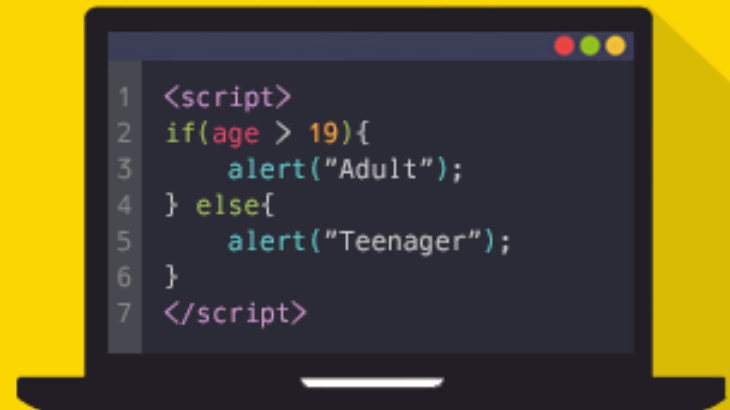




Assalamualaikum



JavaScript





Array JavaScript

Array adalah variabel khusus, yang dapat menampung lebih dari satu nilai pada suatu waktu.

```
var car1 = "Saab";  
var car2 = "Volvo";  
var car3 = "BMW";
```

Array dapat menyimpan banyak nilai dengan satu nama, dan Anda dapat mengakses nilai dengan merujuk ke nomor indeks.



Membuat Array

Menggunakan array literal adalah cara termudah untuk membuat JavaScript Array.

Sintaksis:

```
var nama_array = [item1, item2, ...];
```

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

atau

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



Akses Elemen dari Array

Anda mengakses elemen array dengan merujuk ke **nomor indeks** .

Nomor indeks dimulai dari angka 0.

Mehtod `length()` untuk menghitung jumlah item array.

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



Konversi Array ke String

Metode JavaScript `toString()` mengubah array menjadi string dengan nilai array (dipisahkan koma).

Metode `join()` juga bergabung semua elemen array menjadi string.

Itu berperilaku seperti `toString()`, tetapi selain itu Anda dapat menentukan pemisah:

```
cars.toString();  
cars.join(" * ");
```



Menghapus & Menambahkan Element

Saat Anda bekerja dengan array, mudah untuk menghapus elemen dan menambahkan elemen baru.

Metode **pop()** menghilangkan elemen terakhir dari array.

Metode **push()** menambahkan elemen baru ke array (di akhir).

```
cars.pop();
```

```
cars.push("BMW");
```



Pergeseran Elemen

Menggeser sama dengan popping, bekerja pada elemen pertama, bukan yang terakhir.

Metode **shift()** menghilangkan elemen pertama array.

Metode **unshift()** menambahkan elemen baru ke array (di awal).

```
cars.shift();
```

```
cars.unshift("Saab");
```



Mengubah Elemen

Elemen array diakses menggunakan **nomor indeks** mereka :

Syntaks dibawah akan mangubah nilai pada array dengan index tertentu :

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




Menyambungkan Array

Metode splice() dapat digunakan untuk menambahkan item baru ke array.

```
var buah = ["Banana", "Orange", "Apple", "Mango"];  
buah.splice(2, 0, "Lemon", "Kiwi");  
document.write(buah);
```



Array Penggabungan

Metode `concat()` menciptakan array baru dengan menggabungkan (concatenating) array yang ada:

Metode `concat()` tidak mengubah array yang ada. Itu selalu mengembalikan array baru.

```
var gabung = cars.concat(buah);  
document.write("<br>" + gabung);
```



Mengiris Array

Metode slice() irisan keluar sepotong array ke array baru.

Metode slice() menciptakan array baru. Itu tidak menghapus elemen apa pun dari array sumber.

```
document.write("<br><br>" + buah);

var ambil2 = buah.slice(2);
document.write("<br>" + ambil2);

var ambil = buah.slice(1, 3);
document.write("<br>" + ambil);

var ambil3 = buah.slice();
document.write("<br>" + ambil3);
```



Menyortir Array

Metode **sort()** mengurutkan element array sesuai dengan abjad, macam array abjad:

```
buah.sort()
```

Array Asli :

Banana,Orange,Lemon,Kiwi,Apple,Mango

Setelah sortir :

Apple,Banana,Kiwi,Lemon,Mango,Orange



Membalikkan Array

Metode **reverse()** membalikkan elemen dalam array.

```
var buah = ["Banana", "Orange", "Apple", "Mango"];
```

```
Buah.reverse();
```

Output : Mango,Apple,Orange,Banana



Iterasi Array

Metode iterasi array beroperasi pada setiap item array.

- `forEach()`
- `filter()`
- `reduce()`



Array.forEach ()

Metode forEach() panggilan fungsi (fungsi callback) sekali untuk setiap elemen array.

Fungsi tersebut membutuhkan 3 argumen:

- Nilai (Value)
- No indeks (Index)
- Array itu sendiri (Array)

```
var txt = "";  
var numbers = [45, 4, 9, 16, 25];  
numbers.forEach(myFunction);  
  
function myFunction(value, index, array) {  
    txt = txt + value + "<br>";  
}
```



Array.filter ()

Metode filter() menciptakan sebuah array baru dengan elemen array yang melewati tes.

Fungsi tersebut membutuhkan 3 argumen:

- Nilai (Value)
- No indeks (Index)
- Array itu sendiri (Array)

```
var numbers = [45, 4, 9, 16, 25];  
var over18 = numbers.filter(myFunction);  
document.write(over18);  
  
function myFunction(value, index, array) {  
    return value > 18;  
}
```




Array.reduce ()

Metode reduce() menjalankan fungsi pada setiap elemen array untuk menghasilkan (mengurangi ke) nilai tunggal.

Metode reduce() bekerja dari kiri ke kanan dalam array.

Fungsi tersebut membutuhkan 4 argumen:

- Total (nilai awal / nilai yang dikembalikan sebelumnya)
- Nilai (Value)
- No indeks (Index)
- Array itu sendiri (Array)

```
var numbers1 = [45, 4, 9, 16, 25];  
var sum = numbers1.reduce(myFunction);  
  
function myFunction(total, value, index, array) {  
    return total + value;  
}
```



Objek Tanggal JavaScript

Secara default, JavaScript akan menggunakan zona waktu browser dan menampilkan tanggal sebagai string teks lengkap:
Rab 18 Des 2019 19:44:46 GMT + 0700 (Waktu Indonesia Barat)

Objek tanggal dibuat dengan `new Date()` konstruktor.

Ada **4 cara** untuk membuat objek tanggal baru:

- `new Date()`
- `new Date(year, month, day, hours, minutes, seconds, milliseconds)`
- `new Date(milliseconds)`
- `new Date(date string)`



Objek Tanggal

Meggunakan `new Date()`

```
var d = new Date();  
document.write(d);
```

Meggunakan `new Date(dateString)`.

```
var d = new Date("October 13, 2014 11:13:00");  
document.write(d);
```

Meggunakan `new Date(milliseconds)`.

```
var d = new Date(0);  
document.write(d);
```



Objek Tanggal

Meggunakan `new Date(year, month, ...)`.

- 7 angka menentukan tahun, bulan, hari, jam, menit, detik, dan milidetik (dalam urutan itu):

```
var d = new Date(2018, 11, 24, 10, 33, 30, 0);
```

- 6 angka menentukan tahun, bulan, hari, jam, menit, detik:

```
var d = new Date(2018, 11, 24, 10, 33, 30, );
```

- 5 angka menentukan tahun, bulan, hari, jam, dan menit:

```
var d = new Date(2018, 11, 24, 10, 33);
```

- 4 angka menentukan tahun, bulan, hari, dan jam:

```
var d = new Date(2018, 11, 24, 10);
```



Objek Tanggal

Meggunakan `new Date(year, month, ...)`.

- 3 angka menentukan tahun, bulan, dan hari:

```
var d = new Date(2018, 11, 24);
```

- 2 angka menentukan tahun dan bulan:

```
var d = new Date(2018, 11 );
```

- Jika anda menggunakan 1 angka, maka angka tersebut akan menjadi fungsi milisecond .

```
var d = new Date(2018);
```

- Satu dan dua digit tahun akan ditafsirkan sebagai 19xx:

```
var d = new Date(99, 11, 24);
```



Metode Tanggal

Metode ini dapat digunakan untuk mendapatkan informasi dari objek tanggal:

Untuk menggunakan method tanggal kita harus memanggil objek tanggal itu sendiri.

```
var d = new Date();  
document.write(d.getTime());
```



Metode Tanggal

<code>getFullYear()</code>	Mendapatkan tahun sebagai angka, empat digit (yyyy)
<code>getMonth()</code>	Mendapatkan bulan sebagai angka (0-11)
<code>getDate()</code>	Mendapatkan hari sebagai angka (1-31)
<code>getHours()</code>	Mendapatkan angka jam (0-23)
<code>getMinutes()</code>	Mendapatkan angka menit (0-59)
<code>getSeconds()</code>	Mendapatkan angka detik (0-59)
<code>getMilliseconds()</code>	Mendapatkan angka detik (0-999)
<code>getTime()</code>	Mendapatkan angka (milliseconds terhitung dari January 1, 1970)
<code>getDay()</code>	Mendapatkan angka urutan hari (0-6)



Metode Matematika

Objek JavaScript Matematika memungkinkan Anda untuk melakukan tugas matematika pada angka.

Objem matematika menggunakan kata **Math** untuk menggunakan methodnya.



Metode Tanggal

Math.PI;	Mendapatkan nilai phi 3.141592653589793
Math.round (x)	Membulatkan nilai x ke bilangan terdekat.
Math.ceil(x)	Membulatkan nilai x ke atas.
Math.floor(x)	Membulatkan nilai x ke bawah.
Math.pow(x, y)	Mengembalikan nilai x kuadrat y.
Math.sqrt(x)	Mengembalikan akar kuadrat x
Math.random()	Mengembalikan nilai acak dari 0 sampai 1
Math.abs(x)	Mengembalikan nilai absolut (positif) dari x
Math.min(30, 20, -8, -200)	Menemukan nilai terendah
Math.max(30, 20, -8, -200)	Menemukan nilai tertinggi



Metode prompt ()

Metode prompt () menampilkan kotak dialog yang meminta pengunjung untuk input.

Metode prompt () mengembalikan nilai input jika pengguna mengklik "OK". Jika pengguna mengklik "batal" metode mengembalikan nol.



Sintaksis prompt()

`prompt(text, defaultText)`

<i>text</i>	Teks yang akan ditampilkan oleh prompt. Ini wajib ditulis
<i>defaultText</i>	Optional. Default text yang akan ditulis.

```
var nama = prompt("Masukan nama : ");
```



Studi kasus 1

Buatlah sebuah fungsi untuk menampilkan angka random dari 1 – 7 dan tampilkan dengan popUp.



Studi kasus 2

Membuat kubus dengan nilai sisi dari input user

This page says

Masukan nilai sisi :

OK

Cancel



$$L = 10000 \text{ m}^2$$



Studi kasus 3

Tampilkan data yang terdapat dalam array seperti berikut :

```
<script>
  var buah = ["Mangga", "Apel", "Jeruk", "Durian", "Leci"];
```

Sebelum diklik

Tampilkan daftar buah

Tampilkan

Setelah diklik

Daftar Buah

- Apel
- Durian
- Jeruk
- Leci
- Mangga

Balik Awal

Referensi :

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