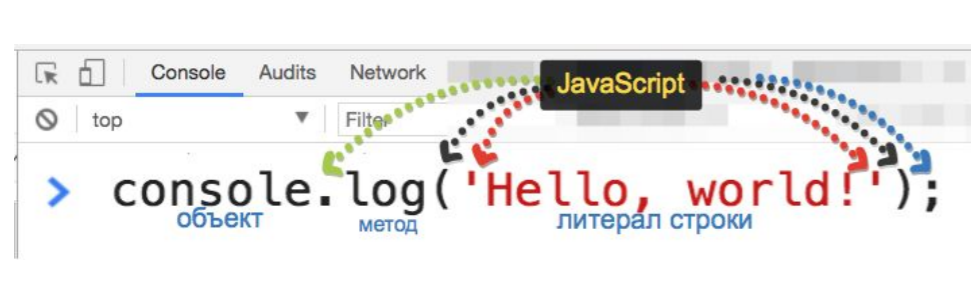
Author: Guzel Garifullina



**Every time console.log is executed, a line saying undefinedis appended to the output log.**

If you're running console.log() from a JS file, this undefined line should not be appended.

If you're running console.log() from the console itself, it makes sense. This is why: In the console you can type a name of a variable (for example try typing window) and it prints info about it. When you run any void function (like console.log) from the console, it also prints out info about the return value, undefined in this case.

I tested both cases on my Chrome (Mac ver 23.0.1271.101) and indeed I see the undefined line when I run it inside the console. This undefined also appears when I write this line in the console: var bla = "sdfdfs"

undefined - The global **undefined** property represents the primitive value [undefined](https://developer.mozilla.org/en-US/docs/Glossary/Undefined). It is one of JavaScript's [primitive types](https://developer.mozilla.org/en-US/docs/Glossary/Primitive)

A variable that has not been assigned a value is of type undefined

**not reserved word**

// foo string

(function() { var undefined = 'foo'; console.log(undefined, typeof undefined); })();

null == undefined // true

null === undefined //false

One reason to use [typeof](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/typeof) is that it does not throw an error if the variable has not been declared.

// x has not been declared before

if (typeof x === 'undefined') { // evaluates to true without errors

// these statements execute

}

if (x === undefined) { // throws a ReferenceError

}

#### **Null**

Null means an empty or non-existent value. Null is assigned, and explicitly means nothing.

null is also an object.

node -e = eval

node -p = eval + print (value + undefined)

Identical to -e but prints the result.

**Not defined and undefined**

not defined variable - not declared

undefined - not difined (without assigned value)