

Penjelasan Javascript

Javascript adalah bahasa pemrograman komputer yang memungkinkan user untuk berinteraksi dengan halaman web sehingga tampak lebih dinamis dan interaktif.

Sederhananya, javascript dapat digunakan untuk

- Mengubah isi konten HTML
- Mengubah atribut HTML
- Mengubah gaya / style HTML
- Memunculkan / Menyembunyikan elemen HTML
- dan masih banyak lagi

Technical Definition :

JavaScript adalah bahasa pemrograman single-threaded yang dapat kita gunakan untuk pengembangan sisi klien atau sisi server. JavaScript adalah bahasa pemrograman yang diketik secara dinamis, yang berarti kita tidak peduli dengan tipe data variabel saat menulis kode JavaScript. Selain itu, bahasa ini juga berisi pernyataan kontrol, operator, dan objek seperti Array, Matematika, Data, dll.

Fitur - Fitur Javascript

- JavaScript adalah bahasa pemrograman yang sangat populer dan banyak digunakan untuk pengembangan web. Bahasa ini memiliki berbagai fitur yang membuatnya kuat dan fleksibel. Beberapa fitur ini termasuk dinamis, ringan, dapat ditafsirkan, fungsional, dan berorientasi objek.
- Easy Setup : Kita tidak memerlukan text editor khusus untuk menulis kode javascript dan bisa langsung dijalankan di browser tanpa mengatur interpreter atau compiler.
- Browser Support : semua browser mendukung JavaScript, karena semua browser modern dilengkapi dengan *execution environment* JavaScript bawaan.
- DOM Manipulation : Javascript dapat digunakan untuk memanipulasi elemen pada halaman web.
- Event Handling : Javascript memungkinkan anda untuk menangani *events* yang digunakan untuk berinteraksi dengan halaman web.
- Dynamic Typing : Javascript menentukan tipe dari variabel pada runtime. Jadi, kita tidak perlu khawatir dengan tipe data variabel saat menulis kode, sehingga memberikan fleksibilitas yang lebih besar untuk menulis kode.
- Functional Programming : Javascript mendukung pemrograman fungsional. Di javascript dapat membuat *first-class function, closures, higher-order functions, arrow functions, function expressions, etc.*
- Cross Platform Support : Javascript dapat digunakan untuk pengembangan web, aplikasi mobile, game, aplikasi desktop, dsb.

- Object Oriented Programming : JavaScript juga mendukung konsep pewarisan, abstraksi, polimorfisme, enkapsulasi, dan lain-lain, konsep pemrograman berorientasi objek.
- Built-in Objects : JavaScript memiliki objek bawaan seperti Math dan Date. Kita dapat menggunakan objek Math untuk melakukan operasi matematika dan objek Date untuk memanipulasi tanggal dengan mudah
- Objects Prototypes : Dalam JavaScript, semuanya adalah objek. Sebagai contoh, array, function, number, string, boolean, set, map, dan lain-lain adalah objek. Setiap objek berisi properti prototipe, yang disembunyikan. Anda dapat menggunakan properti prototipe untuk mendapatkan pewarisan atau memperluas fungsionalitas kelas atau objek, dengan fungsionalitas objek lain.
- Global Object : JavaScript berisi objek global untuk mengakses variabel yang tersedia di mana saja.
- Built-in Methods : JavaScript juga berisi metode bawaan untuk setiap objek. Sebagai contoh, objek Array berisi metode filter() untuk memfilter elemen-elemen larik dan metode sort() untuk mengurutkan array. Objek String berisi metode replace() untuk mengganti teks dalam string, metode trim() untuk menghapus spasi dari string, dll.
- Modular Programming : JavaScript memungkinkan Anda untuk menulis kode dalam modul yang berbeda dan menghubungkannya dengan modul induk.
- JSON : JSON adalah singkatan dari Javascript object notation. Ini adalah format data yang banyak digunakan untuk bertukar data antara dua jaringan. Misalnya, server dan klien. JavaScript juga mendukung format JSON untuk menyimpan data.
- Asynchronous Programming : JavaScript adalah bahasa pemrograman single-threaded. Untuk mengeksekusi kode lebih cepat, Anda dapat menggunakan pemrograman asinkron. Anda dapat menggunakan promise dalam JavaScript untuk menulis kode asinkron, yang memungkinkan kita menangani beberapa tugas secara asinkron
- Event-driven Architecture : Arsitektur event-driven memungkinkan pengembang untuk membuat aplikasi web yang interaktif dan responsif dengan menangani basis pengguna yang besar secara bersamaan.
- Server Side Support : Runtime environment Node.js digunakan secara luas untuk membuat aplikasi backend, karena JavaScript juga dapat digunakan untuk membuat server. Hal ini memungkinkan Anda untuk membuat backend yang dapat diskalakan untuk aplikasi.

Menyisipkan Javascript

Terdapat fleksibilitas dalam penempatan kode javascript di dalam dokumen HTML.

- Script dalam elemen <head>
- Script dalam elemen <body>
- Script dalam elemen <head> dan <body>
- Script di file eksternal dan nantinya di masukkan ke dalam <head>

Internal

- Script dalam elemen <head>

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

    <script>
      // ... kode javascript
    </script>
  </head>
  <body>
    <p>Belajar Javascript</p>
  </body>
</html>
```

- Script dalam elemen <body>

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>
  </head>
  <body>
    <p>Belajar Javascript</p>

    <script>
      // ... kode javascript
    </script>
  </body>
</html>
```

Eksternal

- Membuat file terpisah untuk menulis kode javascript dan memanggilnya pada dokumen html.

```
JS script.js
1 // ... kode javascript
2
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

    <script src="script.js"></script>
  </head>

  <body>
    <p>Belajar Javascript</p>
  </body>
</html>
```

Tentang elemen <script>

- <script> memudahkan untuk mengintegrasikan kode javascript ke dokumen html
- ketika browser menemukan tag <script>, browser akan langsung mengeksekusi kode javascript di dalam tag tersebut.
- tag <script> memungkinkan untuk menggunakan library eksternal dan framework ke dokumen html. misalnya jQuery menggunakan CDN.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Belajar Javascript</title>

    <script src="https://code.jquery.com/jquery-3.7.1.js"></script>
  </head>

  <body></body>
</html>
```

Hello World Program

Berikut kode untuk menulis Hello World menggunakan Javascript di dalam dokumen HTML.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Belajar Javascript</title>

    <script>
      document.write('<p>Hello World!</p>');
    </script>
  </head>

  <body></body>
</html>
```

Hello World!

Console.log()

console.log() merupakan salah satu perintah penting di dalam javascript. Kode ini digunakan untuk mencetak pesan ke console web.

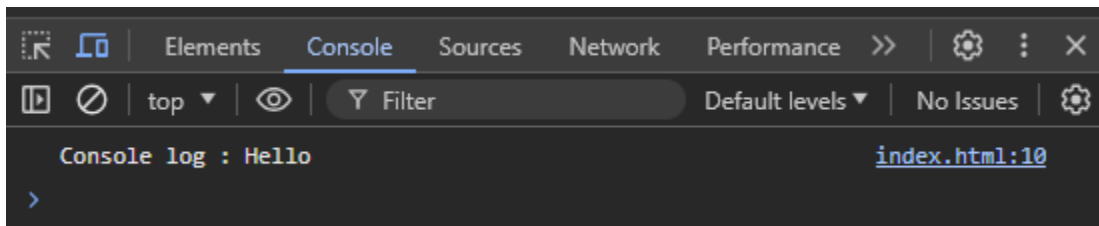
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Belajar Javascript</title>

    <script>
      console.log('Console log : Hello')
    </script>
  </head>

  <body></body>
</html>
```

- Untuk membuka jendela console di Google Chrome dapat menekan :
kombinasi tombol CTRL + SHIFT + J
membuka Devtools Chrome dengan F12 dan tekan panel Console.



Comments

Javascript comments digunakan untuk menulis pesan atau penjelasan dari kode sehingga memudahkan untuk dibaca.

komentar dapat digunakan untuk :

- Menjelaskan tujuan dari bagian kode tertentu.
- Menambahkan dokumentasi ke kode Anda untuk diri Anda sendiri dan orang lain yang mungkin membacanya.
- Menonaktifkan sementara atau “mengomentari” sebuah blok kode untuk pengujian atau debugging tanpa menghapusnya.

Variables

Variabel JavaScript digunakan untuk menyimpan data yang dapat diubah di nantinya. Variabel-variabel ini dapat dianggap sebagai wadah. Anda dapat menempatkan data ke dalam kontainer ini dan kemudian merujuk ke data tersebut hanya dengan memanggil namanya.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Belajar Javascript</title>

    <script>
      let x = 5;
      let y = 5;
      let z = x + y;

      console.log('result x + y = ' + z);
    </script>
  </head>

  <body></body>
</html>
```

Constant

Constant variabel yang nilainya tidak berubah selama eksekusi program. Anda dapat mendeklarasikan konstanta dengan menggunakan kata kunci const.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Belajar Javascript</title>

    <script>
      const x = 100;
      console.log('x = ' + x);
    </script>
  </head>

  <body></body>
</html>

```

Data Types

String

Tipe data string adalah kumpulan karakter. String dapat dideklarasikan dengan 3 cara, yaitu :

```

<script>
  let str1 = "Hello World!"; // Deklarasi string dengan double-quote
  let str2 = 'Hello World!'; // Deklarasi string dengan single-quote
  let str3 = `Hello World!`; // Deklarasi string dengan backtick
</script>

```

Numbers

Dalam javascript, tipe data numbers selalu disimpan dalam bentuk floating point / decimal.

```

<script>
  let num1 = 12; // Deklarasi Integer
  let num2 = 12.05; // Deklarasi Decimals
</script>

```

Boolean

Tipe data boolean adalah tipe data yang menyimpan nilai benar dan salah (true or false)

```

<script>
  let bool1 = true; // Boolean true
  let bool2 = false; // Boolean false
</script>

```

Null

Tipe data null digunakan apabila nilai suatu variabel tidak diketahui atau tidak ada.

```

<script>
  let nama = null;
</script>

```

Undefined

Tipe data undefined sama seperti tipe data null. Ketika variabel dideklarasikan tanpa nilai, nilai variabel tersebut adalah undefined.

```

<script>
  let kelas = undefined;
</script>

```

Symbol

Tipe data ini digunakan untuk membuat nilai primitif yang unik dan tidak dapat diubah. Simbol berguna terutama dalam konteks di mana Anda ingin memastikan bahwa tidak ada dua properti yang bentrok, bahkan jika mereka memiliki nama yang sama.

```

<script>
  let sym1 = Symbol();
  let sym2 = Symbol();

  console.log(sym1 === sym2); // false
</script>

```

BigInt

JavaScript hanya menyimpan angka floating point sepanjang 64-bit. Jika Anda ingin menyimpan angka yang sangat besar, Anda harus menggunakan BigInt. Anda dapat membuat BigInt dengan menambahkan n di akhir angka.

```
<script>
  let bigInt = 123456789012345678901234567890n;
</script>
```

Array

Dalam JavaScript, array adalah daftar elemen dengan tipe data yang berbeda. Anda dapat membuat larik menggunakan dua tanda kurung siku '[]' dan menyisipkan beberapa nilai yang dipisahkan koma di dalam array.

```
<script>
  let array = [10, 'abc', true];
</script>
```

Objects

Dalam JavaScript, tipe data objek memungkinkan kita untuk menyimpan koleksi data dalam format key-value.

```
<script>
  let car = {
    name: 'bmw',
    model: 'x5',
    color: 'blue',
  };
</script>
```

Arithmetic Operator

Operator aritmatika dalam JavaScript melakukan perhitungan matematis pada nilai numerik (operan). Sebagian besar operator aritmatika adalah operator biner karena melakukan perhitungan pada dua operan. Beberapa operator aritmatika adalah operator unary. Operator unary melakukan perhitungan pada satu operan.

Operator	Description	Example (x = 20, y = 10)
Addition (+)	Menambahkan 2 angka	$x + y = 30$
Subtraction (-)	Mengurangi satu angka dari angka lainnya	$x - y = 10$
Multiplication (*)	Mengalikan 2 angka	$x * y = 200$
Division (/)	Membagi 1 angka dengan angka lainnya	$x / y = 2$
Exponentiation (**)	Menghitung pangkat dari suatu angka	$x^{**}2 = 400$
Modulus (%)	Menghasilkan sisa dari pembagian 2 angka	$x \% y = 0$
Increment (++)	Menambah 1 pada nilai	$x++ = 21$
Decrement (--)	Mengurangi 1 pada nilai	$y-- = 9$

Comparison Operator

Operator perbandingan dalam JavaScript membandingkan dua variabel atau nilai dan mengembalikan nilai boolean, baik benar atau salah berdasarkan hasil perbandingan. Sebagai contoh, kita dapat menggunakan operator perbandingan untuk memeriksa apakah dua operan sama atau tidak.

Operator	Description	Example
==	Equal	$x == y$
!=	Not Equal	$x != y$
===	Strict equality (equal value and equal type)	$x === y$
!==	Strict inequality (not equal value or not equal type)	$x !== y$
>	Greater than	$x > y$
<	Less than	$x < y$
>=	Greater than or Equal to	$x >= y$
<=	Less than or Equal to	$x <= y$

Logical Operator

Operator logika dalam JavaScript umumnya digunakan dengan operan Boolean dan mengembalikan nilai boolean. Terdapat tiga jenis operator logika dalam JavaScript, yaitu && (AND), || (OR), dan ! (NOT). Operator-operator ini digunakan untuk mengontrol aliran program.

Operator	Description	Example (x = true, y = false)
&&	Logical AND	(x && y) is false.
	Logical OR	(x y) is true.
!	Logical NOT	!(x) is false.

Assignment Operator

Operator penugasan dalam JavaScript digunakan untuk menetapkan nilai ke variabel. Ini adalah operator biner. Operator penugasan mengambil dua operan, memberikan nilai ke operan kiri berdasarkan nilai operan kanan. Operan kiri selalu berupa variabel dan operan kanan dapat berupa literal, variabel, atau ekspresi.

Assignment Operator	Example	Equivalent To
= (Assignment)	a = b	a = b
+= (Addition Assignment)	a += b	a = a + b
-= (Subtraction Assignment)	a -= b	a = a - b
*= (Multiplication Assignment)	a *= b	a = a * b
/= (Division Assignment)	a /= b	a = a / b
%= (Remainder Assignment)	a %= b	a = a % b
**= (Exponentiation Assignment)	a **= b	a = a ** b

If...else

Pernyataan JavaScript if...else akan mengeksekusi sebuah blok kode ketika kondisi yang ditentukan adalah benar. Ketika kondisi tersebut salah, blok else akan dieksekusi. Pernyataan if-else dapat digunakan untuk mengontrol aliran eksekusi program berdasarkan kondisi yang berbeda.

```
<script>
  let age = 17;

  if(age > 18){
    console.log('Anda boleh mengemudikan motor.')
  } else {
    console.log('Anda tidak boleh mengemudikan motor.')
  }

  // Result : Anda tidak boleh mengemudikan motor.
</script>
```

Switch

JavaScript Switch adalah pernyataan bersyarat yang digunakan untuk mengeksekusi blok kode yang berbeda tergantung pada nilai ekspresi. Ekspresi dievaluasi, dan jika cocok dengan nilai salah satu label kasus, blok kode yang terkait dengan kasus tersebut akan dieksekusi. Jika tidak ada label kasus yang cocok dengan nilai ekspresi, blok kode yang terkait dengan label default akan dieksekusi.

```
<script>
  let day = 3;
  switch (day) {
    case 0:
      day = "Sunday";
      break;
    case 1:
      day = "Monday";
      break;
    case 2:
      day = "Tuesday";
      break;
    case 3:
      day = "Wednesday";
      break;
    case 4:
      day = "Thursday";
      break;
    case 5:
      day = "Friday";
      break;
    case 6:
      day = "Saturday";
      break;
  }

  console.log('Today is ' + day); // Output : Today is Wednesday
</script>
```

While

Tujuan perulangan while adalah untuk mengeksekusi pernyataan atau blok kode berulang kali selama ekspresi bernilai benar. Setelah ekspresi menjadi salah, perulangan akan berakhir.

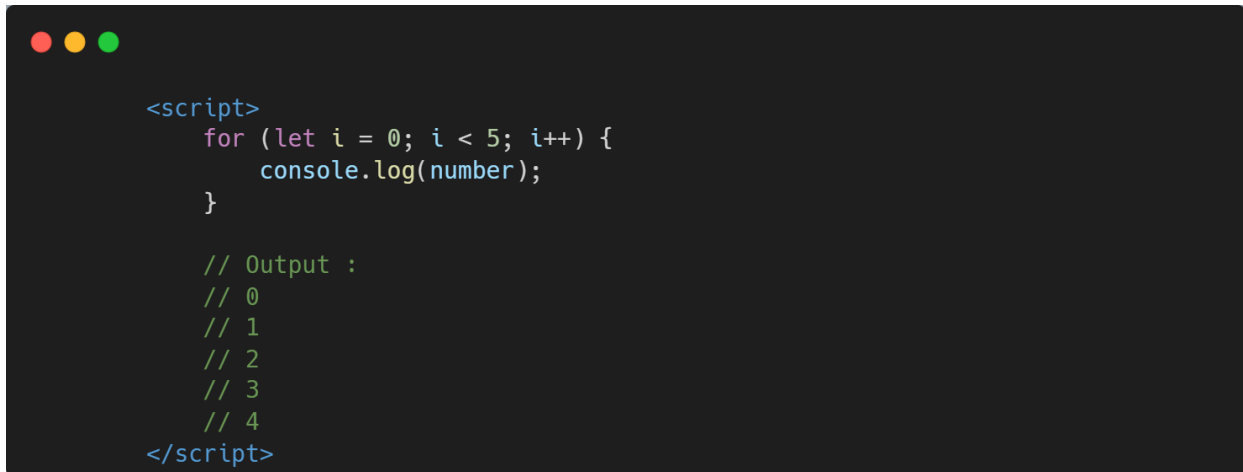
```
<script>
  let number = 0;

  while (number < 5) {
    console.log(number);
    number = number + 1;
  }

  // Result :
  // 0
  // 1
  // 2
  // 3
  // 4
</script>
```

For loop

JavaScript For loop digunakan untuk mengeksekusi blok kode secara berulang-ulang, hingga kondisi tertentu bernilai salah. Perulangan ini dapat digunakan untuk perulangan jika jumlah perulangannya tetap dan diketahui.



```
<script>
  for (let i = 0; i < 5; i++) {
    console.log(number);
  }

  // Output :
  // 0
  // 1
  // 2
  // 3
  // 4
</script>
```

Function

Fungsi dalam JavaScript adalah sekelompok kode yang dapat digunakan kembali yang dapat dipanggil di mana saja dalam program Anda. Hal ini menghilangkan kebutuhan untuk menulis kode yang sama berulang kali. Ini membantu programmer dalam menulis kode modular. Fungsi memungkinkan programmer untuk membagi program besar menjadi sejumlah fungsi kecil dan mudah dikelola.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Belajar Javascript</title>

    <script>
      function handleOnClick() {
        console.log('Button is clicked!');
      }
    </script>
  </head>

  <body>
    <button onclick="handleOnClick()">Click me!</button>
  </body>
</html>

```

Source

- https://www.w3schools.com/js/js_intro.asp
- https://www.tutorialspoint.com/javascript/javascript_overview.htm