a new line.</p>

Deleting Variables

In JavaScript, deleting variables is uncommon. However, in programs where careful memory management is crucial, removing variables is important to ensure more efficient memory usage. This can be achieved using the `delete` keyword.

Example:


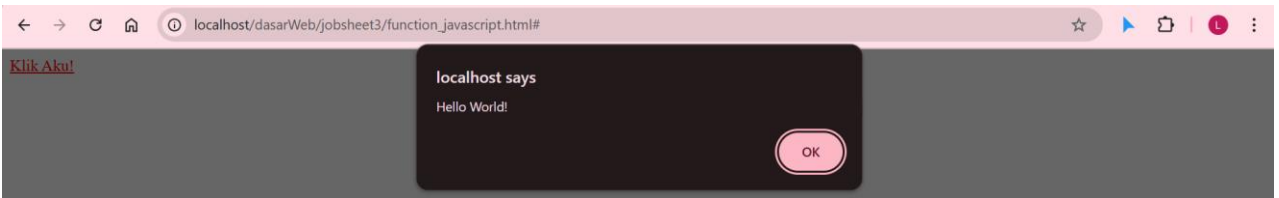
```
var bookTitle = "Learn Javascript Programming";
delete bookTitle;
```

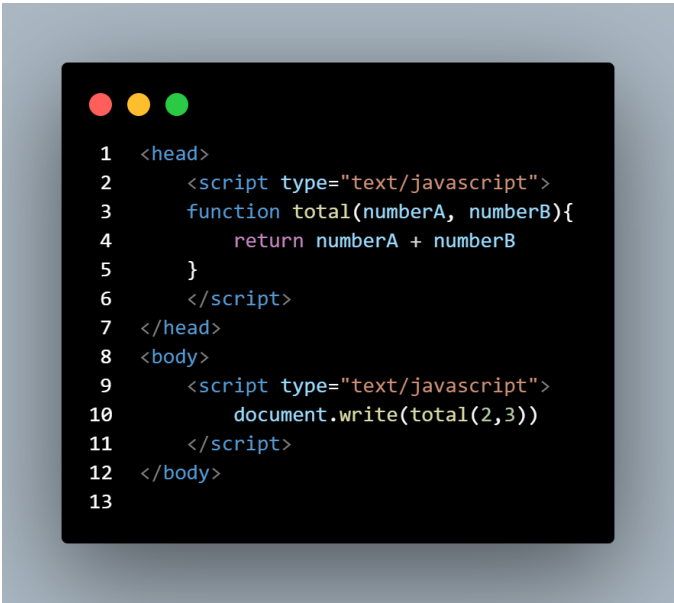
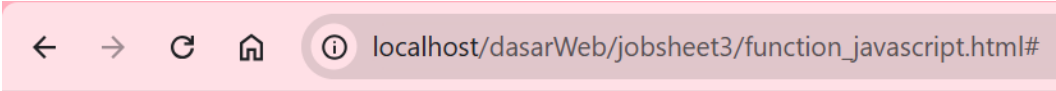
Then the `bookTitle` variable will disappear from memory.

Practical Section 6 : Functions

Functions are sub-programs that can be reused both within the program itself, and in other programs. A function in Javascript is an object. Because it has properties and also *methods*.

Step	Description
1	Create a new File named <code>function_javascript.html</code> and save it in the project folder
2	How to call a function in Javascript code is usually written with: <code>functionName() ;</code>
3	Type the following program code

	<pre> <!DOCTYPE html> <html> <head> <script> // membuat fungsi var sayHello = () => alert("Hello World!"); </script> </head> <body> <!-- Memanggil fungsi saat link diklik --> Klik Aku! </body> </html> </pre> 
4	<p>Observe what appears in the browser</p> 
5	<p>Record your observations</p> <p>(Question No. 15)</p> <p>When loaded in a browser, the text “Click Me!” will appear as a link. When the user clicks on the link, a pop-up dialog will appear with the message “Hello World!”. This happens because the <code><script></code> tag defines a JavaScript function called <code>sayHello()</code> that contains the <code>alert()</code> command. Then, the <code>onclick</code> attribute on the link (<code><a></code>) calls the <code>sayHello()</code> function every time the link is clicked, which triggers the pop-up to appear.</p>
6	<p>A parameter is a variable that stores a value for a process inside a function.</p> <p>How to call a parameter in javascript is:</p>

	<pre>function kali(a, b){ hasilKali = a * b; console.log("Hasil kali a*b = " + hasilKali); }</pre>
7	<p>Type the following program code</p> <pre><html> <head> <script type="text/javascript"> function total(numberA,numberB) { return numberA + numberB } </script> </head> <body> <script type="text/javascript"> document.write(total(2,3)) </script> </body> </html></pre> 
8	<p>Observe what appears in the browser</p>  <p>5</p>
9	<p>Record your observations (Question No. 16)</p>

	The code defines a JavaScript function named total that takes two arguments, numberA and numberB, and returns the sum of these two numbers. In the <body> of the HTML, another script block calls this function with the arguments 2 and 3. The document.write() command then takes the result of that function call, which is 5, and outputs it directly to the web page.
--	--

Practical Section 7: Data Types

Data types are the types of data that we can store in variables. There are several types of data in Javascript programming:

- String (text)
- Integer or Number
- Float (number of Fractions)
- Boolean
- Object

Javascript is a *dynamic typing* language, which means that we don't have to write data types when creating variables like in [C](#), [C++](#), [Java](#), etc. which are *static typing*. There are several rules for writing variables in Javascript:

- Variable naming **should not** use numbers in front of it.
example:

```
// wrong
var 123name = "Polinema";

// right
var name123 = "Polinema";
```

- Variable naming **can** use the initial underscore.
example:

```
var _nama = "Polinema";
```


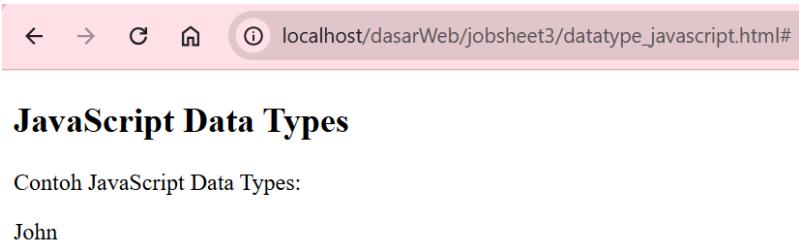
- Variable naming **is recommended** using **camelCase** if it consists of two syllables.
Example:


```
var _fullName = "Polinema";
```

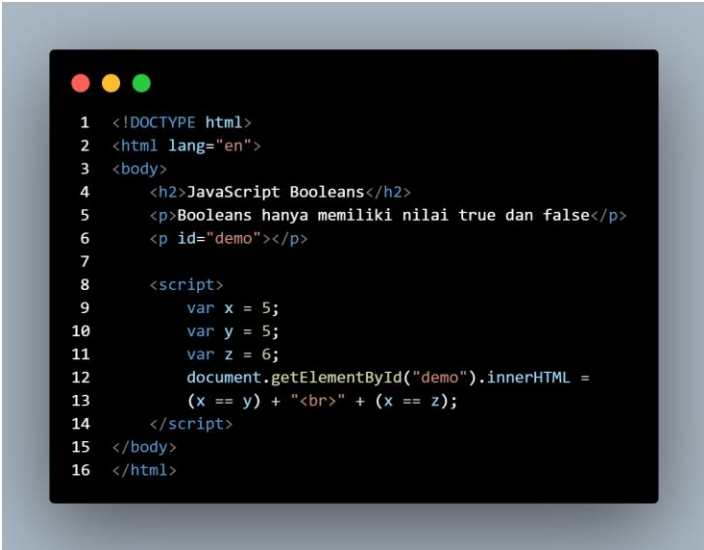
- Variable naming **is recommended** using English
Example:


```
var _postTitle = "Javascript Tutorials";
```


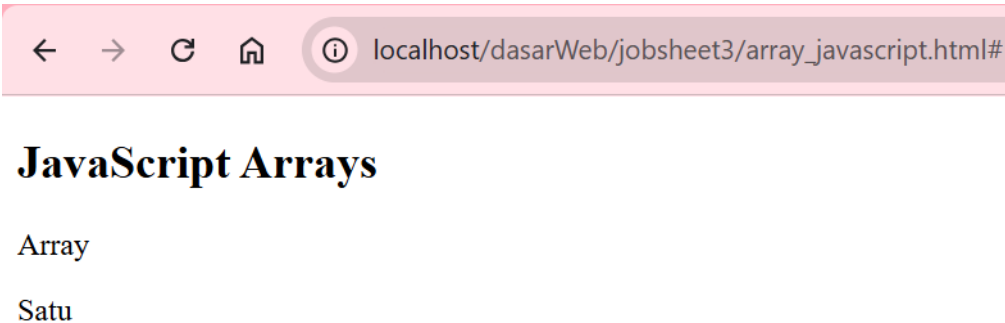
Step	Description
------	-------------

1	Create a new File named <code>datatype_javascript.html</code> and save it in the project folder.
2	<p>Type the following program code</p> <pre> <!DOCTYPE html> <html> <body> <h2>JavaScript Data Types</h2> <p>Contoh Javascript Data Types:</p> <p id="demo"></p> <script> var x; // Now x is undefined x = 5; // Now x is a Number x = "John"; // Now x is a String document.getElementById("demo").innerHTML = x; </script> </body> </html> </pre> 
3	<p>Observe what appears in the browser</p> 
4	<p>Record your observations</p> <p>(Question No. 17)</p>

	<p>The web page display the text "JavaScript Data Types" as a heading and "Contoh JavaScript Data Types:" as a paragraph. Below that, the name "John" will appear. This happens because the JavaScript code first declares a variable x. It then assigns the number 5 to x, but immediately reassigns the string "John" to it. The final line of the script finds the HTML paragraph with the id of "demo" and sets its content to the last value of x, which is "John". The initial value of 5 is overwritten and never displayed.</p>
5	<p>Type the program below and save it with the <code>string_javascript.html</code> name</p> <pre> <!DOCTYPE html> <html> <body> <h2>JavaScript Strings</h2> <p>Membuat Javascript String</p> <p id="demo"></p> <script> var answer1 = "It's alright"; var answer2 = "He is called 'Dilan'"; var answer3 = 'He is called "Dilan"'; document.getElementById("demo").innerHTML = answer1 + "
" + answer2 + "
" + answer3; </script> </body> </html> </pre> 
6	<p>Observe what appears in the browser</p>

	<div> <div> ← → ↺ 🏠 🔍 localhost/dasarWeb/jobsheet3/string_javascript.html# </div> <div> <h2>JavaScript String</h2> <p>Membuat JavaScript String</p> <p>It's alright He is called 'Dilan' He is called 'Dilan'</p> </div> </div>
7	<p>Record your observations</p> <p>(Question No. 18)</p> <p>The browser display a heading "JavaScript String" and a paragraph "Membuat Javascript String". Below that, the following three lines of text appear, each on a new line.</p>
8	<p>Type the program below and save it with the <code>boolean_javascript.html</code> name</p> <pre> <!DOCTYPE html> <html> <body> <h2>JavaScript Booleans</h2> <p>Booleans hanya memiliki nilai true dan false</p> <p id="demo"></p> <script> var x = 5; var y = 5; var z = 6; document.getElementById("demo").innerHTML = (x == y) + "
" + (x == z); </script> </body> </html> </pre> 
9	<p>Observe what appears in the browser</p>

	
10	<p>Record your observations</p> <p>(Question No. 19)</p> <p>The web browser display a heading "JavaScript Booleans" and a paragraph "Booleans hanya memiliki nilai true dan false". This happens because the JavaScript code compares the values of three variables. The expression (x == y) evaluates to true because both x and y have a value of 5. The second expression, (x == z), evaluates to false because x is 5 and z is 6. The document.getElementById("demo").innerHTML command then prints the results of these boolean evaluations to the paragraph with the ID "demo", separated by a line break
.</p>
11	<p>Type the program below and save it with the <code>array_javascript.html</code> name</p> <pre data-bbox="293 877 1010 1396"><!DOCTYPE html> <html> <body> <h2>JavaScript Arrays</h2> <p>Array</p> <p id="demo"></p> <script> var cars = ["Satu", "Dua", "Tiga"]; document.getElementById("demo").innerHTML = cars[0]; </script> </body> </html></pre>

	 <pre> 1 <!DOCTYPE html> 2 <html lang="en"> 3 <body> 4 <h2>JavaScript Arrays</h2> 5 <p>Array</p> 6 <p id="demo"></p> 7 <script> 8 var cars = ["Satu", "Dua", "Tiga"]; 9 document.getElementById("demo").innerHTML = cars[0]; 10 </script> 11 </body> 12 </html> </pre>
12	<p>Observe what appears in the browser</p>  <p>JavaScript Arrays</p> <p>Array</p> <p>Satu</p>
13	<p>Record your observations</p> <p>(Question No.20)</p> <p>The text "JavaScript Arrays" and "Array" will appear on the web page, followed by the word "Satu". The text "JavaScript Arrays" and "Array" will appear on the web page, followed by the word "Satu". The JavaScript code first creates an array named cars containing three string values: "Satu", "Dua", and "Tiga". Arrays in JavaScript use a zero-based index, meaning the first item is at index 0, the second at index 1, and so on. The line <code>document.getElementById("demo").innerHTML = cars[0];</code> targets the HTML paragraph with the id of "demo" and sets its content to the value of the first element in the cars array, which is "Satu".</p>

Practical Section 8: Operator

An operator is a symbol used to perform operations on a value and variable. Operators in programming are divided into 6 types:

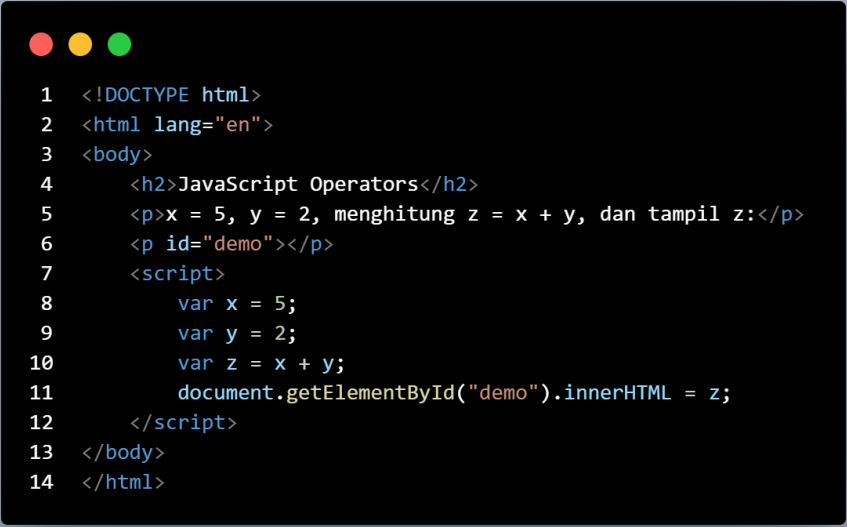

1. Arithmetic operator;

2. Assignment Operator;
3. relationship or comparison operators;
4. Logic Operators;
5. Bitwise Operator;
6. Ternary Operator;

An arithmetic operator is an operator to perform arithmetic operations such as addition, subtraction, division, multiplication, etc. Arithmetic operators consist of:

Operator Name	Symbol
Addition	+
Reduction	-
Multiplication	*
Appointment	**
Division	/
Leftover	%

Step	Description
1	Create a new File named <code>operator_javascript.html</code> and save it in the project folder.
2	<p>Type the program below</p> <pre> <!DOCTYPE html> <html> <body> <h2>JavaScript Operators</h2> <p>x = 5, y = 2, menghitung z = x + y, dan tampil z:</p> <p id="demo"></p> <script> var x = 5; var y = 2; var z = x + y; document.getElementById("demo").innerHTML = z; </script> </body> </html> </pre>

	 <pre> 1 <!DOCTYPE html> 2 <html lang="en"> 3 <body> 4 <h2>JavaScript Operators</h2> 5 <p>x = 5, y = 2, menghitung z = x + y, dan tampil z:</p> 6 <p id="demo"></p> 7 <script> 8 var x = 5; 9 var y = 2; 10 var z = x + y; 11 document.getElementById("demo").innerHTML = z; 12 </script> 13 </body> 14 </html> </pre>
3	<p>Observe what appears in the browser</p>  <p>JavaScript Operators</p> <p>x = 5, y = 2, menghitung z = x + y, dan tampil z:</p> <p>7</p>
4	<p>Record your observations (Question No. 21)</p> <p>The HTML code first sets up a heading and a paragraph explaining the calculation. The embedded JavaScript then declares three variables. It assigns the number 5 to x and the number 2 to y. The variable z is then assigned the result of x + y, which is 7. Finally, the line <code>document.getElementById("demo").innerHTML = z;</code> finds the HTML paragraph with the ID "demo" and replaces its content with the value of z. Therefore, the number 7 is the final output displayed to the user.</p>

Practical Section 9: Branching

It can be said that branching and looping are one of the core methods in all programming languages worldwide. With branching and looping, a dynamic program can be created instead of a linear and static one. Since JavaScript is a method for client-side web programming, it also has this capability.

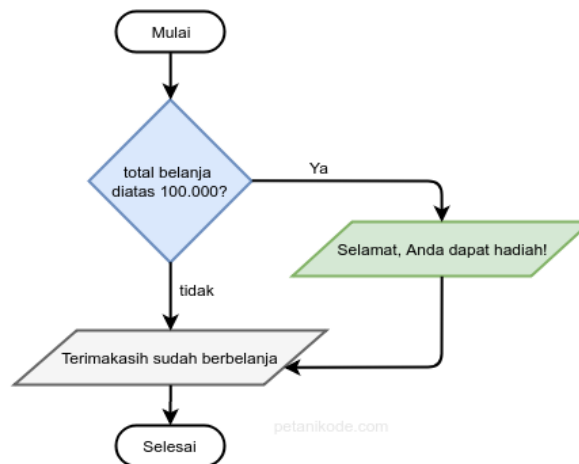
Some branching functions include:

- Use `if` to specify a block of code to be executed, if a specified condition is true
- Use `else` to specify a block of code to be executed, if the same condition is false

- Use else if to specify a new condition to test, if the first condition is false
- Use switch to specify many alternative blocks of code to be executed

❖ if Branching

if branching is a structure that only has one block of choice when the condition is true. Take a look at the following flowchart:



“If the total purchase is greater than Rp 100,000, then display the message: Congratulations, you won a prize.”

What if it is below Rp 100,000?

Yes, the message is not displayed.

Step	Description
1	Create a new File named <code>if_javascript.html</code> and save it in the project folder
2	Type the program below

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Percabangan if</title>
</head>
<body>
  <script>
    var totalBelanja = prompt("Total belanja?", 0);

    if(totalBelanja > 30000){
      document.write("<h2>Selamat Anda dapat hadiah</h2>");
    }

    document.write("<p>Terimakasih sudah berbelanja di toko kami</p>");
  </script>
</body>
</html>

```

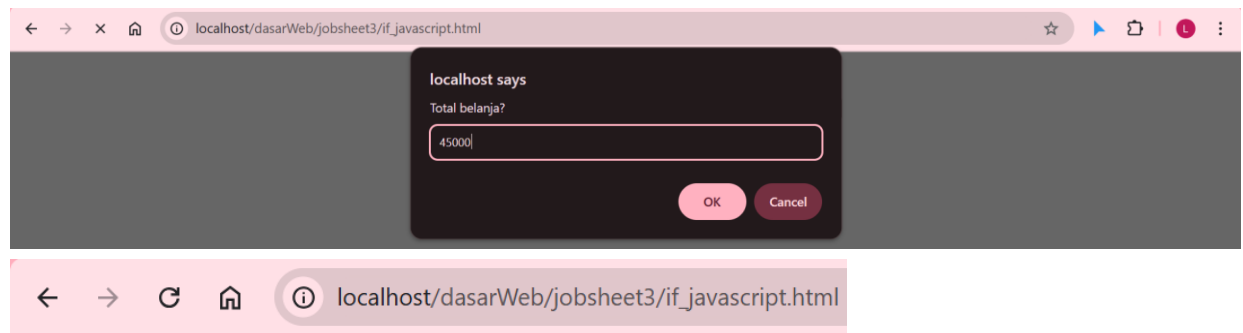
```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <title>Percabangan if</title>
5  </head>
6  <body>
7    <script>
8      var totalBelanja = prompt("Total belanja?", 0);
9      if(totalBelanja > 30000){
10        document.write("<h2>Selamat Anda dapat hadiah</h2>");
11      }
12      document.write("<p>Terimakasih sudah berbelanja di toko kami</p>")
13    </script>
14  </body>
15  </html>

```

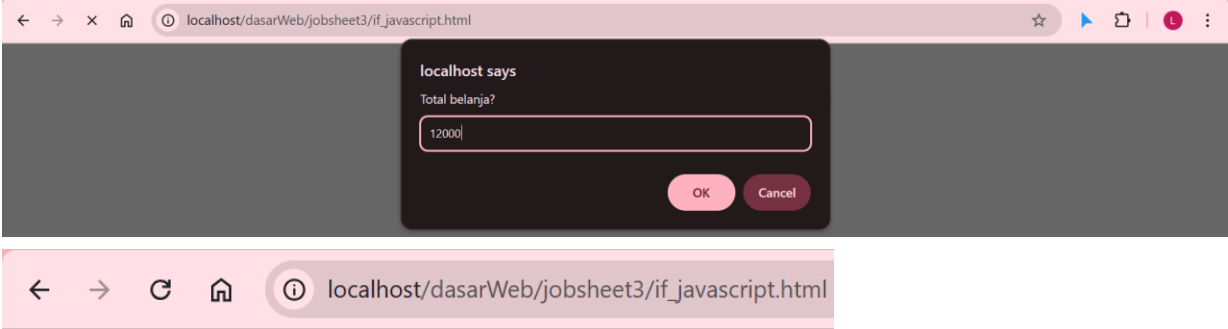
3

Observe what appears in the browser



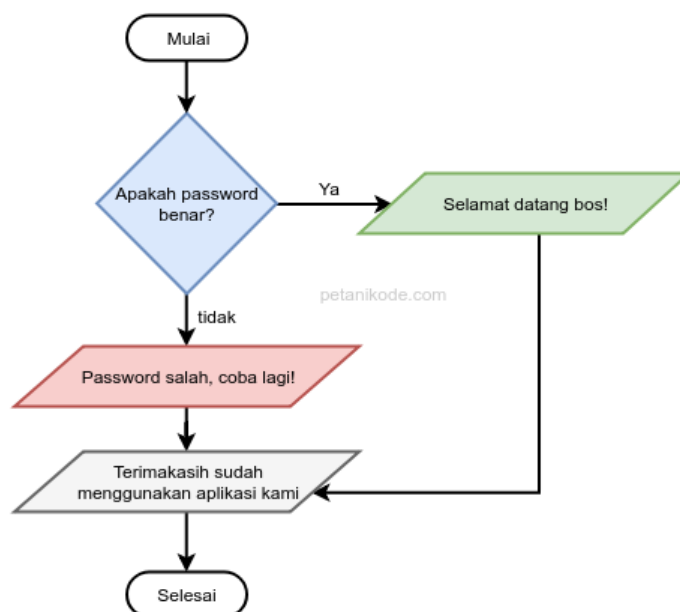
Selamat Anda dapat hadiah

Terimakasih sudah berbelanja di toko kami

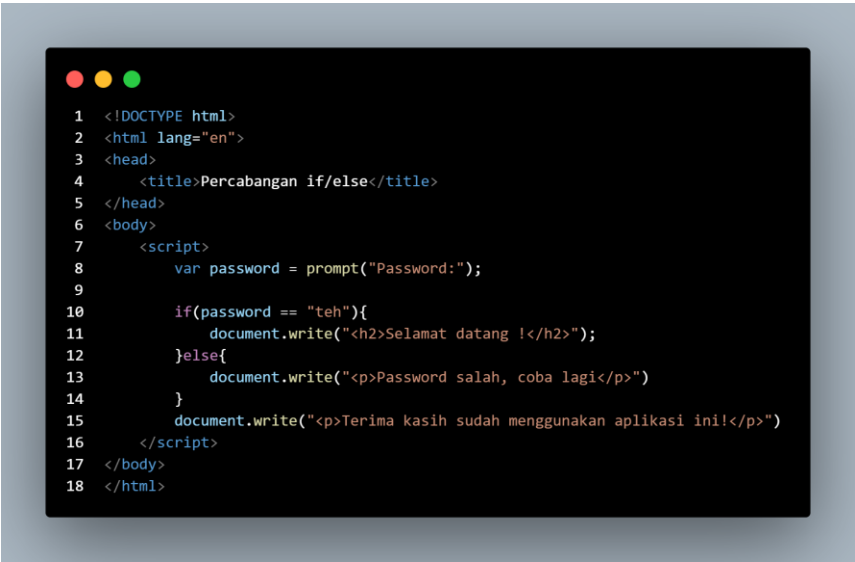
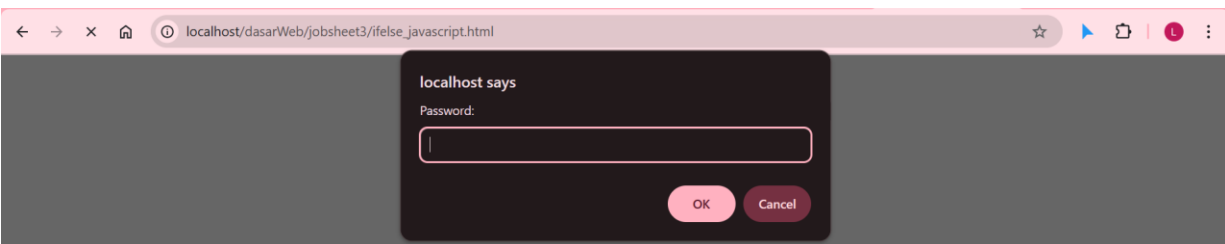
	 <p>Terimakasih sudah berbelanja di toko kami</p>
4	<p>Record your observations (Question No. 22)</p> <p>First, a prompt dialog box will appear asking the user to input their total spending with the message "Total belanja?". If the user enters a number greater than 30000 and clicks OK, the phrase "Selamat Anda dapat hadiah!" will appear on the screen as a heading (<h2>). Below that, the text "Terimakasih sudah berbelanja di toko kami" will be displayed as a paragraph (<p>). But if the user enters a number less than or equal to 30000 (e.g., 25000), or clicks "Cancel" or enters nothing, only the paragraph "Terimakasih sudah berbelanja di toko kami" will be displayed. The if condition is not met, so the heading about the gift is skipped.</p>

❖ if/else Branching

If/Else Branching is a structure that has **two blocks of choices**. The first choice is for when the **condition is true**, and the second choice is for when the **condition is false (else)**. Take a look at this flowchart:



This is a flowchart for checking the password. If the password is correct, the message in the green block will be displayed: **"Welcome, boss!"** However, if it is incorrect, the message in the red block will be shown: **"Incorrect password, please try again!"**

Step	Description
1	Create a new File named <code>ifelse_javascript.html</code> and save it in the project folder.
2	<p>Type the program below</p> <pre> <!DOCTYPE html> <html> <head> <title>Percabangan if/else</title> </head> <body> <script> var password = prompt("Password:"); if(password == "teh"){ document.write("<h2>Selamat datang !</h2>"); } else { document.write("<p>Password salah, coba lagi!</p>"); } document.write("<p>Terima kasih sudah menggunakan aplikasi ini!</p>"); </script> </body> </html> </pre> 
3	<p>Observe what appears in the browser</p> 

	<p>Selamat datang !</p> <p>Terima kasih sudah menggunakan aplikasi ini!</p> <p>Password salah, coba lagi</p> <p>Terima kasih sudah menggunakan aplikasi ini!</p>
4	<p>Record your observations</p> <p>(Question No. 23)</p> <p>Upon opening the file in a browser, a prompt dialog box will appear, asking for a password. If the user types "teh" and clicks OK, the page will display "Selamat datang !" as a heading. Below that, it will show the final message, "Terima kasih sudah menggunakan aplikasi ini!" as a paragraph. But if the user types anything else or nothing at all and clicks OK, the page will display "Password salah, coba lagi" as a paragraph. Below that, it will still show the final message, "Terima kasih sudah menggunakan aplikasi ini!".</p>

❖ switch/case Branching

switch/case branching is an alternative form of the **if/else/if** branching structure. In a **switch/case** statement, instead of evaluating multiple if conditions, the program evaluates the value of a variable or expression and compares it against multiple possible cases. Each case represents a potential value, and when a match is found, the corresponding block of code is executed. If no case matches, the **default** case is executed (if provided), similar to the **else** block in **if/else** statements.

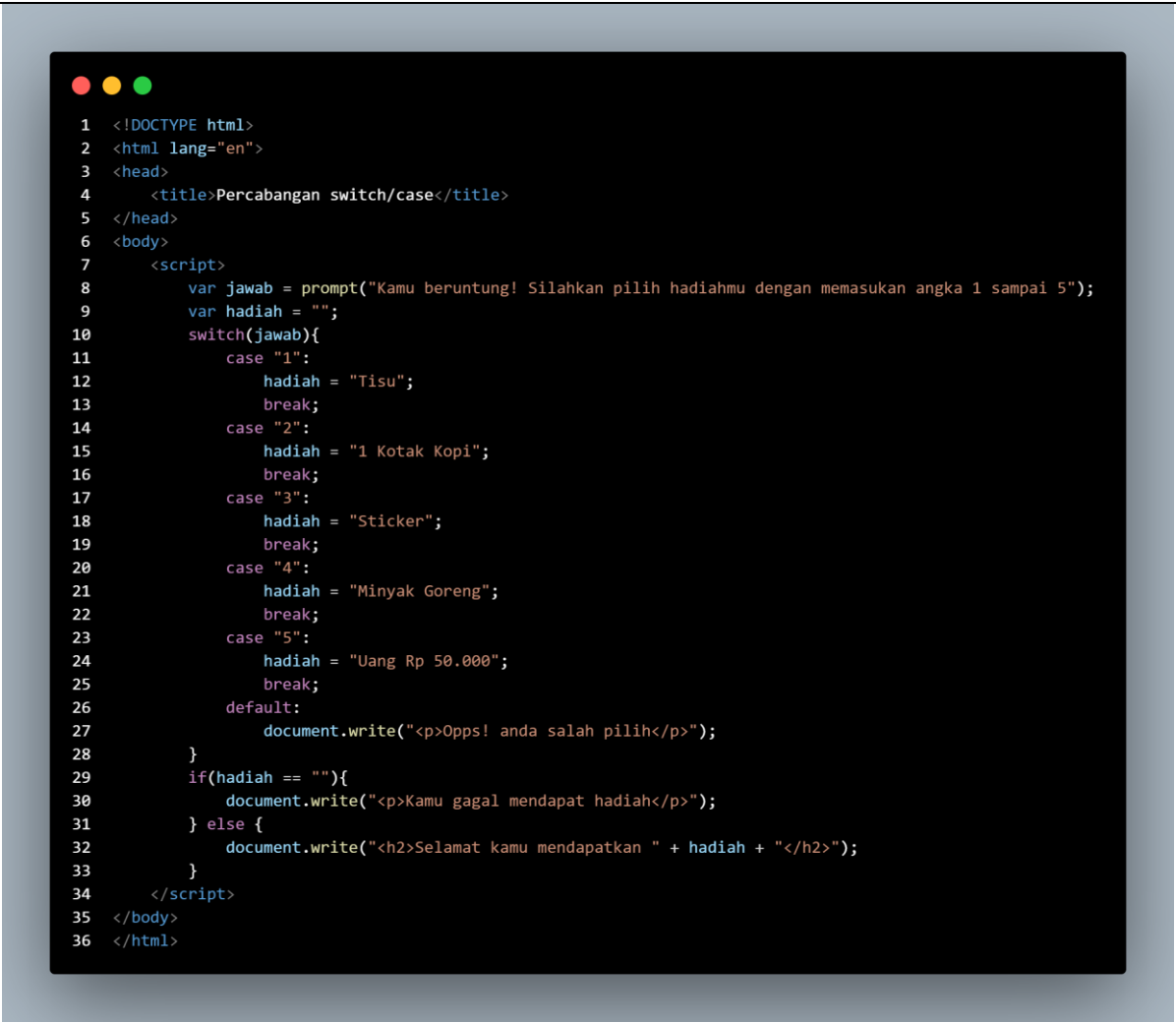

The **switch/case** structure can make code more readable and organized, especially when dealing with multiple conditions based on a single variable. The structure looks like this:

```

switch(variabel){
    case <value>:
        // blok kode
        break;
    case <value>:
        // blok kode
        break;
    default:
        // blok kode
}

```

Step	Description
1	Create a new File named <code>switchcase_javascript.html</code> and save it in the project folder
2	Type the program below <div> <pre> <!DOCTYPE html> <html> <head> <title>Percabangan switch/case</title> </head> <body> <script> var jawab = prompt("Kamu beruntung! Silahakn pilih hadiahmu dengan memasukan angka 1 sampai 5"); var hadiah = ""; switch(jawab){ case "1": hadiah = "Tisu"; break; case "2": hadiah = "1 Kotak Kopi"; break; case "3": hadiah = "Sticker"; break; case "4": hadiah = "Minyak Goreng"; break; case "5": hadiah = "Uang Rp 50.000"; break; default: document.write("<p>Oops! anda salah pilih</p>"); } if(hadiah == ""){ document.write("<p>Kamu gagal mendapat hadiah</p>"); } else { document.write("<h2>Selamat kamu mendapatkan " + hadiah + "</h2>"); } </script> </body> </html> </pre> </div>

	 <pre> 1 <!DOCTYPE html> 2 <html lang="en"> 3 <head> 4 <title>Percabangan switch/case</title> 5 </head> 6 <body> 7 <script> 8 var jawab = prompt("Kamu beruntung! Silahkan pilih hadiahmu dengan memasukan angka 1 sampai 5"); 9 var hadiah = ""; 10 switch(jawab){ 11 case "1": 12 hadiah = "Tisu"; 13 break; 14 case "2": 15 hadiah = "1 Kotak Kopi"; 16 break; 17 case "3": 18 hadiah = "Sticker"; 19 break; 20 case "4": 21 hadiah = "Minyak Goreng"; 22 break; 23 case "5": 24 hadiah = "Uang Rp 50.000"; 25 break; 26 default: 27 document.write("<p>Oops! anda salah pilih</p>"); 28 } 29 if(hadiah == ""){ 30 document.write("<p>Kamu gagal mendapat hadiah</p>"); 31 } else { 32 document.write("<h2>Selamat kamu mendapatkan " + hadiah + "</h2>"); 33 } 34 </script> 35 </body> 36 </html> </pre>
3	<p>Observe what appears in the browser</p>  <p>The browser window shows a prompt dialog box with the text: "localhost says", "Kamu beruntung! Silahkan pilih hadiahmu dengan memasukan angka 1 sampai 5", and an input field. Below the dialog box, the browser address bar shows "localhost/dasarWeb/jobsheet3/switchcase_javascript.html". The main content area displays the text: "Selamat kamu mendapatkan Tisu".</p>
4	<p>Record your observations (Question No. 24)</p>


	A prompt dialog box will appear with the message: “You're lucky! Please choose your prize by entering a number from 1 to 5.” If the user enters 1, 2, 3, 4, or 5, the page will display a heading with the message “Congratulations, you have won [Prize Name].” The gift name will correspond to the number entered (for example, “Tissues” for number 1, “1 Box of Coffee” for number 2, etc.). If the user enters another number (or text, or cancels), the page will display two paragraphs. The first paragraph reads “Oops! You chose the wrong number” and the second paragraph reads “You failed to get a prize”.
--	---

❖ Nested Branching

Nested Branching refers to a condition where one branching statement (such as **if**, **else**, **switch**, etc.) is placed inside another branching statement. This allows for more complex decision-making processes where multiple conditions need to be evaluated at different levels. In nested branching, the outcome of one condition can depend on the result of another, providing more fine-grained control over the program flow.

For example, you can nest an **if** statement inside another **if** statement to first check one condition and then, based on that, check a second condition within the first block.

Step	Description
1	Create a new File named nestedif_javascript.html and save it in the project folder
2	Type the program below <pre> <!DOCTYPE html> <html> <head> <title>Percabangan Ternary</title> </head> <body> <script> var username = prompt("Username:"); var password = prompt("Password:"); if(username == "mahasiswa"){ if(password == "kopi"){ document.write("<h2>Selamat datang </h2>"); } else { document.write("<p>Password salah, coba lagi!</p>"); } } else { document.write("<p>Anda tidak terdaftar!</p>"); } </script> </body> </html> </pre>

	 <pre> 1 <!DOCTYPE html> 2 <html lang="en"> 3 <head> 4 <title>Percabangan Ternary</title> 5 </head> 6 <body> 7 <script> 8 var username = prompt("Username:"); 9 var password = prompt("Password:"); 10 11 if(username == "mahasiswa"){ 12 if(password == "kopi"){ 13 document.write("<h2>Selamat datang </h2>"); 14 }else{ 15 document.write("<p>Password salah, coba lagi!</p>"); 16 } 17 }else{ 18 document.write("<p>Anda tidak terdaftar!</p>"); 19 } 20 </script> 21 </body> 22 </html> </pre>
3	<p>Observe what appears in the browser</p>   <p>Selamat datang</p>
4	<p>Record your observations</p> <p>(Question No. 25)</p> <p>When this code runs in the browser, it first asks the user to enter a username and password, if the username entered is "mahasiswa" and the password is "kopi", the page will show a large</p>

	heading saying "Selamat datang", but if the username is correct while the password is wrong it will display "Password salah, coba lagi!", and if the username itself is not "mahasiswa", regardless of the password, the page will simply show "Anda tidak terdaftar!".
--	---

Practical Section 10: Loops

Loops help us execute code repeatedly, as many times as we want. There are five types of loops in JavaScript. Generally, these loops are categorized into two types: counted loops and uncounted loops.

The difference is as follows:

- ✓ **Counted Loops** are loops where the number of iterations is **known** and **definite**.
- ✓ **Uncounted Loops**, on the other hand, are loops where the number of iterations is **not predetermined**.

The loops that fall under **Counted Loops** are:

1. For Loop
2. Foreach Loop
3. Repeat Loop

The loops that fall under **Uncounted Loops** are:



1. While Loop
2. Do/While Loop

❖ For loops in Javascript

A *for* loop is a loop that is included in a *counted loop*, because it is clear how many times it will repeat. It looks like this:

```
for(let i = 0; i < 10; i++){
    document.write("<p>Perulangan ke-" + i + "</p>")
}
```


Step	Description
1	Create a new File named <code>for_javascript.html</code> and save it in the project folder
2	Type the program below


	<pre><!DOCTYPE html> <html> <body> <h2>JavaScript Loops</h2> <p id="demo"></p> <script> var text = ""; var i; for (i = 0; i < 5; i++) { text += "The number is " + i + "
"; } document.getElementById("demo").innerHTML = text; </script> </body> </html></pre> 
3	<p>Observe what appears in the browser</p> 
4	<p>Record your observations</p> <p>(Question No. 26)</p> <p>When this code is run in the browser, it displays a heading “JavaScript Loops” followed by five lines of text generated by the loop, specifically “The number is 0”, “The number is 1”, “The</p>

	number is 2”, “The number is 3”, and “The number is 4”, each shown on a new line under the heading.
--	---

❖ While loops in Javascript

The while loop is categorized as an uncounted loop. However, the while loop can also function as a counted loop by including a counter within it.

Step	Description
1	Create a new File named <code>while_javascript.html</code> and save it in the project folder
2	<p>Type the program below</p> <pre> <!DOCTYPE html> <html> <body> <h2>JavaScript while</h2> <p id="demo"></p> <script> var text = ""; var i = 0; while (i < 10) { text += "
The number is " + i; i++; } document.getElementById("demo").innerHTML = text; </script> </body> </html> </pre> 
3	Observe what appears in the browser

	 <p>JavaScript while</p> <p>The number is 0 The number is 1 The number is 2 The number is 3 The number is 4 The number is 5 The number is 6 The number is 7 The number is 8 The number is 9</p>
4	<p>Record your observations</p> <p>(Question No. 27)</p> <p>When this code is run in the browser, it displays a heading “JavaScript While” followed by ten lines of text generated by the loop, specifically “The number is 0”, “The number is 1”, “The number is 2”, “The number is 3”, and “The number is 4” until “The number is 9”, each shown on a new line under the heading.</p>

❖ Do/While Loops in Javascript


The **do/while loop** is a variation of the while loop in JavaScript. The main difference between them is that the do/while loop will always execute the code inside the loop **at least once**, regardless of whether the condition is true or false. This is because the condition is evaluated **after** the code block is executed, not before, as in the standard while loop.:


```
do {
    // blok kode yang akan diulang
} while (<kondisi>);
```

Key Characteristics:

- The code inside the do block runs first, and then the condition is checked.
- If the condition is true, the loop repeats; if false, the loop stops.
- This type of loop ensures that the code inside the loop executes at least once, even if the condition is false from the beginning.

Step	Description
1	Create a new File named <code>dowhile_javascript.html</code> and save it in the project folder
2	Type the program below

	<pre> <!DOCTYPE html> <html> <body> <h2>JavaScript do ... while</h2> <p id="demo"></p> <script> var text = "" var i = 0; do { text += "
The number is " + i; i++; } while (i < 10); document.getElementById("demo").innerHTML = text; </script> </body> </html> </pre> 
3	Observe what appears in the browser

	 <p>JavaScript do ... while</p> <p>The number is 0 The number is 1 The number is 2 The number is 3 The number is 4 The number is 5 The number is 6 The number is 7 The number is 8 The number is 9</p>
4	<p>Record your observations</p> <p>(Question No. 28)</p> <p>When this code runs in the browser, it will display a list of numbers from 0 to 9, each on a new line, in the <p> element with the id "demo". This happens because the JavaScript do...while loop starts with i = 0 and appends the text "The number is 0", then increments i by 1. The loop keeps repeating until i reach 10. Unlike a while loop, the do...while loop guarantees that the code inside runs at least once before checking the condition.</p>

Reference:

- 1) Jason Beaird, The principles of Beautiful Web Design
- 2) Rian Ariona, Learn HTML and CSS (Fundamental Tutorial in Learning HTML and CSS)
- 3) Adi Hadisaputra, HTML and CSS Fundamentals from the Roots to the Leaves of John Duckett, HTML and CSS design and build websites

Github link : <https://github.com/Lovie-Tonimba/semester3-PemrogramanWeb.git>