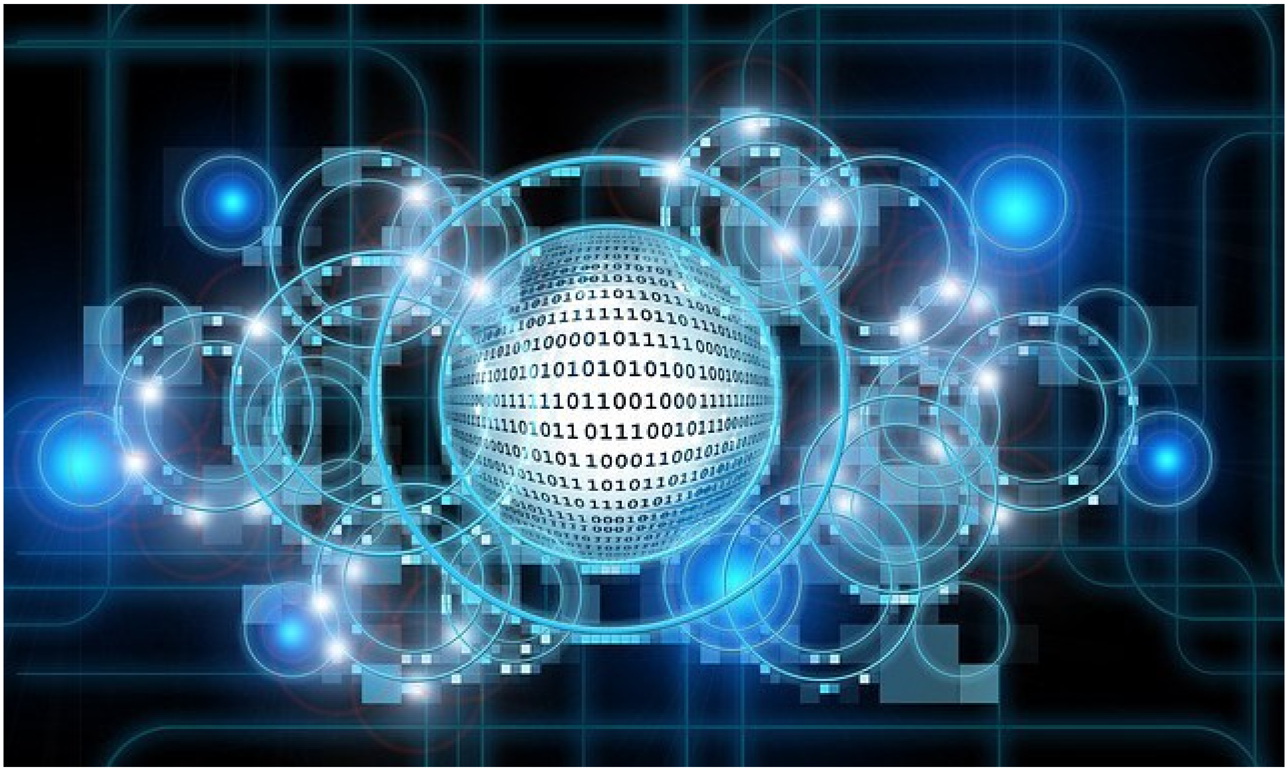
# Difference between null, undefined and not defined in javascript

Saurabh Mhatre Follow
Oct 22, 2018 · 2 min read



In today's post we are going to cover a question that is asked quite often in javascript interviews and technical rounds. Javascript has three ways to describe values which don't exist or should not exist in general.

**Undefined:-**

There are certain cases when undefined value is returned in javascript as follows:-

1)Whenever we declare a variable without assigning any value to it, javascript implicitly assigns its value as undefined.

```
let name;
console.log(name); //undefined
```

2) When value is not assigned in array or object

```
let numArray = [1,2,,4];
console.log(numArray);
//[1, 2, empty, 4]

typeof(numArray[2])
//"undefined"
```

3) When functions don’t have a return statement but are called for assigning a value to a variable.

```
let add = (a,b) => {
  let c = a+b;
  //return c;
}

let sum = add(2,3);

console.log(sum);
//Output: undefined
```

In the code block above, since we commented the return statement, the value of variable sum is given as undefined in the output

**Null:-**

*`null`* is a reserved keyword in javascript. We can assign a null value to a variable explicitly using the keyword `null`. Null essentially represents a non-existent or an empty value i.e. we explicitly tell javascript interpreter that the variable has no value.

```
let life = null;
console.log(life); //null
```

### Not defined:-

A not defined is a variable which is not declared at a given point of time with declaration keyword like var, let or const.

```
console.log(a);
var a = 5;
//Output:- undefined
```

While if we don’t use var keyword above the output will be:-

```
console.log(b);
b = 5;
//Output:- "ReferenceError: b is not defined
```

The reason why the variable `a` was printed with `undefined` value above but `b` was declared as not defined was because of the way variable hoisting works in javascript i.e. the variable declarations are processed before code execution takes place in javascript.

Internally the execution for the variable `a` above was as follows:-

```
var a;
console.log(a); //undefined
a = 5;
```

While for variable `b` above it was as follows:-

```
console.log(b); // b is not defined
window.b = 5;
```

Differences:-

The type of null and undefined are different as follows:

```
console.log(typeof(undefined));   //"undefined"
console.log(typeof(null));        //"object"
```

*null* is an object with a valid non-existent value and it is non-mutable while object type of an *undefined* value is in itself undefined.

Also any arithmetic operation with n*ull* value will result in integer value while any arithmetic operation with *undefined* with result in value of variable being changed to NaN.

```
let a = 7 + null;
console.log(a); // 7

let b = 7 * null;
console.log(b); // 0

let c = 8 + undefined;
console.log(c); //NaN

let d = 8 * undefined;
console.log(d); //NaN
```

Hope your concepts of the three got a bit more clear today. Stay tuned for more such posts in near future.

JavaScript Programming Tech Technology Software Development