

JAVASCRIPT

A. Pengenalan

JavaScript adalah bahasa pemrograman paling populer di dunia.

JavaScript adalah bahasa pemrograman Web.

JavaScript mudah dipelajari.

Tutorial ini akan mengajarkan Anda JavaScript dari dasar.

Mengapa Belajar JavaScript?

JavaScript adalah salah satu dari 3 bahasa yang harus dipelajari semua pengembang web:

1. HTML untuk mendefinisikan konten halaman web
2. CSS untuk menentukan tata letak halaman web
3. JavaScript untuk memprogram perilaku halaman web

B. Penulisan Javascript

Tag `<script>`

Dalam HTML, kode JavaScript disisipkan di antara tag `<script>` dan `</script>`.

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

Contoh JavaScript lama mungkin menggunakan: `<script type="text/javascript">`.

Atribut type tidak diperlukan. JavaScript adalah bahasa skrip default dalam HTML.

Fungsi dan Event JavaScript

Fungsi JavaScript adalah blok kode JavaScript, yang dapat dijalankan saat "dipanggil".

Misalnya, suatu fungsi dapat dipanggil ketika suatu peristiwa terjadi, seperti ketika pengguna mengklik tombol.

Anda akan belajar lebih banyak tentang fungsi dan acara di bab-bab selanjutnya.

JavaScript di `<head>` atau `<body>`

Anda dapat menempatkan sejumlah skrip dalam dokumen HTML.

Skrip dapat ditempatkan di <body>, atau di bagian <head> pada halaman HTML, atau di keduanya.

JavaScript di <head>

Dalam contoh ini, fungsi JavaScript ditempatkan di bagian <head> halaman HTML.

Fungsi dipanggil (dipanggil) ketika tombol diklik:

```
<!DOCTYPE html>
<html>

<head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>

<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

</body>
</html>
```

JavaScript di <body>

Dalam contoh ini, fungsi JavaScript ditempatkan di bagian <body> halaman HTML.

Fungsi dipanggil (dipanggil) ketika tombol diklik:

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

<h1>A Web Page</h1>
<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>
```

Menempatkan skrip di bagian bawah elemen <body> meningkatkan kecepatan tampilan, karena interpretasi skrip memperlambat tampilan.

JavaScript eksternal

Skrip juga dapat ditempatkan di file eksternal:

External file: myScript.js

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

Skrip eksternal praktis ketika kode yang sama digunakan di banyak halaman web yang berbeda.

File JavaScript memiliki ekstensi file .js.

Untuk menggunakan skrip eksternal, letakkan nama file skrip di atribut src (source) dari tag <script>:

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

Anda dapat menempatkan referensi skrip eksternal di <head> atau <body> sesuka Anda.

Script akan berperilaku seolah-olah terletak persis di mana tag <script> berada.

Skrip eksternal tidak boleh berisi tag <script>.

Keuntungan JavaScript Eksternal

Menempatkan skrip di file eksternal memiliki beberapa keuntungan:

- Ini memisahkan HTML dan kode
- Itu membuat HTML dan JavaScript lebih mudah dibaca dan dipelihara
- File JavaScript yang di-cache dapat mempercepat pemuatan halaman

Untuk menambahkan beberapa file skrip ke satu halaman - gunakan beberapa tag skrip:

```
<script src="myScript1.js"></script>  
<script src="myScript2.js"></script>
```

C. JavaScript Output

Kemungkinan Tampilan JavaScript

JavaScript dapat "menampilkan" data dengan cara yang berbeda:

1. Menulis ke dalam elemen HTML, menggunakan innerHTML.
2. Menulis ke dalam output HTML menggunakan document.write().
3. Menulis ke dalam kotak peringatan, menggunakan window.alert().
4. Menulis ke konsol browser, menggunakan console.log().

Menggunakan innerHTML

Untuk mengakses elemen HTML, JavaScript dapat menggunakan metode document.getElementById(id).

Atribut id mendefinisikan elemen HTML. Properti innerHTML mendefinisikan konten HTML:

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

<h1>My First Web Page</h1>
<p>My First Paragraph</p>

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

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

</body>
</html>
```

Mengubah properti innerHTML dari elemen HTML adalah cara umum untuk menampilkan data dalam HTML.

Menggunakan document.write()

Untuk tujuan pengujian, akan lebih mudah menggunakan document.write():

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

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

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

```
</script>
```

```
</body>
```

```
</html>
```

Menggunakan document.write() setelah dokumen HTML dimuat, akan menghapus semua HTML yang ada:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My First Web Page</h1>
```

```
<p>My first paragraph.</p>
```

```
<button type="button" onclick="document.write(5 + 6)">Try it</button>
```

```
</body>
```

```
</html>
```

Metode document.write() hanya boleh digunakan untuk pengujian.

Menggunakan window.alert()

Anda dapat menggunakan kotak peringatan untuk menampilkan data:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My First Web Page</h1>
```

```
<p>My first paragraph.</p>
```

```
<script>
```

```
window.alert(5 + 6);
```

```
</script>
```

```
</body>
```

```
</html>
```

Anda dapat melewati kata kunci window.

Dalam JavaScript, objek window adalah objek lingkup global, yang berarti bahwa variabel, properti, dan metode secara default milik objek window. Ini juga berarti bahwa menentukan kata kunci window adalah opsional:

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

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

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

</body>
</html>
```

JavaScript Print

JavaScript tidak memiliki objek cetak atau metode cetak.

Anda tidak dapat mengakses perangkat keluaran dari JavaScript.

Satu-satunya pengecualian adalah Anda dapat memanggil metode window.print() di browser untuk mencetak konten dari jendela saat ini.

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

<button onclick="window.print()">Print this page</button>

</body>
</html>
```

D. JavaScript Statements

Contoh

```
var x, y, z;    // Statement 1
x = 5;          // Statement 2
y = 6;          // Statement 3
z = x + y;      // Statement 4
```

Program JavaScript

Program komputer adalah daftar "instruksi" yang harus "dieksekusi" oleh komputer.

Dalam bahasa pemrograman, instruksi pemrograman ini disebut statement / pernyataan.

Program JavaScript adalah daftar pernyataan pemrograman.

Dalam HTML, program JavaScript dijalankan oleh browser web.

JavaScript Statements

Pernyataan JavaScript terdiri dari:

Nilai, Operator, Ekspresi, Kata Kunci, dan Komentar.

Pernyataan ini memberitahu browser untuk menulis "Hello Dolly." di dalam elemen HTML dengan id="demo":

```
document.getElementById("demo").innerHTML = "Hello Dolly.";
```

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

<h2>JavaScript Statements</h2>

<p>In HTML, JavaScript statements are executed by the browser.</p>

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

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

</body>
</html>
```

Sebagian besar program JavaScript berisi banyak pernyataan JavaScript.

Pernyataan dieksekusi, satu per satu, dalam urutan yang sama seperti yang ditulis.

Program JavaScript (dan pernyataan JavaScript) sering disebut kode JavaScript.

Semicolons ;

Titik koma memisahkan pernyataan JavaScript.

Tambahkan titik koma di akhir setiap pernyataan yang dapat dieksekusi:

```
var a, b, c;      // Declare 3 variables
a = 5;           // Assign the value 5 to a
b = 6;           // Assign the value 6 to b
c = a + b;        // Assign the sum of a and b to c
```

Jika dipisahkan dengan titik koma, beberapa pernyataan pada satu baris diperbolehkan:

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

Di web, Anda mungkin melihat contoh tanpa titik koma. Mengakhiri pernyataan dengan titik koma tidak diperlukan, tetapi sangat disarankan.

JavaScript White Space

JavaScript mengabaikan banyak spasi. Anda dapat menambahkan spasi putih ke skrip Anda agar lebih mudah dibaca.

Baris berikut setara:

```
var person = "Hege";
var person="Hege";
```

Praktik yang baik adalah menempatkan spasi di sekitar operator (= + - * /):

```
var x = y + z;
```

JavaScript Line Length and Line Breaks

Untuk keterbacaan terbaik, pemrogram sering kali ingin menghindari baris kode yang lebih panjang dari 80 karakter.

Jika pernyataan JavaScript tidak muat pada satu baris, tempat terbaik untuk memecahnya adalah setelah operator:

```
document.getElementById("demo").innerHTML =
"Hello Dolly!";
```


JavaScript Code Blocks

Pernyataan JavaScript dapat dikelompokkan bersama dalam blok kode, di dalam tanda kurung kurawal {...}.

Tujuan dari blok kode adalah untuk mendefinisikan pernyataan yang akan dieksekusi bersama-sama.

Satu tempat Anda akan menemukan pernyataan yang dikelompokkan bersama dalam blok, adalah dalam fungsi JavaScript:

```
function myFunction() {  
    document.getElementById("demo1").innerHTML = "Hello Dolly!";  
    document.getElementById("demo2").innerHTML = "How are you?";  
}
```

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h2>JavaScript Statements</h2>  
  
<p>JavaScript code blocks are written between { and }</p>  
  
<button type="button" onclick="myFunction()">Click Me!</button>  
  
<p id="demo1"></p>  
<p id="demo2"></p>  
  
<script>  
function myFunction() {  
    document.getElementById("demo1").innerHTML = "Hello Dolly!";  
    document.getElementById("demo2").innerHTML = "How are you?";  
}  
</script>  
  
</body>  
</html>
```

JavaScript Keywords

Pernyataan JavaScript sering dimulai dengan kata kunci untuk mengidentifikasi tindakan JavaScript yang akan dilakukan.

Kunjungi referensi Kata Cadangan kami untuk melihat daftar lengkap kata kunci JavaScript.

Berikut adalah daftar beberapa kata kunci yang akan Anda pelajari dalam tutorial ini:

Deskripsi	Kata Kunci
break	Mengakhiri sakelar atau loop
continue	Melompat keluar dari lingkaran dan mulai dari atas
debugger	Menghentikan eksekusi JavaScript, dan memanggil (jika tersedia) fungsi debugging
do ... while	Mengeksekusi blok pernyataan, dan mengulangi blok, sementara kondisinya benar
for	Menandai blok pernyataan yang akan dieksekusi, selama kondisinya benar
function	Mendeklarasikan fungsi
if ... else	Menandai blok pernyataan yang akan dieksekusi, tergantung pada suatu kondisi

Deskripsi	Kata Kunci
return	Keluar dari fungsi
switch	Menandai blok pernyataan yang akan dieksekusi, tergantung pada kasus yang berbeda
try ... catch	Menerapkan penanganan kesalahan ke blok pernyataan
var	Mendeklarasikan variabel

Kata kunci JavaScript adalah kata-kata yang dicadangkan. Kata-kata yang dicadangkan tidak dapat digunakan sebagai nama untuk variabel.

E. JavaScript Syntax

Sintaks JavaScript adalah seperangkat aturan, bagaimana program JavaScript dibangun:

```
var x, y, z;           // Declare Variables
x = 5; y = 6;         // Assign Values
z = x + y;            // Compute Values
```

JavaScript Values

Sintaks JavaScript mendefinisikan dua jenis nilai:

1. Nilai tetap disebut Literal.
2. Nilai variabel

JavaScript Literal

Dua aturan sintaks yang paling penting untuk nilai tetap adalah:

1. Angka ditulis dengan atau tanpa desimal:

Contoh

10.50

1001

```
<!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>
```

2. String adalah teks, ditulis dalam tanda kutip ganda atau tunggal:

Contoh :

"John Doe"

'John Doe'

```

<!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>

```

JavaScript Variables

Dalam bahasa pemrograman, variabel digunakan untuk menyimpan nilai data.

JavaScript menggunakan kata kunci var untuk mendeklarasikan variabel.

Tanda sama dengan digunakan untuk memberikan nilai pada variabel.

Dalam contoh ini, x didefinisikan sebagai variabel. Kemudian, x diberikan (diberikan) nilai 6:

```

var x;
x = 6;

```

```

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

</body>
</html>

```

JavaScript Operators

JavaScript menggunakan operator aritmatika (+ - * /) untuk menghitung nilai:

(5 + 6) * 10

```

<!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 menggunakan operator penugasan (=) untuk menetapkan nilai ke variabel:

```

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

```

```

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

</body>
</html>

```

JavaScript Expressions

Ekspresi adalah kombinasi nilai, variabel, dan operator, yang menghitung ke suatu nilai.

Perhitungan itu disebut evaluasi.

Misalnya, 5 * 10 dievaluasi menjadi 50:

5 * 10

JavaScript Keywords

Kata kunci JavaScript digunakan untuk mengidentifikasi tindakan yang akan dilakukan.

Kata kunci var memberitahu browser untuk membuat variabel:

```

var x, y;
x = 5 + 6;
y = x * 10;

```

```

<!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>

```

Ekspresi juga dapat berisi nilai variabel:

x * 10

```

<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Expressions</h2>

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

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

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

</body>
</html>

```

Nilai dapat dari berbagai jenis, seperti angka dan string.

Misalnya, "John" + " " + "Doe", dievaluasi menjadi "John Doe":

"John" + " " + "Doe"

```

<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Expressions</h2>

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

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

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

</body>
</html>

```

JavaScript Comments

Tidak semua pernyataan JavaScript "dieksekusi".

Kode setelah garis miring ganda // atau antara /* dan */ diperlakukan sebagai komentar.

Komentar diabaikan, dan tidak akan dieksekusi:

```
var x = 5;    // I will be executed

// var x = 6;    I will NOT be executed
```

JavaScript Identifiers

Identifier adalah nama.

Dalam JavaScript, pengidentifikasi digunakan untuk memberi nama variabel (dan kata kunci, dan fungsi, dan label).

Aturan untuk nama resmi hampir sama di sebagian besar bahasa pemrograman.

Dalam JavaScript, karakter pertama harus berupa huruf, atau garis bawah (_), atau tanda dolar (\$).

Karakter berikutnya dapat berupa huruf, angka, garis bawah, atau tanda dolar.

Angka tidak diperbolehkan sebagai karakter pertama. Dengan cara ini JavaScript dapat dengan mudah membedakan pengidentifikasi dari angka.

JavaScript is Case Sensitive

Semua pengidentifikasi JavaScript peka terhadap huruf besar-kecil.

Variabel lastName dan lastname, adalah dua variabel yang berbeda:

```
var lastname, lastName;
lastName = "Doe";
lastname = "Peterson";
```

JavaScript tidak menafsirkan VAR atau Var sebagai kata kunci var.

JavaScript and Camel Case

Secara historis, programmer telah menggunakan berbagai cara untuk menggabungkan beberapa kata menjadi satu nama variabel:

Tanda hubung:

first-name, last-name, master-card, inter-city.

Tanda hubung tidak diperbolehkan dalam JavaScript. Mereka dicadangkan untuk pengurangan.

Underscore:

first_name, last_name, master_card, inter_city.

Upper Camel Case (Pascal Case):

FirstName, LastName, MasterCard, InterCity.

Lower Camel Case:

Pemrogram JavaScript cenderung menggunakan camel case yang dimulai dengan huruf kecil:

firstName, lastName, masterCard, interCity.

F. JavaScript Variables

Variabel JavaScript adalah wadah untuk menyimpan nilai data.

Dalam contoh ini, x, y, dan z, adalah variabel, dideklarasikan dengan kata kunci var:

```
var x = 5;  
var y = 6;  
var z = x + y;
```

Dari contoh di atas, Anda dapat mengharapkan:

x menyimpan nilai 5

y menyimpan nilai 6

z menyimpan nilai 11

Aljabar

Dalam contoh ini, price1, price2, dan total, adalah variabel:

```
var price1 = 5;  
var price2 = 6;  
var total = price1 + price2;
```

Dalam pemrograman, seperti dalam aljabar, kami menggunakan variabel (seperti harga1) untuk menyimpan nilai.

Dalam pemrograman, seperti dalam aljabar, kita menggunakan variabel dalam ekspresi (total = price1 + price2).

Dari contoh di atas, Anda dapat menghitung total menjadi 11.

Variabel JavaScript adalah wadah untuk menyimpan nilai data.

JavaScript Identifiers

Semua variabel JavaScript harus diidentifikasi dengan nama unik.

Nama-nama unik ini disebut pengidentifikasi.

Pengidentifikasi dapat berupa nama pendek (seperti x dan y) atau nama yang lebih deskriptif (usia, jumlah, volume total).

Aturan umum untuk membangun nama untuk variabel (pengidentifikasi unik) adalah:

- Nama dapat berisi huruf, angka, garis bawah, dan tanda dolar.
- Nama harus diawali dengan huruf
- Nama juga dapat dimulai dengan \$ dan _ (tetapi kami tidak akan menggunakannya dalam tutorial ini)

- Nama peka huruf besar/kecil (y dan Y adalah variabel yang berbeda)
- Kata-kata yang dicadangkan (seperti kata kunci JavaScript) tidak dapat digunakan sebagai nama

Pengidentifikasi JavaScript peka terhadap huruf besar-kecil.

Operator Penugasan

Dalam JavaScript, tanda sama dengan (=) adalah operator "tugas", bukan operator "sama dengan".

Ini berbeda dengan aljabar. Berikut ini contoh:

```
x = x + 5
```

Namun, dalam JavaScript, ia memberikan nilai $x + 5$ ke x .

(Ini menghitung nilai $x + 5$ dan memasukkan hasilnya ke dalam x . Nilai x bertambah 5.)

Operator "sama dengan"/ "equal to" ditulis seperti `==` dalam JavaScript.

JavaScript Data Types

Variabel JavaScript dapat menampung angka seperti 100 dan nilai teks seperti "John Doe".

Dalam pemrograman, nilai teks disebut string teks.

JavaScript dapat menangani banyak jenis data, tetapi untuk saat ini, pikirkan saja angka dan string.

String ditulis di dalam tanda kutip ganda atau tunggal. Angka ditulis tanpa tanda kutip.

Jika Anda memasukkan angka dalam tanda kutip, itu akan diperlakukan sebagai string teks.

```
var pi = 3.14;
var person = "John Doe";
var answer = 'Yes I am!';
```

Mendeklarasikan (Membuat) Variabel JavaScript

Membuat variabel dalam JavaScript disebut "mendeklarasikan" variabel.

Anda mendeklarasikan variabel JavaScript dengan kata kunci `var`:

```
var carName;
```

Setelah deklarasi, variabel tidak memiliki nilai (secara teknis memiliki nilai `undefined`).

Untuk menetapkan nilai ke variabel, gunakan tanda sama dengan:

```
carName = "Volvo";
```

Anda juga dapat menetapkan nilai ke variabel saat Anda mendeklarasikannya:

```
var carName = "Volvo";
```

Pada contoh di bawah, kami membuat variabel bernama carName dan menetapkan nilai "Volvo" padanya.

Kemudian kami "mengeluarkan" nilai di dalam paragraf HTML dengan id="demo":

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

<script>
var carName = "Volvo";
document.getElementById("demo").innerHTML = carName;
</script>
```

Ini adalah praktik pemrograman yang baik untuk mendeklarasikan semua variabel di awal skrip.

Satu Pernyataan Banyak Variabel

Anda dapat mendeklarasikan banyak variabel dalam satu pernyataan.

Mulai pernyataan dengan var dan pisahkan variabel dengan koma:

```
var person = "John Doe", carName = "Volvo", price = 200;
```

Deklarasi dapat menjangkau beberapa baris:

```
var person = "John Doe",
carName = "Volvo",
price = 200;
```

Value = undefined

Dalam program komputer, variabel sering dideklarasikan tanpa nilai. Nilai dapat berupa sesuatu yang harus dihitung, atau sesuatu yang akan diberikan kemudian, seperti input pengguna.

Variabel yang dideklarasikan tanpa nilai akan memiliki nilai yang tidak terdefinisi.

Variabel carName akan memiliki nilai yang tidak ditentukan setelah eksekusi pernyataan ini:

```
var carName;
```

Re-Declaring JavaScript Variables

Jika Anda mendeklarasikan ulang variabel JavaScript, itu tidak akan kehilangan nilainya.

Variabel carName akan tetap memiliki nilai "Volvo" setelah eksekusi pernyataan berikut:

```
var carName = "Volvo";  
var carName;
```

JavaScript Arithmetic

Seperti aljabar, Anda dapat melakukan aritmatika dengan variabel JavaScript, menggunakan operator seperti = dan +:

```
var x = 5 + 2 + 3;
```

Anda juga dapat menambahkan string, tetapi string akan digabungkan:

```
var x = "John" + " " + "Doe";
```

Coba juga ini:

```
var x = "5" + 2 + 3;
```

hasilnya : 523

Jika Anda memasukkan angka dalam tanda kutip, sisa angka akan diperlakukan sebagai string, dan digabungkan.

Sekarang coba ini:

```
var x = 2 + 3 + "5";
```

hasilnya : 55