



Assalamualaikum



JavaScript





JavaScript adalah bahasa pemrograman tingkat tinggi dan dinamis. JavaScript populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Google Chrome, Internet Explorer, Mozilla Firefox, Netscape dan Opera. Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag SCRIPT.





JavaScript dan Java adalah bahasa yang sama sekali berbeda, baik dalam konsep maupun desain.





JavaScript ditemukan oleh Brendan Eich pada tahun 1995, dan menjadi standar ECMA pada tahun 1997.



ECMAScript adalah nama resmi bahasa tersebut.



Mengapa Belajar JavaScript?

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

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

Halaman web bukan satu-satunya tempat di mana JavaScript digunakan. Banyak program desktop dan server menggunakan JavaScript. Node.js adalah yang paling dikenal. Beberapa database, seperti MongoDB dan CouchDB, juga menggunakan JavaScript sebagai bahasa pemrograman mereka.



Dimana menuliskan script javascript ?

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

Anda dapat menempatkan sejumlah skrip dalam dokumen HTML.

Script dapat ditempatkan di `<body>`, atau di `<head>` bagian halaman HTML, atau di keduanya.



Script di <head>

```
<head>
  <meta charset="UTF-8">
  <title>Pengenal JavaScript</title>
  <script>
    // Tulis script disini
    var nama = "Fikri";
  </script>
</head>
```

Script di <body>

```
<body>
  <h1>JavaScript</h1>
  <script>
    // Tulis script disini
    var nama = "Fikri";
  </script>
</body>
```

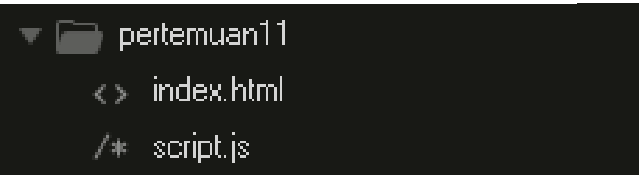


Skrip juga dapat ditempatkan di file eksternal.

File JavaScript memiliki ekstensi file **.js** .

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

Buat file JavaScript



Pada tag script tambahkan atribut src dengan nilai nama file javascript yang sudah dibuat.

```
<body>  
  <h1>JavaScript</h1>  
  <script src="script.js"></script>  
</body>
```




Output JavaScript

JavaScript dapat "menampilkan" data dengan berbagai cara:

- Menulis menjadi elemen HTML, menggunakan `innerHTML`.
- Menulis ke dalam output HTML menggunakan `document.write()`.
- Menulis ke dalam kotak peringatan, menggunakan `window.alert()`.
- Menulis ke konsol browser, menggunakan `console.log()`.



Output JavaScript

Meggunakan **innerHTML**.

```
<h1>Heading 1</h1>
<p>Paragraf satu</p>

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

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

Meggunakan **document.write**.

```
<h1>Heading 1</h1>
<p>Paragraf satu</p>

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

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

Meggunakan **console.log**.

```
<h1>Heading 1</h1>
<p>Paragraf satu</p>

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

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

Meggunakan **window.alert**.

```
<h1>Heading 1</h1>
<p>Paragraf satu</p>

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

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



Sintaks JavaScript

Sintaks JavaScript mendefinisikan dua jenis nilai:

Nilai tetap dan nilai variabel.

Nilai tetap disebut **literal**.

Nilai **variabel** disebut **variabel**.

Literal JavaScript

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

Variabel JavaScript

```
<script>  
    var x = 70;  
    document.getElementById("demo").innerHTML = x;  
</script>
```



Operator JavaScript

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

Contoh :

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

JavaScript menggunakan **operator penugasan** (**=**)
untuk **menetapkan** nilai ke variabel:

Contoh :

var x = 5;

var y = 6;



Variabel JavaScript

Semua **variabel** JavaScript harus **diidentifikasi** dengan **nama unik** .

Nama-nama unik ini disebut **pengidentifikasi** .

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

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

- Nama dapat berisi huruf, angka, garis bawah, dan tanda dolar.
- Nama harus dimulai dengan huruf.
- Nama juga dapat dimulai dengan \$ dan _
- Nama-nama adalah case-sensitive .
- Kata-kata yang dicadangkan (seperti kata kunci JavaScript) tidak dapat digunakan sebagai nama.



Tipe Data JavaScript

Variabel JavaScript dapat menampung banyak **tipe data** :
angka, string, objek, dan lainnya

JavaScript memiliki tipe dinamis. Ini berarti bahwa variabel yang sama dapat digunakan untuk menampung berbagai tipe data:

```
var x;  
x = 70;  
x = "Berubah";
```



String JavaScript

String (atau string teks) adalah serangkaian karakter seperti "Ini String".

String ditulis dengan tanda kutip.

Menggunakan tanda kutip tunggal atau ganda:

```
var mobil = "BMW";    // menggunakan kutip dua  
var motor = 'Scoopy'; // menggunakan kutip satu
```



Nomor JavaScript

JavaScript hanya memiliki satu jenis angka.

Angka dapat ditulis dengan, atau tanpa desimal:

```
var x1 = 34.00;    // desimal  
var x2 = 34;       // bilangan bulat
```




Boolean JavaScript

Boolean hanya dapat memiliki dua nilai: true atau false.

```
var boolean1 = true;  
var boolean2 = false;
```



Array JavaScript

Array JavaScript ditulis dengan tanda kurung siku.
Item array dipisahkan dengan koma.

Indeks array berbasis nol, yang berarti item pertama adalah [0], kedua adalah [1], dan seterusnya.

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



Objek JavaScript

Objek JavaScript ditulis dengan kurung kurawal {}.
Properti objek ditulis sebagai nama: pasangan nilai,
dipisahkan oleh koma.

Dalam contoh Objek (orang) memiliki 4 properti:
firstName, lastName, age, dan eyeColor.

```
var orang = { firstName: "Muhammad",  
              lastName: "Fikri",  
              age: 17,  
              eyeColor: "biru" };
```



Objek JavaScript

Objek juga dapat memiliki **metode** .

Metode adalah **tindakan** yang dapat dilakukan pada objek.

Metode disimpan di properti sebagai **definisi fungsi** .

Mengakses Metode Objek

objectName.methodName()

```
var orang = {  
  firstName:"Muhammad",  
  lastName:"Fikri",  
  age:17,  
  eyeColor:"biru" ,  
  fullName : function(){  
    return this.firstName + " " + this.lastName;  
  }  
};  
  
document.getElementById("demo").innerHTML = orang.fullName();
```



Jenis operator JavaScript

Anda dapat menggunakan operator `typeof` JavaScript untuk menemukan jenis variabel JavaScript.

Operator `typeof` mengembalikan jenis variabel atau ekspresi:

```
typeof mobil;  
typeof x1;  
typeof boolean1;  
typeof cars;  
typeof orang;
```



Fungsi JavaScript

Fungsi JavaScript adalah blok kode yang dirancang untuk melakukan tugas tertentu.

Fungsi JavaScript dijalankan ketika "sesuatu" memanggilnya (menyebutnya).

```
function myFunction(angka1, angka2) {  
    return angka1 * angka2;  
}  
  
document.getElementById("demo").innerHTML = myFunction(4, 3);
```



Sintaks Fungsi JavaScript

Fungsi JavaScript didefinisikan dengan kata kunci **function**, diikuti oleh **nama** , diikuti oleh tanda kurung **()**.

Nama fungsi dapat berisi **huruf**, **angka**, **garis bawah**, dan **tanda dolar** (aturan yang sama dengan variabel).

Tanda kurung dapat menyertakan nama parameter yang dipisahkan dengan koma:

(*parameter1, parameter2, ...*)

Kode yang akan dieksekusi, berdasarkan fungsinya, ditempatkan di dalam kurung keriting: **{ }**



Sintaks Fungsi JavaScript

```
function nama_fungsi(parameter1, parameter2, parameter3) {  
  // code perintah yang akan dijalankan  
}
```

Fungsi **parameter** tercantum di dalam tanda kurung () dalam definisi fungsi.

Argumen fungsi adalah **nilai yang** diterima oleh fungsi ketika dipanggil.

Di dalam fungsi, argumen (parameter) berperilaku sebagai variabel lokal.



Fungsi return JavaScript

Saat JavaScript mencapai pernyataan `return`, fungsi akan berhenti dijalankan.

Jika fungsi dipanggil dari pernyataan, JavaScript akan "kembali" untuk mengeksekusi kode setelah pernyataan pemanggilan.

Fungsi sering menghitung **nilai balik**. Nilai kembali "dikembalikan" kembali ke "pemanggilan"

```
function myFunction(angka1, angka2) {  
  return angka1 * angka2; // Function returns angka1 dan angka2  
}  
  
var x = myFunction(4, 3); // Fungsi dipanggil dan mengembalikan nilai
```



Event JavaScript

JavaScript memungkinkan Anda mengeksekusi kode ketika peristiwa terdeteksi.

HTML memungkinkan atribut event handler, **dengan kode JavaScript**, ditambahkan ke elemen HTML.

Dengan kutipan tunggal:

<element event='some JavaScript'>

Dengan kutipan ganda:

<element event="some JavaScript">



Event JavaScript

```
<button onclick="this.innerHTML = Date()">The time is?</button>
```

Pada contoh di atas, kode JavaScript mengubah konten elemen dengan id = "demo".

Dalam contoh berikut, kode mengubah konten elemennya sendiri (menggunakan **this.innerHTML**):



Event JavaScript

Batas	Cara Penulisan
Onchange	Elemen HTML telah diubah
OnClick	Pengguna mengklik elemen HTML
Onmouseover	Pengguna menggerakkan mouse di atas elemen HTML
Onmouseout	Pengguna memindahkan mouse menjauh dari elemen HTML
Onkeydown	Pengguna menekan tombol keyboard
Onload	Browser telah selesai memuat halaman



Studi kasus 1

Muhammad Fikri

- Nama Depan : Muhammad
- Nama Belakang : Fikri

Buatlah data diri anda dengan menggunakan javascript.

*hanya menggunakan tag HTML jangan menuliskan isinya :
contoh :

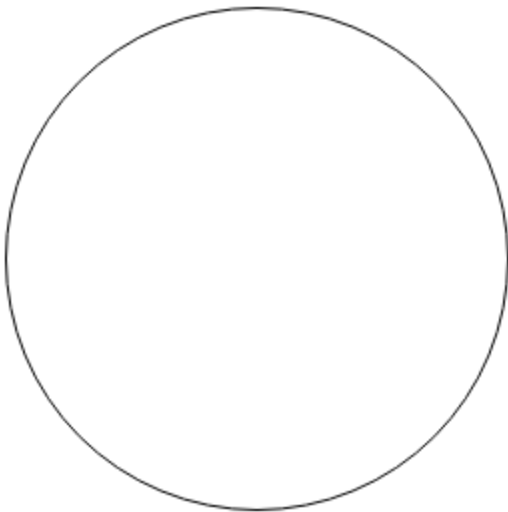
`<h1></h1>` (Benar)

`<h1>Muhammad Fikri</h1>` (Salah)

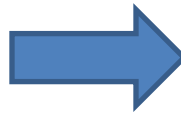
Studi kasus 2

```
<body>  
  <div id="demo"></div>  
  <a href="#" onclick="showImage();">Klik Me</a>  
</body>
```

Sebelum diklik



Klik Me



Setelah diklik



Klik Me

Referensi :

<https://id.wikipedia.org/wiki/JavaScript>

<https://www.w3schools.com/js/>