**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:

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

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;

Jika angka tersebut merupakan sebuah bilangan desimal, maka kita bisa gunakan tanda titik pada pecahan bilangannya.

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

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 |

1. let a = 12;
2. let b = 9;
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)
10. */\* output:*
11. *21*
12. *3*
13. *108*
14. *1.3333333333333333*
15. *3*

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 mengembalikan nilai variabel saat ini sebelum ditingkatkan nilainya.
* Jika dituliskan sebelum variabel (++x), *statement* akan mengembalikan nilai variabel setelah ditingkatkan nilainya.

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

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

**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:

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

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:

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

**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.

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

Atau kita bisa gunakan operator komparasi seperti lebih dari (>) atau kurang dari (<). Contohnya:

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

**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.

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

Meskipun terdapat beberapa tipe data dalam JavaScript, tetapi variabel pada JavaScript memiliki sifat tipe data yang dinamis. Artinya, kita dapat memberikan tipe data yang berubah-ubah pada satu variabel yang sama. Contohnya:

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*

Berikut snippet code dari contoh kode yang digunakan pada materi di atas, Anda bisa mencobanya sendiri dengan berbagai macam tipe data pada JavaScript dan menampilkannya pada *console*.

<https://glot.io/snippets/fhzkayhcom>