

# JavaScript

JavaScript atau JS adalah bahasa pemrograman yang umum digunakan untuk mengembangkan aplikasi berbasis Web. JavaScript memungkinkan aplikasi Web lebih dinamis dan interaktif sehingga Web lebih dari sekedar tampilan statis untuk dibaca, tak heran jika JavaScript digunakan oleh lebih dari 95% Website di dunia.

Namun kini JavaScript sudah jauh berevolusi, tidak hanya disematkan di web yang eksekusi Browser, JavaScript kini populer juga sebagai teknologi Backend yang bisa digunakan untuk berkomunikasi dengan Server dan Database.

Kini bahkan kamu bisa secara khusus menekuni salah satunya yaitu focus ke Frontend atau Backend development dengan JavaScript. Untuk backend kamu bisa pelajari Node.JS atau Deno.

## Sejarah JavaScript

September 1995, seorang programmer berbakat bernama **Brendan Eich** mengembangkan bahasa scripting baru untuk digunakan di browser Netscape 2 hanya dalam 10 hari. Netscape 2 atau Netscape Navigator 2 adalah *browser* besutan perusahaan teknologi Netscape yang sangat populer dimasa itu.

Awalnya bahasa scripting baru ini dinamai Mocha lalu berubah nama menjadi LiveScript dan akhirnya menjadi JavaScript. **Brendan Eich** sendiri merupakan salah satu co-founder Mozilla Corporation dengan produk browser populer nya, Firefox.

Kini JavaScript menjadi *lingua franca* nya Web dan semua browser yang ada “wajib” men-support JavaScript.

JavaScript saat ini adalah salah satu bahasa pemrograman paling populer di dunia dengan *demand* yang tinggi baik dibidang Frontend ataupun Backend. Lowongan kerja untuk mereka para programmer JavaScript sudah tentu sangat banyak dan terus meningkat, inilah mengapa JavaScript adalah pilihan paling menjanjikan untuk kamu.

## Block scope

Sebuah variable dikatakan berada di *Block scope* jika dideklarasikan menggunakan `let` dan `const` dan berada di dalam block `{ }`. Jadi variable ini hanya bisa diakses di dalam block tersebut.

contoh:

```
let score = 70;

if (score >= 70) {
  let isPass = 'Yes';
  console.log(isPass);
}

// Output: Lulus
```

Jika kita coba akses `isPass` di luar kurawal `{ }` statement `if` nya, maka akan error. Karena variable `isPass` hanya bisa diakses di dalam block `{ }` ini.

```
let score = 70;

if (score >= 70) {
  let isPass = 'Yes';
}
console.log(isPass);

// ReferenceError: isPass is not defined
```

Lain hal jika kita menggunakan `var`, karena `var` selalu bersifat *Local scope* maka meski dideklarasikan di dalam block `{ }` baik itu dalam `if`, `for`, `while` dll (kecuali dalam `Function`), maka dia akan di *hoisting* alias diangkat ke atas oleh JavaScript.

contoh:

```
let score = 70;

if (score >= 70) {
  var isPass = 'Yes';
}
console.log(isPass);
```

```
setTimeout()  
setInterval()
```

Function-function di atas adalah bagian dari object global window dan tersedia secara global. Jadi bisa di akses di mana saja bahkan tanpa menuliskan object window nya khusus seperti berikut

```
window.alert('hello');  
window.console.log('hello');
```

cukup kita tulis seperti ini

```
alert('hello');  
console.log('hello');
```

## Local scope

Atau biasa disebut juga *Function scope*. Sebuah variable dikatakan berada di *Local scope* jika dideklarasi di dalam suatu function dan hanya bisa diakses di dalam function itu sendiri pula.

contoh:

```
function sayHello() {  
  let greeting = 'Hello world!';  
  console.log(greeting);  
}  
sayHello();
```

// Output: Hello world!

variable greeting di atas berada di *local scope* jadi hanya bisa di akses di dalam function sayHello() saja. Jika kita coba akses di luar function nya maka akan error.

```
function sayHello() {  
  let greeting = 'Hello world!';  
}  
console.log(greeting);  
sayHello();
```

// ReferenceError: greeting is not defined

- Local scope
- Block scope

## Global scope

Sebuah variable dikatakan berada di *Global scope* jika dideklarasi di luar function atau block tertentu. Global scope ini juga berarti variable yang ada di sana bisa diakses di mana pun di script js kita bahkan di halaman web kita.

Misal kita punya sebuah file `index.js` yang isinya seperti berikut:

```
let greeting = 'Hello world!';

// code yang lain
// code yang lain

console.log(greeting);

// Output: Hello world!

function sayHello() {
  console.log(greeting);
}

sayHello();

// Output: Hello world!
```

variable `greeting` di atas berada di global scope karenanya ia bersifat global, jadi bisa di akses di baris manapun di file `index.js` tersebut.

Para pembuat browser sudah menyematkan global variable / object bernama `window` yang sering kita gunakan dan bisa di akses dimanapun dan di dalam file script js manapun.

Function-function seperti berikut ini sering kali digunakan dalam keseharian coding JavaScript nantinya.

```
alert()
console.log()
```

Sebagai contoh:

```
window.setTimeout(() => {
    console.log('Timer finished')
}, 1000 );

window.alert('Buku Saku JavaScript');
```

Karena **window** object yang bersifat **Global**, maka kita tidak perlu coding seperti di atas. Cukup kita tulis seperti berikut

```
setTimeout(() => {
    console.log('Timer finished')
}, 1000 );

alert('Buku Saku JavaScript');
```

Nah sekarang kita jadi tahu bahwa **tidak semua** codingan JS yang kita tulis nantinya adalah asli bawaan JS Engine. Ada banyak sekali fungsi/fitur yang ternyata disediakan oleh Browser.

List Web API selengkapnya yang bisa kamu pelajari lebih lanjut di <https://developer.mozilla.org/en-US/docs/Web/API>

## Mengenal Scope di JavaScript

Scope adalah salah satu konsep fundamental yang **wajib** diketahui dan dipahami sebelum lebih jauh belajar JavaScript. Scope di JavaScript berbicara tentang bagaimana dan dimana variable-variable dapat diakses atau tidak dapat diakses.

Memahami scope di JavaScript begitu penting karena dapat membantu kita menulis code yang lebih efisien, meminimalisir bugs dan bisa juga membantu kita membuat pure function, yaitu function murni yang tidak bergantung kepada variable atau object lain yang ada di luar function tersebut. Kita akan lebih dalam mengenal pure function di pembahasan berikutnya mengenai Function.

Ada tiga scope utama di JavaScript yang wajib diketahui yaitu:

- Global scope

# Mengenal Web API

Suatu saat kita akan sampai ke tahap web development yang lebih dalam dan menemukan kode-kode seperti:

```
alert()  
  
console.log()  
  
setTimeout()  
  
setInterval()  
  
localStorage  
  
sessionStorage  
  
// event listener  
const btn = document.getElementById('btn');  
  
btn.addEventListener('click', function() {  
  alert('Buku Saku JavaScript');  
});
```

Function atau fitur di atas bukanlah bagian dari spesifikasi resmi JavaScript alias tidak ada implementasinya di JavaScript Engine. Fitur-fitur di atas adalah Web API yakni **fitur yang disediakan oleh browser** yang dapat dipanggil oleh JavaScript.

Jadi para pembuat browser seperti Google (Chrome), Mozilla (Firefox), Apple (Safari), Microsoft (Edge), dll semacam memiliki konsensus bersama untuk mengimplementasikan Web API di atas di browser masing-masing sehingga lebih universal dan lebih mudah bagi para developer. Bayangkan misal fitur `console.log()` hanya tersedia di Chrome saja tapi di Firefox tidak ada...misal saja di Firefox maunya `console.print()`, Edge `console.display()` maka ini akan membuat repot para developer/programmer.

Web API ada di dalam **window** object yang tersedia secara global (*Global scope*), bisa dipakai di mana pun di halaman suatu aplikasi web.