# Null & Undefined

Learn two more variable types in JavaScript: null and undefined. Learn what they represent, how they differ, and how to use them.

We'll cover the following


- undefined
- null
  - undefined vs null
- Quiz


There are two more variable types that we should discuss.

## `undefined` #

`undefined` is meant to represent the idea that something doesn't exist. When we try to use a variable that has no value, we get `undefined`. Here's an example.

```
let variable;
console.log(variable); // -> undefined
```

When we declare a variable using `let` but don't give it a value, it receives the default value of `undefined`. This is JavaScript telling us that we're trying to use something that isn't there.

**`undefined` is a special variable type used to indicate that something is missing.** If we see it, it usually means something is wrong with our code.

For example, we'd have no reason to use a variable if we haven't given it a value, so when we do try to use it, JavaScript gives us `undefined`.

We can use it ourselves and assign the value to variables directly.

```
let variable = 'abc';
console.log(variable); // -> abc

variable = undefined;
console.log(variable); // -> undefined
```

Although we can do this, we pretty much never want to assign a variable a value of `undefined`. It's meant to show that something has gone wrong and some value is missing.

If we want to specify that something has no value, we should use `null`.

## `null`#

`null` is another variable type. It represents something that's empty. The difference between `null` and `undefined` lies in they're implemented by JavaScript and used by developers.

`null` is something that is safe to use and to assign to variables.

```
let variable = null;
console.log(variable); // -> null
```

## `undefined` vs `null`#

As mentioned, the JavaScript engine will make variables that have no value `undefined`. If we want to specify that a variable should be empty, the "correct" way to do so is to make it equal to `null`.

Code will work whether we use `undefined` or `null`. However, the development community often has a set of standard "best practices" to make code easier to read, interpret, and debug.

Preferring `null` over `undefined` is one of those best practices. Setting a variable to `null` is clear and communicates that we want to essentially delete the variable. We're done with it.

Setting it to `undefined` will make people look twice. They'll see it, be confused, and might look down upon the failure to follow common practice.

At this point, just know that these are two more variable types in JavaScript. We won't be using them too much.

# Quiz#

Feel free to test your understanding.

1 What will this code print?

```
let variable;
let variableCopy = variable;
variableCopy = 'def';
console.log(variable);
```

A) def

B) null

C) undefined

D) error

Submit Answer

Question 1 of 3

0 attempted

Reset Quiz

Back

Working with variables

Next

Operators: +, -, *, /, %

Completed

Report an Issue