

Front End Bootcamp

PT. Mahardika Solusi Teknologi

Ariel Arliyanus, S.Kom

Chapter 6 - JavaScript

2021 © All Right Reserved

Dilarang memperbanyak dan/atau meng-copy sebagian atau seluruh material dalam dokumen ini tanpa persetujuan tertulis dari PT. Mahardika Solusi Teknologi

Chapter 6 JavaScript

Javascript merupakan salah satu bahasa pemrograman yang terpopuler di dunia. Javascript membuat aplikasi web menjadi interaktif tanpa reload halaman. Hampir semua browser modern yang kita ketahui dapat menjalankan javascript agar website yang ditampilkan lebih menarik.

Menjalankan Javascript

Ada bermacam cara untuk menjalankan kode javascript yang kita buat, di antaranya:

1. Menjalankan javascript di console browser
2. Menjalankan javascript di tools online seperti jsbin.com
3. Menjalankan javascript dengan nodejs

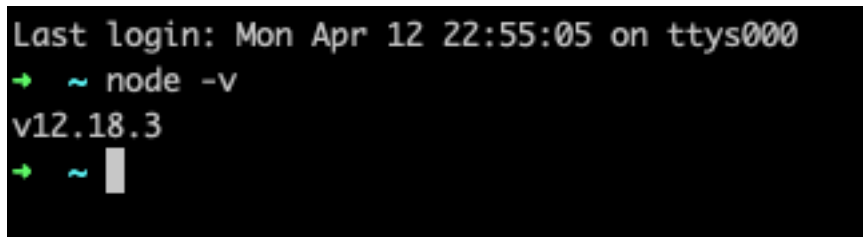
Pada materi ini kita akan mencoba untuk menjalankan script dengan nodejs.

Nodejs adalah software berbasis pemrograman javascript yang dijalankan di sisi server. Jika biasanya javascript kita kenal erat kaitannya dengan client/browser tapi dengan nodejs ini kita bisa membangun server menggunakan bahasa javascript.

Install Nodejs

Pertama-tama tentu kita harus menginstall terlebih dahulu nodejs di komputer kita. Berikut ini link untuk download [nodejs](https://nodejs.org/) (disarankan memilih versi LTS). Untuk OS Windows dan macOS tinggal diikuti saja instalasinya sampai selesai, sedangkan untuk Ubuntu 18.04 kamu bisa install mengikuti [tutorial dari digitalocean](#).

Untuk mengecek apakah instalasi nodejs sudah berhasil kita bisa jalankan script di terminal kita:

A terminal window with a black background and green text. The first line shows the login message: "Last login: Mon Apr 12 22:55:05 on ttys000". The second line shows the command "node -v" being entered. The third line shows the output "v12.18.3". The fourth line shows the prompt character "~" followed by a cursor.

```
Last login: Mon Apr 12 22:55:05 on ttys000
➔ ~ node -v
v12.18.3
➔ ~
```

dengan perintah tersebut, diketahui saat ini terinstall node dengan versi 12.16.2 dan npm versi 6.14.4 (untuk versi bisa menggunakan yang terbaru ini hanya contoh untuk mengecek version nya)

Hello world

Untuk menjalankan javascript dengan nodejs mari kita coba dengan menuliskan script di sebuah file index.js .

Buatlah file dengan nama index.js kemudian tulis code di dalam file tersebut seperti berikut:

```
var sayHello = "Hello World!"  
console.log(sayHello)
```

kemudian simpan file tersebut (save). Setelah itu kita bisa menjalankan script pada index.js tersebut dengan memberikan perintah seperti berikut pada terminal:

```
$ node index.js
```

secara otomatis pada terminal kita akan muncul "Hello World!" . (Selamat Anda telah berhasil menjalankan program pertama Anda ^_^)

Demikian cara untuk menjalankan Javascript dengan nodejs.

Variable

Variable adalah suatu blok data atau wadah untuk menampung sekumpulan data dengan berbagai tipe data apapun. Dengan variable kita bisa menyimpan suatu nilai untuk kemudian kita olah kembali pada program kita. Untuk deklarasi variable dalam javascript kita bisa gunakan sintaks **var** lalu diikuti nama variabelnya.

```
var name = "John" // Tipe  
var angka = 12  
var todayIsFriday = false
```

```
console.log(name) // "John"  
console.log(angka) // 12  
console.log(todayIsFriday) // false  
Waspada! deklarasi variable yang tidak bernilai !
```

```
var items  
console.log(items) // Undefined
```

Data Type

Data Type atau dalam bahasa indonesia Tipe Data adalah sekumpulan informasi yang memiliki nilai dan karakteristik tertentu. Beberapa contoh tipe data pada javascript di antaranya:

1. **Integer**: tipe data yang berupa angka
2. **String** : tipe data berupa text atau kumpulan karakter, biasanya string dibungkus dalam tanda petik ganda (double quote) atau tanda petik tunggal (single quote).
3. **Boolean**: tipe data dengan nilai **true** atau **false**

String

String adalah tipe data yang berisi karakter-karakter dibungkus dalam tanda petik ("" atau ''). Karakter-karakter pada suatu string dapat diakses dengan menggunakan indeks atau posisi karakter berada. Indeks pada string selalu mulai dari 0.

```
var sentences = "Javascript"
console.log(sentences[0]) // "J"
console.log(sentences[2]) // "v"
```

String pada javascript juga memiliki property dan methods tertentu. Property dan methods tersebut bisa kita gunakan dalam memanipulasi data agar sesuai dengan program yang kita inginkan.

String Properties

.length

mengembalikan panjang atau jumlah karakter pada suatu string.

```
var word = "Javascript is awesome"
console.log(word.length) // 21
```

String Methods

.charAt([indeks])

Mengembalikan karakter pada indeks yang diinginkan

```
console.log('i am a string'.charAt(3)); // 'm'
.concat([string])
```

Menggabungkan beberapa string dan mengembalikannya menjadi string baru.

```
var string1 = 'good';
var string2 = 'luck';
console.log(string1.concat(string2)); // goodluck
.indexOf([string/karakter])
```

Mengembalikan indeks dari string/karakter yang dicari, yang pertama kali ditemukan, atau -1 apabila tidak ditemukan.

```
var text = 'dung dung ces!';
console.log(text.indexOf('dung')); // 0
console.log(text.indexOf('u')); // 1
console.log(text.indexOf('jreng')); // -1
.substring([indeks awal], [indeks akhir (optional)])
```

Mengembalikan potongan string mulai dari indeks pada parameter pertama (indeks awal) sampai dengan indeks pada parameter kedua (indeks akhir). Bila parameter kedua tidak ditentukan, maka secara otomatis berakhir pada karakter terakhir. Karakter pada indeks yang ditentukan pada parameter kedua tidak diikuti sebagai output.

```
var car1 = 'Lykan Hypersport';
var car2 = car1.substr(6);
console.log(car2); // Hypersport
```

.substr([indeks awal], [jumlah karakter yang diambil (optional)])

Mendapatkan potongan string mulai dari indeks pada parameter pertama (indeks awal) dengan jumlah indeks pada parameter kedua (jumlah karakter). Bila parameter kedua tidak ditentukan, maka secara otomatis berakhir pada karakter terakhir. Karakter pada indeks yang ditentukan pada parameter kedua tidak diikuti sebagai output.

```
var motor1 = 'zelda motor';  
var motor2 = motor1.substr(2, 2);  
console.log(motor2); // ld
```

.toUpperCase()

Mengembalikan string baru dengan semua karakter yang diubah menjadi huruf kapital.

```
var letter = 'This Letter Is For You';  
var upper = letter.toUpperCase();  
console.log(upper); // THIS LETTER IS FOR YOU
```

.toLowerCase()

Mengembalikan string baru dengan semua karakter yang diubah menjadi huruf kecil

```
var letter = 'This Letter Is For You';  
var lower = letter.toLowerCase();  
console.log(lower); // this letter is for you
```

.trim()

Mengembalikan string baru yang sudah dihapus karakter *whitespace* (" ") pada awal dan akhir string tersebut.

```
var username = ' administrator '  
var newUsername = username.trim();  
console.log(newUsername) // 'administrator'
```

Mengubah tipe data dari dan ke String

Di Javascript terkadang kita ingin mengubah sebuah data string menjadi tipe data lain atau sebaliknya. Contoh diperoleh data angka tapi dalam tipe data String maka kita dapat mengubah string tersebut menjadi tipe data angka.

String([angka/array])

Fungsi global **String()** dapat dipanggil kapan saja pada program JavaScript dan akan mengembalikan data dalam tipe data String dari parameter yang diberikan.

```
var int = 12;  
var real = 3.45;  
var arr = [6, 7, 8];  
  
var strInt = String(int);  
var strReal = String(real);  
var strArr = String(arr);  
  
console.log(strInt); // '12'  
console.log(strReal); // '3.45'  
console.log(strArr); // '6,7,8'
```

.toString()

Mengonversi tipe data lain menjadi string. Bila data tersebut adalah array, setiap nilai akan dituliskan dan dipisah dengan karakter koma.

```
var number = 21;
console.log(number.toString()); // '21'
var array = [1,2];
console.log(array.toString()); // '1,2'
Number([String])
```

Fungsi global **Number()** mengonversi tipe data string menjadi angka. Data yang diberikan pada parameter harus berupa karakter angka saja, dengan titik (separator) bila angka adalah bilangan desimal. Bila parameter berisi karakter selain angka dan/atau titik, **Number()** akan mengembalikan **NaN (Not a Number)**.

```
var number1 = Number("90"); // number1 = 90
var number2 = Number("1.23"); // number2 = 1.23
var number3 = Number("4 5"); // number3 = NaN
parseInt([String]) dan parseFloat([String])
```

Fungsi global **parseInt([String])** dan **parseFloat([String])** mengembalikan angka dari string. Bila angka adalah bilangan desimal maka gunakan **parseFloat()**, bila tidak bilangan dibelakang koma akan diabaikan.

```
var int = '89';
var real = '56.7';
var strInt_1 = parseInt(int); // strInt_1 = 89
var strInt_2 = parseInt(real); // strInt_2 = 56
var strReal_1 = parseFloat(int); // strReal_1 = 89
var strReal_2 = parseFloat(real); // strReal_2 = 56.7
```

Operator

Operator adalah karakter khusus yang merepresentasikan sebuah tindakan. Operator terbagi ke dalam beberapa jenis:

1. Operator Aritmatika Operator yang melibatkan operasi matematika seperti tambah, kurang, kali, bagi.
 - o Tambah (+)
 - o Kurang (-)
 - o Kali (*)
 - o Bagi (/)
 - o Modulus (%)Modulus adalah sisa bagi. Contohnya $5\%3$ hasilnya adalah 2, $100\%5$ hasilnya 0.
2. Operator Assignment (=), Operator untuk mendaftarkan atau meng-assign suatu nilai ke dalam suatu variable

```
var angka
angka = 10 // Contoh assignment variable angka dengan nilai 10
```

3. Operator Perbandingan, Operator yang membandingkan suatu nilai dengan nilai yang lain. Hasil dari perbandingan ini akan dikembalikan dalam tipe data boolean **true** atau **false**.
 - o Equal Operator (==)

```
var angka = 100
console.log(angka == 100) // true
console.log(angka == 20) // false
```
 - o Not Equal (!=)

```
var sifat = "rajin"
console.log(sifat != "malas") // true
console.log(sifat != "bandel") //true
```
 - o Strict Equal (===) Selain membandingkan dua nilai nya, strict equal juga membandingkan tipe datanya apakah sama atau tidak

```
var angka = 8
console.log(angka == "8") // true, padahal "8" adalah string.
console.log(angka === "8") // false, karena tipe data nya berbeda
console.log(angka === 8) // true
```
 - o Strict not Equal (!==) Kebalikan dari strict equal.

```
var angka = 11
console.log(angka != "11") // false, padahal "11" adalah string
console.log(angka !== "11") // true, karena tipe datanya berbeda
console.log(angka !== 11) // false
```
 - o Kurang dari & Lebih Dari (<, >, <=, >=)

```
var number = 17
console.log( number < 20 ) // true
console.log( number > 17 ) // false
console.log( number >= 17 ) // true, karena terdapat sama dengan
console.log( number <= 20 ) // true
```

4. Operator Kondisional, Operator yang mengkombinasikan dua nilai kebenaran . Terdapat operator AND (&&) dan OR (||)

- OR (||)

```
console.log(true || true); // true
console.log(true || false); // true
console.log(true || false || false); // true
console.log(false || false); // false
```

- AND (&&)

```
console.log(true && true); // true
console.log(true && false); // false
console.log(false && false); // false
console.log(false && true && true); // false
console.log(true && true && true); // true
```


Tugas

1. Menambahkan Folder Baru dan file baru

Gunakanlah project yang sama dengan Tugas-Bootcamp (**jangan buat repository baru lagi**). Lalu buatlah folder baru didalam folder "gitlab" dengan nama "Tugas-JS-1", lalu buat file baru di dalam folder tersebut dengan nama "tugas.js".

sehingga struktur foldernya seperti di bawah ini

```
Tugas-Bootcamp/  
  Tugas-JS-1/  
  Tugas-JS-2/  
  Tugas-JS-3/
```

2. Kerjakan Soal di bawah ini

kerjakan soal di bawah ini dan masukkan semua kode jawabannya ke dalam file "tugas3.js"

kerjakan soal ini **tanpa** menggunakan **looping**, **function** dan kode-kode **ES6** (bagi yang sudah paham ES6), jadi kerjakan dengan kode-kode sesuai materi

soal 1

buatlah variabel-variabel seperti di bawah ini

```
var kataPertama = "saya";  
var kataKedua = "senang";  
var kataKetiga = "belajar";  
var kataKeempat = "javascript";  
gabungkan variabel-variabel tersebut agar menghasilkan output
```

saya Senang belajar JAVASCRIPT

soal 2

buatlah variabel-variabel seperti di bawah ini

```
var panjangPersegiPanjang = "8";  
var lebarPersegiPanjang = "5";  
  
var alasSegitiga= "6";  
var tinggiSegitiga = "7";  
ubah lah variabel diatas ke dalam integer dan gunakan pada operasi perhitungan dari keliling  
persegi panjang dan luas segitiga dengan variabel di bawah ini:
```

```
var kelilingPersegiPanjang;  
var luasSegitiga;
```

lalu tampilkan dengan console.log

soal 3

buatlah variabel-variabel seperti di bawah ini

```
var sentences= 'wah javascript itu keren sekali';

var firstWord= sentences.substring(0, 3);
var secondWord; // do your own!
var thirdWord; // do your own!
var fourthWord; // do your own!
var fifthWord; // do your own!

console.log('Kata Pertama: ' + firstWord);
console.log('Kata Kedua: ' + secondWord);
console.log('Kata Ketiga: ' + thirdWord);
console.log('Kata Keempat: ' + fourthWord);
console.log('Kata Kelima: ' + fifthWord);
selesaikan variabel yang belum diisi dan hasilkan output seperti berikut:
```

```
Kata Pertama: wah
Kata Kedua: javascript
Kata Ketiga: itu
Kata Keempat: keren
Kata Kelima: sekali
```

soal 4

buatlah variabel seperti di bawah ini

```
var nilaiJohn = 80;
var nilaiDoe = 50;
```

tentukan indeks nilai dari **nilaiJohn** dan **nilaiDoe** (tampilkan dengan console.log) dengan kondisi :

```
nilai >= 80 indeksnya A
nilai >= 70 dan nilai < 80 indeksnya B
nilai >= 60 dan nilai < 70 indeksnya c
nilai >= 50 dan nilai < 60 indeksnya D
nilai < 50 indeksnya E
```

soal 5

buatlah variabel seperti di bawah ini

```
var tanggal = 22;
var bulan = 7;
var tahun = 2020;
ganti tanggal ,bulan, dan tahun sesuai dengan tanggal lahir anda dan buatlah switch case pada
bulan, lalu muncul kan string nya dengan output seperti ini 22 Juli 2020 (isi di sesuaikan dengan
tanggal lahir masing-masing)
```

3. Tips

Agar soal mudah di periksa trainer maka sebelum jawaban soal sisipkan komentar seperti di bawah ini

```
// soal 1  
.... jawaban soal 1
```

```
// soal 2  
.... jawaban soal 2
```

dan seterusnya

4. Kumpulkan Tugas

Update tugasnya ke repository masing-masing.