

ARRAY & OBJECTS

Batam 25 Feb, 2019

DEFINISI

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.

```
> var animals = ["Lion", "Tiger", "Puma"]
```

```
> animals
```

```
> animals[0]
```

```
> animals[3] = "Jaguar"
```

```
> animals[8] = "Leopard"
```

```
> animals[1] = undefined
```

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.



ARRAY BUILT IN FUNCTION

- 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

```
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"
```

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.

```
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
```

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).  
Index 3 tidak ikut diambil.  
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 adalah fungsi array yang menghapus dan/atau menambah nilai ke dalam array. Fungsi splice menerima satu, dua, atau lebih parameter.

Strukturnya 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.

```
var arr = ["buku", "laptop", "komputer"];

arr.splice(2, 0, "televisi"); // Menghapus 0 nilai dari index 2, dan menambah 1 nilai yaitu "televisi" pada
index 2.

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 "majalah"
dan "koran"

console.log(arr); // [majalah", "koran", "komputer"]
```

SPLIT

`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 penggunaannya sama dengan array biasa (satu dimensi), hanya saja jumlah indeks yang digunakan saat mengakses nilai didalam array adalah sebanyak dimensi dari array tersebut.

```
// 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] ]
```

LATIHAN : MELAKUKAN LOOPING PADA ARRAY

Buatlah sebuah fungsi dengan nama `dataHandling` dengan sebuah parameter untuk menerima argumen. Argumen yang akan diterima adalah sebuah array yang berisi beberapa array sejumlah n. Contoh input dapat dilihat dibawah:

```
var input = [  
    ["0001", "Badai", "Surabaya", "21/05/1989", "Nonton youtube", "perumahan citramas"],  
    ["0002", "Maenggar", "Medan", "10/10/2000", "Bermain Biola", "Nongsa"],  
    ["0003", "Maria", "Ambon", "25/12/2006", "Memasak", "Bungaraya "],  
    ["0004", "Hendri", "Jakarta", "6/4/1998", "Berkebun", "Botania"]  
];
```

Tugas kamu adalah mengimplementasikan fungsi `dataHandling` agar dapat menampilkan data-data pada dari argumen seperti di pada gambar di bawah ini:

answer

Nomor ID : 0001
nama lengkap : Badai
ttl : Surabaya 21/05/1989
hobby : Nonton youtube
alamat :perumahan citramas

Nomor ID : 0002
nama lengkap : Maenggar
ttl : Medan 10/10/2000
hobby : Bermain Biola
alamat :Nongsa

Nomor ID : 0003
nama lengkap : Maria
ttl : Ambon 25/12/2006
hobby : Memasak
alamat :Bungaraya

Nomor ID : 0004
nama lengkap : Hendri
ttl : Jakarta 6/4/1998
hobby : Berkebun
alamat :Botania

LOGIC CHALLENGE

Diberikan sebuah function `hitungJumlahKata(kalimat)` yang menerima parameter berupa string yang merupakan sebuah kalimat. Function akan mengembalikan jumlah kata dari kalimat tersebut. Contoh, "I have a dream" akan menghasilkan nilai 4, karena memiliki 4 kata.

```
function hitungJumlahKata(kalimat) {  
  // you can only write your code here!  
  
}  
  
// TEST CASES  
console.log(hitungJumlahKata('Glints Academy')); // 2  
console.log(hitungJumlahKata('Lorem Ipsum dolor sit amet')); //5  
console.log(hitungJumlahKata('Game of thrones is overatted')); // 5  
console.log(hitungJumlahKata('I')); // 1  
console.log(hitungJumlahKata('I love to eat' )); //4
```

answer

LATIHAN : MENGELOMPOKKAN ANGKA

`/*Diberikan sebuah function mengelompokkanAngka(arr) yang menerima satu parameter berupa array yang berisi angka-angka.`

`Function akan me-return array multidimensi dimana array tersebut berisikan 3 kategori/kelompok:`

`kelompok pertama (baris pertama) merupakan angka-angka genap`

`kelompok ke-2 (baris ke-2) merupakan angka-angka ganjil`

`kelompok ke-3 (baris ke-3) merupakan angka-angka kelipatan 3`

`Contoh jika arr inputan adalah [45, 20, 21, 2, 7] maka output: [[20, 2], [7], [45, 21]]*/`

```
function mengelompokkanAngka(arr) {
```

```
  // you can only write your code here!Answer
```

```
}
```

```
// TEST CASES
```

```
console.log(mengelompokkanAngka([2, 4, 6])); // [ [2, 4], [], [6] ]
```

```
console.log(mengelompokkanAngka([1, 2, 3, 4, 5, 6, 7, 8, 9])); // [ [ 2, 4, 8 ], [ 1, 5, 7 ], [ 3, 6, 9 ] ]
```

```
console.log(mengelompokkanAngka([100, 151, 122, 99, 111])); // [ [ 100, 122 ], [ 151 ], [ 99, 111 ] ]
```

```
console.log(mengelompokkanAngka([])); // [ [], [], [] ]
```

BOTOLCHECKER

botolChecker adalah sebuah function yang menerima satu parameter berupa array of string. function tersebut akan menghitung jumlah botol yang penuh (F), setengah penuh (H) dan botol kosong (E) dan function akan mereturn jumlah masing2 jenis botol dan total botol secara keseluruhan.

Adapun ketentuannya:

Full (F) = 1

Half (H) = 1/2

Empty (E) = 0

Answer

```
function botolChecker(array) {  
  //write your code here  
}  
  
console.log(botolChecker(['F','H','F']));  
// jumlah F: 2, jumlah H: 1, jumlah E: 0. Dan totalnya  
adalah: 2.5  
  
console.log(botolChecker(['H','H','F','F','E']));  
// jumlah F: 2, jumlah H: 2, jumlah E: 1. Dan totalnya  
adalah: 3  
  
console.log(botolChecker(['E','F','F','F','H']));  
// jumlah F: 3, jumlah H: 1, jumlah E: 1. Dan totalnya  
adalah: 3.5  
  
console.log(botolChecker(['E','H']));  
// jumlah F: 0, jumlah H: 1, jumlah E: 1. Dan totalnya  
adalah: 0.5  
  
console.log(botolChecker(['H','H','H','H']));  
// jumlah F: 0, jumlah H: 4, jumlah E: 0. Dan totalnya  
adalah: 2  
  
console.log(botolChecker([])); //NO DATA
```

OBJECT

JavaScript merupakan bahasa pemrograman yang berbasis simple-object (Objek sederhana). Objek adalah kumpulan tidak berurut yang merangkai beberapa property dan property memiliki nama/key dan value (key-value pairs).

Objek dalam JavaScript, sama seperti banyak bahasa pemrograman lainnya, bisa dibandingkan dengan objek dalam kehidupan nyata.

Untuk membuat sebuah object literal bisa dengan cara menuliskan kurung kurawal (curly braces) kemudian menuliskan nama property yang harus memiliki keyName dan value.

```
var myObj = {  
  myKey: 'myValue'  
};
```



```
var supermanObj = {  
  id: "1a2b3c",  
  name: "Superman",  
  age: 200,  
  favorites: [  
    "coding",  
    "reading",  
    {  
      sports: ["parkour", "hill climbing"]  
    }  
  ],  
  address: {  
    street: "Planet Krypton",  
    zipCode: 54213  
  }  
};
```

```
console.log(supermanObj.name); // "Superman"  
console.log(supermanObj.age); // 200  
console.log(supermanObj.favorites[0]); //  
"coding"  
console.log(supermanObj.favorites[2].sports); //  
["parkour", "hill climbing"]  
console.log(supermanObj.favorites[2].sports[0]);  
// "parkour"  
console.log(supermanObj.address); // {street:  
"Planet Krypton", zipCode: 54213}  
console.log(supermanObj.address.zipCode); //54213
```

