

Koding dan Kopi Siri 2

Javascript 101

Bismillah.

Dah lama juga abam tak menulis, seingatnya last keluar masa zaman floppy disk mula-mula diperkenalkan. Masa tu orang pakai magnetic disk lagi. Oleh disebabkan itu, abam nak mula dengan Javascript 101 atau orang kampung abam selalu panggil *introduction to javascript*.

Dynamic & Weak Type Language

Dynamic type ni maksudnya type sesuatu variable itu disemak sewaktu runtime (code execution)

```
let foo = 42; // foo is now a number
foo = "bar"; // foo is now a string
foo = true; // foo is now a boolean
```

Weak type pula maksudnya "implicit type conversion" atau dalam bahasa jerman : "dia boleh agak-agak je type apa yang kita cuba maksudkan bila kita buat sesuatu proses".

```
const foo = 12; // foo is a number
const result = foo + "1"; // number + string
console.log(result); // 121 - result is now string
```

Datatype & data structure

Contoh primitive type:

```
null, undefined, Boolean, Number, BigInt, String, Symbol
```

contoh object type (data structure)

```
Object, Array, Date, Map, Set, WeakMap, WeakSet, Error, RegExp, Function
```

*Map & Set ni bukannya Array method yang kalian selalu guna untuk return data baru tau.

Declaration

Var: global scope

```
var greeting = "Mochi mochi";
var times = 4;

if (times > 3) {
  var hello = "Hallu";
  console.log(hello); // "Hallu"
}

console.log(hello); // output: Hallu
```

let/const: block scope

```
let greeting = "Moshi moshi";
let times = 4;

if (times > 3) {
  let hello = "Hallu";
  const world = "World";
  console.log(hello); // "Hallu"
  console.log(times) // 4
}
console.log(hello); // hello is not defined
console.log(world); // world is not defined
```

untuk var atau let kita boleh reassign value tapi tidak untuk const

```
let str2 = "String";
str2 = "Another String"; // ok

const str = "String";
str = "Another String"; // error
```

cara untuk declare object dan retrieve value

```
const obj = {
  foo: "bar",
};
console.log(obj.foo); // output: bar
console.log(obj["foo"]); // output: bar
const foo = "foo";
console.log(obj[foo]); // output: bar
```

cara untuk declare dan retrieve array

```
const arr = [1, 2, 3];
arr[0] = 4;
console.log(arr[0]); // output: 4
const n = 1;
console.log(arr[n]); // output: 2
```

const declaration untuk object type

```
const obj = {
  foo: "bar",
  biz: "buz",
};
obj.biz = "baz"; // ok

obj = {
  foo: "bar",
}; // error
```

```
const arr = [1, 2, 3];  
arr[0] = 4; // ok
```

```
arr = [1, 2, 3]; // error
```

kenapa kita dah declare const tapi data dia boleh berubah?

Pass by value & pass by reference

Primitives are passed by value, and Objects are passed by "copy of a reference".

```
// pass by value  
let a = 1;  
let b = a;  
b = 2;  
console.log(a); // 1  
console.log(b); // 2  
  
// pass by reference  
let c = { value: 1 };  
let d = c;  
d.value = 2;  
console.log(c.value); // 2  
console.log(d.value); // 2
```

Hoisting of "var" and "function"

Hoisting bermaksud variable & function declaration akan dipindahkan ke atas di dalam scopenya sebelum execution

```
// var hoisting utk file js yang bukan strict mode/module  
console.log(greeter); // say hello  
var greeter = "say hello";  
console.log(func); // hello  
function func() {  
  return "hello";  
}
```

Rasa macam setakat ini sahaja dulu abam menulis/menaip. Nah hadiah untuk sesiapa yang habis baca:

Non-zero value



null



0



undefined



Artikel ini disediakan oleh:

Shah Erhan
(keyboard warrior)