

Lesson 2: Data Types in JavaScript

Variable Declaration: var, let and const

var	let	const
var has the function or global scope.	let's have the block scope.	const variable has the block scope.
It gets hoisted to the top of its scope and initialized undefined.	It also got hoisted to the top of its scope but didn't initialize.	It also got hoisted to the top of its scope but didn't initialize.
It can be updated or re-declared.	It can only be updated and can't be re-declared.	It can't be updated or re-declared.
It's an old way to declare a variable.	It's a new way to declare variables introduced in ES6.	It's also a new way to declare a variable, which introduces in ES6.

Number

Declaring Integer type variable in JavaScript :-

```
var x = 23;  
let x = 20;
```

```
> a = 20  
20  
> b = 5.7894  
5.7894  
> typeof(a)  
'number'  
> typeof(b)  
'number'  
>
```

JavaScript has a single number type. Internally, it is represented as 64-bit floating point, the same as Java's double. Unlike most other programming languages, there is no separate integer type, so 1 and 1.0 are the same value. This is a significant convenience because problems of overflow in short integers are completely avoided, and all you need to know about a number is that it is a number. A large class of numeric type errors is avoided.

- Ref: Douglas Crockford - JavaScript_The Good Parts (2008)

Number

```
(base) ashish@ashish:~/Desktop$ nodejs
Welcome to Node.js v12.22.9.
Type ".help" for more information.
> 1 == 1.0
true
> a = 1
1
> b = 1.0
1
> typeof(a)
'number'
> typeof(b)
'number'
>
```

```
(base) ashish@ashish:~/Desktop$ python
Python 3.9.13 (main, Aug 25 2022, 23:26:11)
[GCC 11.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "quit()"
>>> 1 == 1.0
True
>>> a = 1
>>> b = 1.0
>>> type(a)
<class 'int'>
>>> type(b)
<class 'float'>
>>>
>>>
```

Strings

Declaring String variable in JavaScript :-

// Using string template

```
var variable = "Gokul Krishna"
```

// Using String class

```
var variable = new String("Gokul Krishna");
```

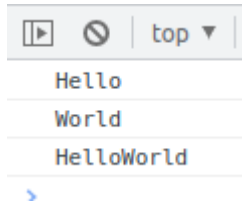
String Concatenation: Part 1

```
>  
> a = "Gokul Krishna"  
'Gokul Krishna'  
> a.length  
13  
> b = " and Dipendra"  
' and Dipendra'  
> a + b  
'Gokul Krishna and Dipendra'  
> c = a + b  
'Gokul Krishna and Dipendra'  
> c.length  
26  
>
```

String Concatenation: Part 2

home > ashish > Desktop > mypage.html > ...

```
1  <script>
2
3  var a = "Hello"
4  var b = "World"
5
6  console.log(a)
7  console.log(b)
8  console.log(a+b)
9  </script>
10
```



Boolean in JavaScript

```
let x = true;  
let y = false;  
let result = x.toString();  
console.log(result);
```

```
>  
> x = true  
true  
> y = false  
false  
> typeof(x)  
'boolean'  
>  
> x = 'true'  
'true'  
> typeof(x)  
'string'  
>
```

In Python:

```
>>> x = True  
>>> type(x)  
<class 'bool'>  
>>>
```

Arrays

```
var arr_num = [1, 2, 3, 0, -5, -2]
```

```
let arr_names = ['Dip', 'Gokul', 'Krishna', 'Ashish']
```

```
let arr_of_mixed_types = [1, 'Krishna', true]
```

To name a few operations that we can do on an array:

(1) Traversing or 'Iterating over the elements of an array'

(2) Sort

(3) Search

(4) Insertion

(5) Deletion

(6) Reversing

Hello, Arrays in Code!

```
var arr_num = [1, 2, 3, 0, -5, -2]
```

```
console.log(arr_num)
```

```
console.log(arr_num.sort()) // Changes the actual object. Return type: modified array.
```

```
console.log(arr_num.reverse()) // Changes the actual object.
```

```
// These operations sort() and reverse() change the actual object.
```

```
// The push() method adds new items to the end of an array.
```

```
arr_num.push(10)
```

```
console.log(arr_num)
```

Date

Using new Date()

new Date() without arguments, creates a date object with the current date and time:

```
const d = new Date();
```

Fri Aug 18 2023 16:49:21 GMT+0530 (India Standard Time)

Thank you!