WEB COMPLETE

HTML, CSS, JavaScript & PHP

#3
JavaScript

The programming language of HTML and the Web

JS Introduction

What is JavaScript?

- JavaScript is the programming language of HTML and the Web.
- JavaScript is easy to learn.

Why Study JavaScript?

JavaScript is one of the **3 languages** all web developers **must** learn:

1. **HTML** to define the content of web pages
2. **CSS** to specify the layout of web pages
3. **JavaScript** to program the behavior of web pages

What JavaScript can do

- JavaScript Can Change HTML Content
- JavaScript Can Change HTML Attributes
- JavaScript Can Change HTML Styles (CSS)
- JavaScript Can Hide HTML Elements

JS Example

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

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello
JavaScript!"'>Click Me!</button>

</body>
</html>
```

JS Example

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

<h2>What Can JavaScript Do?</h2>
<p>JavaScript can change HTML attributes.</p>
<p>In this case JavaScript changes the src (source) attribute of an image.</p>
<button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn on the
light</button>
<img id="myImage" src="pic_bulboff.gif" style="width:100px">
<button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn off
the light</button>

</body>
</html>
```

JS Example

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

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<button type="button"
onclick="document.getElementById('demo').style.fontSize='35px'">Click Me!</button>



</body>
</html>
```

JS Example

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

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can hide HTML elements.</p>

<button type="button"
onclick="document.getElementById('demo').style.display='none'">Click Me!</button>



</body>
</html>
```

JS Example

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

<h2>What Can JavaScript Do?</h2>

<p>JavaScript can show hidden HTML elements.</p>

<p id="demo" style="display:none">Hello JavaScript!</p>

<button type="button"
onclick="document.getElementById('demo').style.display='block'">Click Me!</button>

</body>
</html>
```

JS Where To

The `<script>` Tag

In HTML, JavaScript code must be inserted between `<script>` and `</script>` tags.

JS Example

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

<h2>JavaScript in Body</h2>

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

<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>

</body>
</html>
```

JavaScript in <head> or <body>

You can place any number of scripts in an HTML document.

Scripts can be placed in the **<body>**, or in the **<head>** section of an HTML page, or in both.

JavaScript in <head>

In this example, a JavaScript function is placed in the <head> section of an HTML page.

The function is invoked (called) when a button is clicked:

JS Example

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>
<h2>JavaScript in Head</h2>
<p id="demo">A Paragraph.</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
```

JavaScript in <body>

In this example, a JavaScript function is placed in the <body> section of an HTML page.

The function is invoked (called) when a button is clicked:

* Tips: Placing scripts at the bottom of the <body> element improves the display speed, because script compilation slows down the display.

JS Example

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript in Head</h2>
<p id="demo">A Paragraph.</p>
<button type="button" onclick="myFunction()">Try it</button>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</body>
</html>
```

External JavaScript

Scripts can also be placed in external files.

External scripts are practical when the same code is used in many different web pages.

JavaScript files have the file extension **.js**

To use an external script, put the name of the script file in the src (source) attribute of a `<script>` tag:

External JavaScript

You can place an external script reference in <head> or <body> as you like.

The script will behave as if it was located exactly where the <script> tag is located.

External scripts cannot contain <script> tags.

JS Example

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

<h2>External JavaScript</h2>
<p id="demo">A Paragraph.</p>
<button type="button" onclick="myFunction()">Try it</button>
<p>(myFunction is stored in an external file called "myScript.js")</p>
<script src="myScript.js"></script>

</body>
</html>
```

JS Example

```
function myFunction() {
   document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

External JavaScript Advantages

Placing scripts in external files has some advantages:

- It separates HTML and code
- It makes HTML and JavaScript easier to read and maintain
- Cached JavaScript files can speed up page loads

External References

External scripts can be referenced with a full URL or with a path relative to the current web page.

This example uses a full URL to link to a script:

JS Example

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

<h2>External JavaScript</h2>
<p id="demo">A Paragraph.</p>
<button type="button" onclick="myFunction()">Try it</button>
<p>(myFunction is stored in an external file called "myScript.js")</p>
<script src="https://www.w3schools.com/js/myScript.js"></script>

</body>
</html>
```

JS Output

JavaScript Display Possibilities

JavaScript can "display" data in different ways:

- Writing into an HTML element, using **innerHTML**.
- Writing into the HTML output using **document.write()**.
- Writing into an alert box, using **window.alert()**.
- Writing into the browser console, using **console.log()**.

Using innerHTML

To access an HTML element, JavaScript can use the **document.getElementById(id)** method.

The **id** attribute defines the HTML element. The **innerHTML** property defines the HTML content:

JS Example

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

<h2>My First Web Page</h2>
<p>My First Paragraph.</p>

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

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

</body>
</html>
```

Using document.write()

For testing purposes, it is convenient to use document.write():

The document.write() method should only be used for testing.

JS Example

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

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

Using window.alert()

You can use an alert box to display data:

JS Example

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

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

Using console.log()

For debugging purposes, you can use the console.log() method to display data.

JS Example

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

<h2>Activate debugging with F12</h2>

<p>Select "Console" in the debugger menu. Then click Run again.</p>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

JS Syntax

JS Syntax

JavaScript syntax is the set of rules, how JavaScript programs are constructed.

JavaScript Programs

A computer program is a list of "instructions" to be "executed" by the computer.

In a programming language, these program instructions are called statements.

JavaScript is a programming language.

JavaScript statements are separated by semicolons:

JS Example

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

<h2>JavaScript Statements</h2>
<p id="demo"></p>
<script>
var x, y, z;
x = 5;
y = 6;
z = x + y;
document.getElementById("demo").innerHTML = z;
</script>

</body>
</html>
```

JavaScript Statements

JavaScript statements are composed of:

Values, Operators, Expressions, Keywords, and Comments.

JavaScript Values

The JavaScript syntax defines two types of values: Fixed values and variable values.

Fixed values are called **literals**. Variable values are called **variables**.

JavaScript Literals

The most important rules for writing fixed values are:

Numbers are written with or without decimals:

```
10.50   1001
```

Strings are text, written within double or single quotes:

```
"John Doe"    'John Doe'
```

JavaScript Variables

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

JavaScript uses the **var** keyword 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:

```
var x;
x = 6;
```

JavaScript Operators

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

```
(5 + 6) * 10
```

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

```
var x, y;
x = 5;
y = 6;
```

JavaScript Comments

To make comment, use **//this is comment**

or **/* this is comment */**

JavaScript Variables

- How to create variable: just use **var | or use let (modern js)**
- Example: var variableName; let variableModern;
- **JavaScript Has Dynamic Types** which means the variable (data) types will change according to the variable value. The same variable can be used as different types
- Data type are: number, string, array, object, boolean, etc.

```
var length = 16;                                // Number
var lastName = "Johnson";                       // String
var cars = ["Saab", "Volvo", "BMW"];            // Array
var x = {firstName:"John", lastName:"Doe"};     // Object
```

JavaScript Array

- Array didefinisikan

```
var array-name = [item1, item2, ...];
```

```
var cars = ["Saab", "Volvo", "BMW"];
```

```
var points = [];
```

- Akses Data Array

```
var name = cars[0];
```

```
cars[0] = "Opel";
```

JavaScript Array

- You can have different objects in one array
- Array properties : length
- Array methods: sort()
- Adding array

```
var x = cars.length;
cars
var y = cars.sort();
```

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.push("Lemon");                 // adds a new element (Lemon) to fruits
```

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits[fruits.length] = "Lemon";      // adds a new element (Lemon) to fruits
```

JavaScript Object

How to create object

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

```
var person = {
    firstName:"John",
    lastName:"Doe",
    age:50,
    eyeColor:"blue"
};
```

```
var person = {
    firstName: "John",
    lastName : "Doe",
    id       : 5566,
    fullName : function() {
       return this.firstName + " " + this.lastName;
    }
};
```

JavaScript Object

How to access object properties

```
objectName.propertyName
```

```
objectName["propertyName"]
```

How to access object methods

```
objectName.methodName()
```

Operator Arimatika

Sama dengan Bahasa pemrograman lainnya

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
++	Increment
--	Decrement

Operator	Example
=	x = y
+=	x += y
-=	x -= y
*=	x *= y
/=	x /= y
%=	x %= y

Operator Logika

JavaScript Comparison and Logical Operators

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

Javascript Function

Function syntax

```
function name(parameter1, parameter2, parameter3) {
    code to be executed
}
```

```
1. ...
2. <script type="text/javascript">
3. function salam()
4. {
5.    alert("Assalamualaikum Saudaraku !");
6. }
7. function salamKenal(teman){
8.    alert("Apa kabar " + teman);
9. }
10. function jumlah(a,b){
11.    let c = a + b;
12.    return c;
13. }
14.</script>
...
```

Javascript Conditional & Loop

if else

```
if (condition1) {
    block of code to be executed if condition1 is true
} else if (condition2) {
    block of code to be executed if the condition1 is false and condition2 is true
} else {
    block of code to be executed if the condition1 is false and condition2 is false
}
```

switch case

```
switch(expression) {
    case n:
        code block
        break;
    case n:
        code block
        break;
    default:
        default code block
}
```

Javascript Conditional & Loop

for loop

```
for (statement 1; statement 2; statement 3) {
    code block to be executed
}
```

```
for (i = 0; i < 5; i++) {
    text += "The number is " + i + "<br>";
}
```

for in loop

```
var person = {fname:"John", lname:"Doe", age:25};

var text = "";
var x;
for (x in person) {
    text += person[x];
}
```

Javascript While Loop

while loop

```
while (condition) {
    code block to be executed
}
```

```
while (i < 10) {
    text += "The number is " + i;
    i++;
}
```

do while loop

```
do {
    code block to be executed
}
while (condition);
```

```
do {
    text += "The number is " + i;
    i++;
}
while (i < 10);
```

Javascript Event

Event Handlers	Deskripsi	Objek
onClick	Reaksi terhadap aksi meng-click-mouse pada suatu objek	checkboxes, links, radio buttons, reset buttons, dan submit buttons
onMouseOver	Reaksi jika suatu cursor atau mousepointer yang menunjuk ke suatu objek	links
onMouseOut	Reaksi jika suatu cursor atau mousepointer yang meninggalkan suatu objek	links
onLoad	Reaksi jika suatu objek selesai di-load.	images, windows
onUnload	Reaksi jika suatu dokumen ditutup/diakhiri.	windows
onAbort	Reaksi jika suatu objek diberhentikan dari proses loading	images
onChange	Reaksi jika suatu nilai(value) dari objek dimodifikasi	file uploads, select objects, text boxes, textarea
onSelect	Reaksi jika sutau teks dipilih dari objek text box atau textarea	text boxs, text areas
onError	Reaksi jika terjadi error JavaScript	images, windows
onReset	Reaksi jika suatu tombol reset dalam form ditekan	forms
onSubmit	Reaksi jika suatu tombol button dalam form ditekan	forms

```
<button onclick="displayDate()">The time is?</button>
```

HTML DOM (Document Object Model)

Halaman web HTML adalah sebuah object document yang mempresentasikan dokumen HTML yang di load pada saat itu.

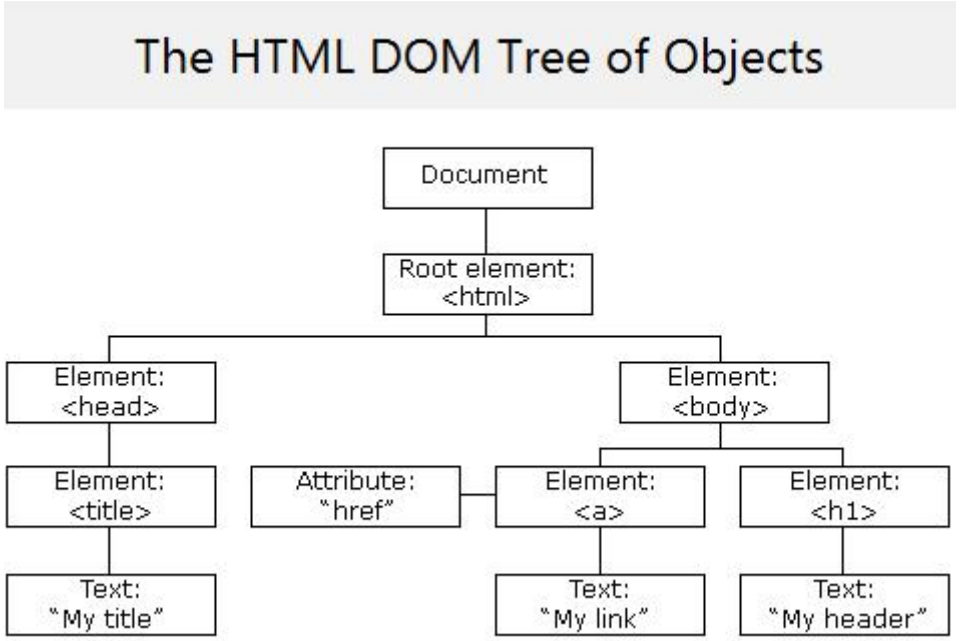
Didalam dokumen HTML terdapat object-object lainnya seperti :

- Images
- Forms : elemen array form
- Links
- Frame
- Table
- List
- Elemen-elemen HTML lainnya

HTML DOM

Dengan HTML DOM , JavaSCript dapat mengakses dan mengubah semua elemen dalam dokumen HTML

http://www.w3schools.com/jsref/default.asp



HTML DOM

Finding element

Method	Description
document.getElementById()	Find an element by element id
document.getElementsByTagName()	Find elements by tag name
document.getElementsByClassName()	Find elements by class name

Changing element

Method	Description
element.innerHTML=	Change the inner HTML of an element
element.attribute=	Change the attribute of an HTML element
element.setAttribute(*attribute,value*)	Change the attribute of an HTML element
element.style.*property*=	Change the style of an HTML element

HTML DOM

Create element

Method	Description
document.createElement()	Create an HTML element
document.removeChild()	Remove an HTML element
document.appendChild()	Add an HTML element
document.replaceChild()	Replace an HTML element
document.write(*text*)	Write into the HTML output stream

Changing style

```
document.getElementById(id).style.property=new style
```

Thanks!

Any questions?

You can find me at:
@edoriansyah
edo@nurulfikri.com

Credits

Special thanks to :

- Presentation template by SlidesCarnival
- w3schools.com