



[Beranda](#) / [Academy](#) / [Belajar Dasar Pemrograman Web](#) / Tipe Data

! Pembaruan! Modul ini dibuat pada tanggal 6 December 2019. Pembaruan terakhir adalah: **Konversi potongan kode menjadi kode interaktif..**

[Lihat riwayat »](#)

Tipe Data

Nilai yang kita tetapkan pada variabel pasti memiliki tipe data. Tipe data merupakan pengklasifikasian data berdasarkan jenis data tersebut. Pada JavaScript terdapat beberapa tipe data sebagai berikut:

Undefined

Tipe data ini terbentuk ketika sebuah variabel tidak memiliki nilai, dalam arti lain ketika kita mendeklarasikan variabel tanpa menginisialisasikan nilainya, maka variabel tersebut menjadi **undefined** . Contoh:

main.js +

```
1 let x;  
2 console.log(typeof(x));  
3  
4 /* output: undefined */
```

INPUT

RESET

JALANKAN

Output:

Pada contoh kode di atas, kita mendeklarasikan variabel x, tetapi kita tidak menginisialisasikan dengan nilai apapun. Ketika kita memastikan tipe data dengan menggunakan fungsi **typeof()** maka menghasilkan output undefined.

Fungsi typeof() digunakan untuk memastikan tipe data pada variabel dengan mengembalikan tipe data tersebut dalam bentuk teks

Numbers

Nilai dari tipe data number adalah angka. JavaScript variabel bertipe data number dituliskan seperti ini:

```
1. let x = 10;
```





main.js +

```
1 let x = 10;
2 console.log(typeof(x))
3
4 /* output: number */
5
6 let y = 17.25;
7 console.log(typeof(y))
8
9 /* output: number */
```

☐ INPUT RESET

JALANKAN

Output:

Pada tipe data number kita juga dapat melakukan perhitungan aritmatika seperti penambahan, pengurangan, perkalian, dll. Berikut operator yang dapat kita gunakan dalam melakukan perhitungan aritmatika pada tipe data *number*:

| Operator | Fungsi | Contoh |
|----------|-----------------|--------------|
| + | Penambahan | 10 + 10 = 20 |
| - | Pengurangan | 15 - 7 = 8 |
| / | Pembagian | 21 / 7 = 3 |
| * | Perkalian | 9 * 9 = 81 |
| % | Sisa hasil bagi | 5 % 2 = 1 |

main.js +

```
1 let a = 12;
2 let b = 9;
3
4 console.log(a + b)
5 console.log(a - b)
6 console.log(a * b)
7 console.log(a / b)
8 console.log(a % b)
9
10 /* output:
11 21
12 3
13 108
```

☐ INPUT RESET





Pada operator aritmatika juga terdapat operator *increment* (++) dan *decrement* (--). Operator *increment* dan *decrement* digunakan untuk menambahkan atau mengurangi nilai 1 pada nilai variabel yang ada sekarang.

Operator ini dapat dituliskan sebelum atau sesudah variabel, tetapi hal tersebut bukan berarti sama. Berikut ketentuannya:

- Jika dituliskan setelah variabel (x++), *statement* akan menghasilkan nilai variabel sebelum ditingkatkan nilainya.
- Jika dituliskan sebelum variabel (++x), *statement* akan menghasilkan nilai variabel setelah ditingkatkan nilainya.

Lebih jelasnya berikut contoh kode dalam penerapan operator tersebut, perhatikan hasil yang di dapat.

main.js +

```
1 /* Increment dan Decrement */
2
3 let postfix = 5;
4 console.log(postfix++);
5 /* output: 5 */
6
7 let prefix = 5;
8 console.log(++prefix);
9 /* output: 6 */
```

INPUT

RESET

JALANKAN

Output:

Strings

Tipe data selanjutnya adalah strings, String ini dasarnya adalah sebuah teks. Pada JavaScript untuk menetapkan nilai string pada variabel, gunakan tanda *single* (') atau *double quote* (") di antara teksnya. Contohnya:

main.js +

```
1 let greet = "Hello";
2 console.log(typeof(greet))
3
4 /* output: string */
```

INPUT

RESET



Output:



Apa pun yang berada di antara tanda *double quote* atau *single quote* merupakan sebuah string, baik itu angka, huruf maupun karakter spesial.

Pada string juga kita dapat menggunakan operator *plus* (+). Operator tersebut pada string berfungsi untuk menggabungkan dua teks yang terpisah menjadi satu buah teks. Contohnya seperti ini:

main.js +

```
1 let greet = "Hello";
2 let moreGreet = greet + greet;
3 console.log(moreGreet);
4
5 /* output: HelloHello */
```

INPUT

RESET

JALANKAN

Output:

Boolean

Boolean hanya dapat memiliki dua nilai, yakni **true** atau **false**. Tipe data ini menjadi kunci utama dalam penentuan logika, kita akan memahaminya nanti ketika pembahasan *if/else statement*. Untuk menetapkan nilai **boolean** pada variabel kita bisa menggunakan keyword **true** atau **false**.

main.js +

```
1 let x = true;
2 let y = false;
3
4 console.log(typeof(x))
5 console.log(typeof(y))
6
7 /* output:
8 boolean
9 boolean
10 */
```

INPUT

RESET

JALANKAN

Output:



DIBANTU

main.js +

```
1 const a = 10;
2 const b = 12;
3
4 let isGreater = a > b;
5 let isLess = a < b;
6
7 console.log(isGreater);
8 console.log(isLess);
9
10 /* output:
11 false
12 true
13 */
```

☐ INPUT

RESET

JALANKAN

Output:

Null

Yang terakhir adalah **null**. Serupa dengan **undefined**, namun **null** perlu diinisialisasikan pada variabel. **null** biasa digunakan sebagai nilai sementara pada variabel, tapi sebenarnya nilai tersebut “tidak ada”.

Terkadang kita perlu membuat sebuah variabel, namun kita belum memerlukan nilai apa apa dan tidak ingin terikat oleh tipe data apapun. Nah, daripada kita tidak menetapkan nilai apapun (variabel akan **undefined**) sebaiknya kita beri nilai **null** pada variabel tersebut, dan ubah nanti ketika kita membutuhkannya.

Untuk menetapkan **null** pada variabel, kita dapat gunakan *keyword* **null** ketika variabel tersebut diinisialisasikan.

main.js +

```
1 let someLaterData = null;
2 console.log(someLaterData);
3
4 /* output:
5 null
6 */
```

☐ INPUT

RESET

JALANKAN

Output:



DIBANTU



- 1. `let` `x`; *// x merupakan undefined*
- 2. `x = 1` *// sekarang x merupakan number*
- 3. `x = true` *// sekarang x merupakan boolean*
- 4. `x = "Harry"` *// sekarang x merupakan string*

← KEMBALI KE MATERI SEBELUMNYA

LANJUTKAN KE MATERI BERIKUTNYA →



Dicoding Space
Jl. Batik Kumeli No.50, Sukaluyu,
Kec. Cibeunying Kaler, Kota Bandung
Jawa Barat 40123



Penghargaan



Decode Ideas
Discover Potential

➤ [Tentang Kami](#)

[Blog](#),
[Reward](#)
[Showcase](#)

[Hubungi Kami](#)
[FAQ](#)

