

Join GitHub today

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

Sign up

Dismiss

Branch: master

phase-0-activities / modules / js-array.md

Find file

Copy path

adhywiranata

Update js-array.md

6394216 on Sep 13, 2017

2 contributors

# JavaScript Data Types - Array

## Objectives

Tipe data ataupun struktur data adalah konsep penting dalam mengkategorisasikan beberapa hal/jenis berbeda pada unit, pengukuran, juga memungkinkan kita untuk beroperasi pada variabel.

## Learnings

- [Array](#)
- [Array Built-in functions](#)
  - [push\(\)](#)
  - [pop\(\)](#)
  - [unshift\(\)](#)
  - [shift\(\)](#)
  - [sort\(\)](#)
  - [slice\(\)](#)
  - [splice\(\)](#)
  - [split\(\)](#)
- [Multidimensional Array](#)

## Array

Array adalah kumpulan dari berbagai data. Kita bisa tulis dengan kurung kotak (square brackets) dan butir-butirnya dipisah dengan koma. Indeks array berbasis-nol yang berarti urutan awalnya merupakan [0] , keduanya [1] , dan seterusnya. Kita juga bisa memasukkan (insert), memperbarui/mengubah (update/change), atau bahkan meniadakan (undefine) nilai yang ada di dalam array. Spesifik di JS, kita bisa gunakan beberapa tipe data berbeda di dalam suatu array. Bahkan memasukkan array ke dalam array!

```
> var animals = ["Lion", "Tiger", "Puma"]
> animals
> animals[0]
> animals[3] = "Jaguar"
> animals[0] = "Leopard"
> animals[1] = undefined
> animals
> animals[5]
> animals[0] = 100
> animals[1] = true
> animals
> animals[3] = ["Zero", 1, "Two"]
> animals[3][3] = "Three"
```

Array seperti tumpukan data. Bayangkan sebuah botol atau toples yang berisi beberapa lapis biskuit di dalamnya. Kita bisa mendorong (push) seperti memasukkan yang baru, mengeluarkan (pop) yang paling terakhir atau teratas, atau bahkan mengambil beberapa lapis data.

```
> animals.push("Dragon")
> animals.pop()
> animals.slice(0, 2)
```

## Array Built-in functions

- push: menambah 1 nilai ke array ke index paling belakang
- pop: menghapus 1 nilai dari array index paling belakang
- unshift: menambah 1 nilai ke array index paling depan (index 0)
- shift: menghapus 1 nilai dari array index paling depan (index 0)
- join: menggabungkan seluruh element array menjadi sebuah string dan mengambil parameter sebagai simbol penyambung antar elemen
- sort: mengurutkan elemen di dalam array sesuai alphabet
- slice: mengambil beberapa lapis data
- splice: mengubah nilai array dengan menghapus dan/atau menambah nilai baru ke array
- split: memecah string dan mengembalikan array sesuai dengan separator / pemisah yang didefinisikan

## Penggunaan Built-in Function Array secara singkat

```
var animalsArray = ["lion", "horse"];
animalsArray.push("duck"); // menambahkan 1 nilai ke array bagian paling belakang
console.log(animalsArray); // ["lion", "horse", "duck"]
animalsArray.pop(); // menghapus 1 nilai array terbelakang
console.log(animalsArray); // ["lion", "horse"]
animalsArray.unshift("dog");
console.log(animalsArray); // ["dog", "lion", "horse"]
animalsArray.shift();
console.log(animalsArray); // ["lion", "horse"]
animalsArray.sort();
console.log(animalsArray); // ["horse", "lion"]
console.log(animalsArray.join(" and ")); // "horse and lion"
```

Beberapa fungsi dari array ada yang mengubah nilai dari array itu sendiri (sort, splice, push, pop, shift, unshift) atau mengembalikan array / nilai baru (). Jika kamu bertanya kenapa berbeda, akan dijelaskan di Week 5 tentang prinsip mutability dan immutability.

## push()

Push adalah fungsi array yang akan menambahkan nilai di belakang nilai terakhir di array. Layaknya sebuah antrian, push adalah orang yang mengantri selanjutnya, akan berada di posisi paling belakang. Fungsi push akan menerima satu parameter, berupa nilai yang akan ditambahkan.

```
var arr = [0, 1, 2, 3];
arr.push(4);
console.log(arr); // 0, 1, 2, 3, 4
```

## pop()

Pop adalah fungsi array yang akan menghapus nilai paling belakang / terakhir dari array. Bayangkan dalam sebuah antrian, orang yang mengantri di paling belakang tidak jadi mengantri dan pulang. Fungsi pop tidak menerima parameter apapun.

```
var arr = [0, 1, 2, 3];
arr.pop();
console.log(arr); // 0, 1, 2
```

## unshift()

Unshift adalah fungsi array yang akan menambah nilai ke depan array (ke index 0), dan menggeser seluruh isi array kebelakang. Fungsi unshift menerima satu parameter, berupa nilai yang akan ditambahkan ke dalam array.

```
var arr = [0, 1, 2];
arr.unshift(3);
console.log(arr); // 3, 0, 1, 2
```

## shift()

Shift adalah fungsi array yang akan menghapus nilai paling depan dari array. Fungsi shift tidak menerima parameter apapun.

```
var arr = [3, 0, 1, 2];
arr.shift();
console.log(arr); // 0, 1, 2
```

## sort()

Sort adalah fungsi array yang akan mengurutkan nilai dari array. Perlu diperhatikan, sort mengurutkan otomatis secara ascending dan diurutkan berdasarkan unicode dari karakter. Kamu akan menemukan masalah ini saat mengurutkan angka.

```
var arr = [3, 5, 7, 1, 2];
arr.sort();
console.log(arr); // 1, 2, 3, 5, 7

var arrChar = ['Tono', 'Budi', 'Charlie', 'Ahmad'];
arrChar.sort();
console.log(arrChar); // 'Ahmad', 'Budi', 'Charlie', 'Tono'
```

Contoh masalah sort() pada angka:

```
var arr = [1, 2, 15];
arr.sort();
console.log(arr); // 1, 15, 2
```

Hasil dari sort di atas tidak sesuai dengan ekspektasi kita! Seharusnya 1, 2, 15 namun yang terjadi adalah 1, 15, 2. Hal ini karena JavaScript melakukan sort secara unicode atau sebagai karakter, dan karena 15 diawali dengan karakter '1', maka akan dianggap lebih awal daripada karakter '2'. Untuk menyelesaikan masalah ini, kamu perlu menambahkan satu parameter berupa fungsi pembanding. Mungkin kamu akan sedikit bingung dengan code dibawah ini, namun intinya adalah kita membuat satu fungsi yang menilai apakah nilai pertama lebih kecil dari nilai kedua.

```
var arr = [1, 2, 15];
arr.sort(function(value1, value2) { return value1 > value2 });
console.log(arr); // 1, 2, 15
```

dengan fungsi di dalam ini, kita juga bisa membuat fungsi sort yang descending:

```
var arr = [1, 2, 15];
arr.sort(function(value1, value2) { return value1 < value2 });
console.log(arr); // 15, 2, 1
```

## slice()

Slice adalah fungsi array yang akan mengambil irisan atau porsi dari sebuah array. Fungsi slice menerima satu atau dua parameter. Parameter pertama adalah index irisan diambil, dan parameter kedua adalah index irisan terakhir diambil. Jika parameter kedua tidak didefinisikan, irisan akan diambil hingga akhir dari array.

```
var arr = [0, 1, 2, 3, 4];
var irisan1 = arr.slice(1,3); // mengambil irisan array mulai dari index 1 hingga index 2 (sebelum index 3)
console.log(irisan1); // [1, 2]
var irisan2 = arr.slice(1);
console.log(irisan2); // [1, 2, 3, 4]
var irisan3 = arr.slice(2, 3);
console.log(irisan3); // [2]
var irisan4 = arr.slice(2, 2);
console.log(irisan4); // [] KOSONG!
```

## splice()

Splice adalah fungsi array yang menghapus dan/atau menambah nilai ke dalam array. Fungsi splice menerima satu, dua, atau lebih parameter.

Strukturanya adalah sebagai berikut:

```
array.splice([IndexMulai], [JumlahNilaiYangDihapus], [NilaiYangDitambahkan1], [NilaiYangDitambahkan2], ...)
```

IndexMulai = index array yang akan dimodifikasi. JumlahNilaiYangDihapus = jumlah nilai yang dihapus, dimulai dari IndexMulai. Misal, IndexMulai nya 0, dan JumlahNilaiYangDihapus adalah 2. Maka kita akan menghapus dua nilai mulai dari index 0, yaitu nilai di index 0 dan index 1. NilaiYangDitambahkan1, NilaiYangDitambahkan2, ... = nilai-nilai yang akan ditambahkan setelah IndexMulai.

Gambarannya dicontohkan sebagai berikut:

```
var arr = ["buku", "laptop", "komputer"];
arr.splice(2, 0, "televisi"); // Menghapus 0 nilai dari index 2, dan menambah 1 nilai yaitu "televisi" pada console.log(arr); // ["buku", "laptop", "televisi", "komputer"]

arr.splice(0, 2); // Menghapus 2 nilai dari index 0
console.log(arr); // ["televisi", "komputer"]

arr.splice(0, 1, "majalah", "koran"); // Menghapus 1 nilai dari index 0, dan menambah 2 nilai yaitu "majala console.log(arr); // ["majalah", "koran", "komputer"]
```

## split()

Split adalah fungsi yang memecah string dan mengembalikan nilai berupa array sesuai dengan separator atau pemisah tertentu yang didefinisikan. Fungsi split menerima satu parameter, yaitu karakter apa yang akan menjadi pemisah/separator.

```
var kalimat = "saya adalah full-stack javascript programmer!";
var kata = kalimat.split(" "); // kalimat dipecah dengan separator berupa spasi.
console.log(kata); // ["saya", "adalah", "full-stack", "javascript", "programmer!"]
```

## Multidimensional Array

Multidimensional array (array multidimensi) adalah sebuah array yang berisikan array lagi didalamnya. Konsepnya penggunaan array sama dengan array biasa (satu dimensi), hanya saja jumlah indeks yang digunakan saat mengakses nilai didalam array adalah sebanyak dimensi array tersebut.

Contoh array multidimensi antara lain diagram kartesius dan matriks. Diagram kartesius dan matriks adalah array multidimensi yang sama-sama membutuhkan koordinat untuk mengakses nilai didalamnya.

```
(y)
4 |           *
3 |           *
2 |   *
1 | *
+ - - - - (x)
0 1 2 3 4
```

Diagram kartesius adalah contoh array 2 dimensi. Setiap titik pada diagram di atas memiliki koordinat (x,y)

## berikut contoh array 2 dimensi pada JavaScript dan cara penggunaannya

```
// cara deklarasi array 2 dimensi kosong
var arr2D = [[]];

// contoh array 2 dimensi
var arr2D = [ [1,2], [3,4], [5,6] ];
var murid = [ ['Budi', 'SD 1 Cicayur'], ['Suci', 'SD 23 Beji'] ];

// cara mengakses nilai didalam array 2 dimensi
console.log(arr2D[0]); // [1,2]
console.log(arr2D[0][1]); // 2
console.log(murid[1]); // ['Suci', 'SD 23 Beji']
console.log(murid[1][1]); // 'SD 23 Beji'

// array 2 dimensi dengan built-in functions
arr2D.push([7,8]); // arr2D = [ [1,2], [3,4], [5,6], [7,8] ]
arr2D[1].push(0); // arr2D = [ [1,2], [3,4,0], [5,6], [7,8] ]
arr2D[0].pop(); // arr2D = [ [1], [3,4,0], [5,6], [7,8] ]
arr2D[2].pop(); // arr2D = [ [1], [3,4,0], [5], [7,8] ]
```