nerdjfpb

nerdjfpb

What is Javascript?

JS

nerd_jfpb

nerdjfpb

JavaScript, often abbreviated as JS, is a high-level, just-in-time compiled, multi-paradigm programming language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. (Wiki)

nerdjfpb

So mainly javascript is a programming language which helps us to make many things!
But mostly we knew it for web development.

Ever select anything from dropdown menu ? That one made by javascript. Javascript changed the whole things of web interactions.

You can use javascript in everywhere now. Like - Web, mobile app, desktop app, robotics, machine learning etc sector.

nerdjfpb

JavaScript has the most job in the world, so you can learn it for a better career!.

nerdjfpb

Some learning resources -

JavaScript

On this Page

Tutorials
Reference
Tools & resources

Related Topics

JavaScript

Tutorials:

- Complete beginners
- JavaScript Guide
- Intermediate
- Advanced

JavaScript (**JS**) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. Read more about JavaScript.

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about APIs specific to Web pages, please see Web APIs and DOM.

The standard for JavaScript is ECMAScript. As of 2012, all modern browsers fully support ECMAScript 5.1. Older browsers support at least ECMAScript 3. On June 17, 2015, ECMA International published the sixth major version of ECMAScript, which is officially called ECMAScript 2015, and was initially referred to as ECMAScript 6 or ES6. Since then, ECMAScript standards are on yearly release cycles. This documentation refers to the latest draft version, which is currently ECMAScript 2020.

Do not confuse JavaScript with the Java programming language. Both "Java" and "JavaScript"

w3schools.com

THE WORLD'S LARGEST WEB DEVELOPER SITE

HTML CSS JAVASCRIPT SQL PYTHON PHP BOOTSTRAP HOW TO W3.CSS JQUERY MORE REFERENCES EXERCISES

JS Tutorial
JS HOME
JS Introduction
JS Where To
JS Output
JS Statements
JS Syntax
JS Comments
JS Variables
JS Operators
JS Arithmetic
JS Assignment
JS Data Types
JS Functions
JS Objects
JS Events
JS Strings
JS String Methods
JS Numbers
JS Number Methods
JS Arrays
JS Array Methods
JS Array Sort

JavaScript Tutorial

‹ Home

Next ›

JavaScript is the programming language of HTML and the Web.

JavaScript is easy to learn.

This tutorial will teach you JavaScript from basic to advanced.

Examples in Each Chapter

With our "Try it Yourself" editor, you can edit the source code and view the result.

Example

Introduction to JavaScript

JavaScript is a high-level programming language that all modern web browsers support. It is also one of the core technologies of the web, along with HTML and CSS that you may have learned previously. This section will cover basic JavaScript programming concepts, which range from variables and arithmetic to objects and loops.

Go to the first lesson

View the curriculum

Upcoming Lessons

Comment Your JavaScript Code

Declare JavaScript Variables

Storing Values with the Assignment Operator

Initializing Variables with the Assignment Operator

Understanding Uninitialized Variables

nerdjfpb

nerdjfpb

Javascript Types

JS

nerd_jfpb

nerdjfpb

nerdjfpb

We'll writing js code on the console of google chrome

nerd_jfpb

nerdjfpb

Javascript have 7 types.
You can think types as values.
They are -

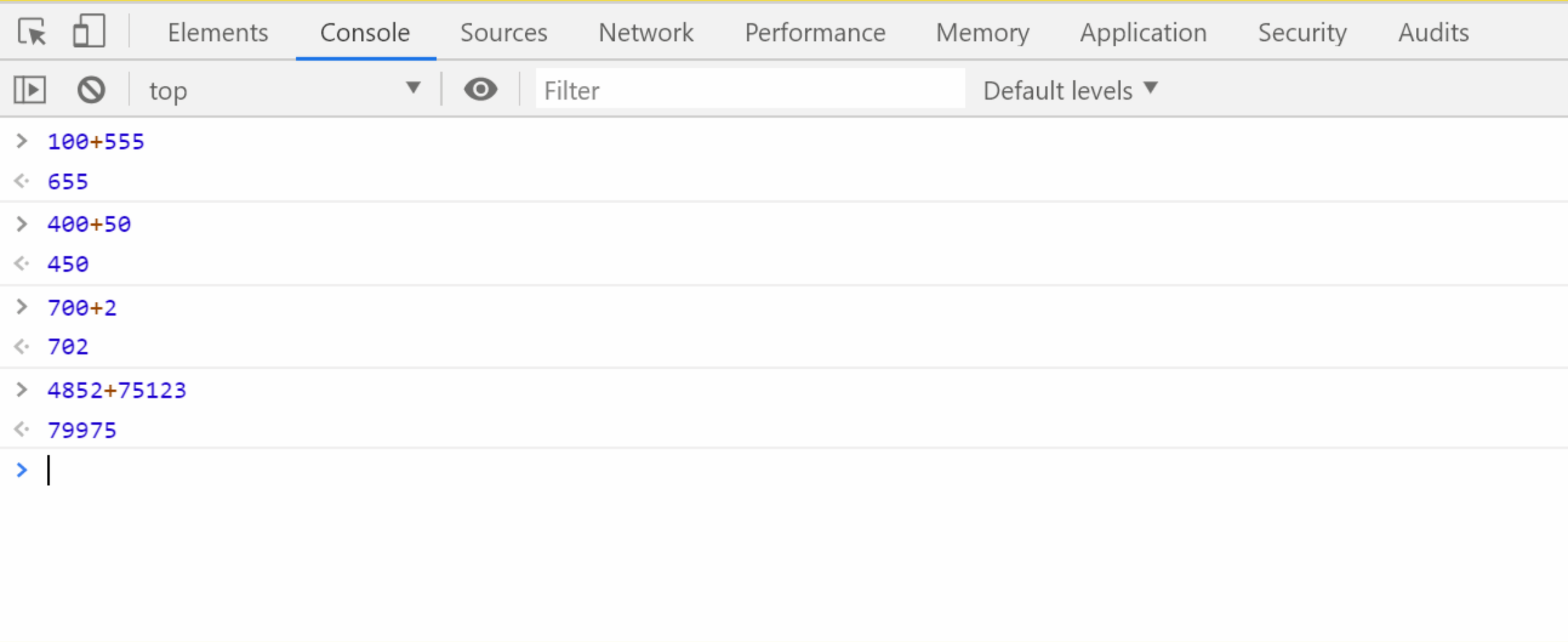
a. Number
b. String
c. Boolean
d. Undefined
e. Null
f. Symbol (new in es6)
g. Object

First we'll talk about Number.

nerdjfpb

nerdjfpb

We can easily add number in javascript. Just adding the '+' operator in middle.

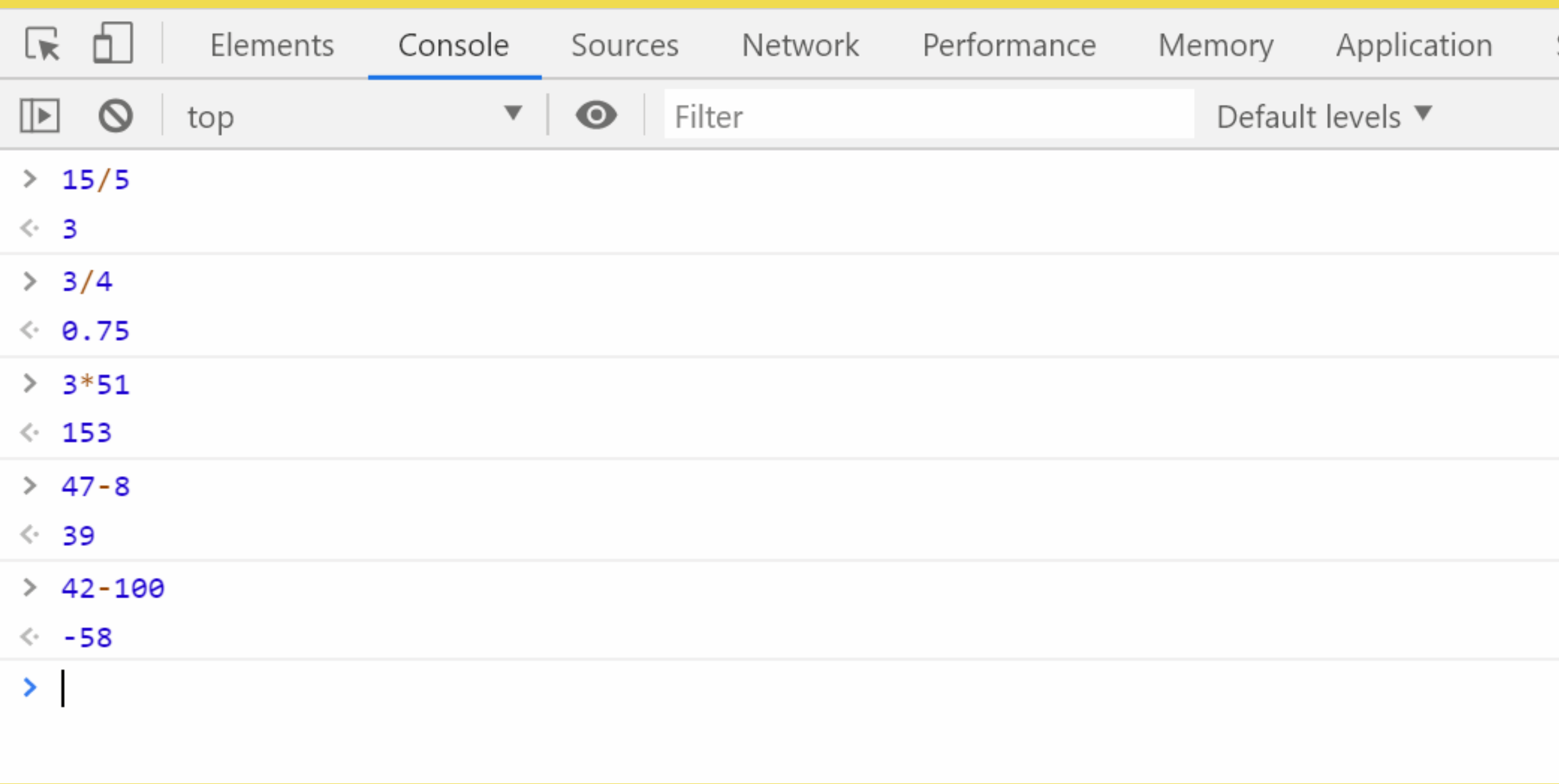


nerd_jfpb

nerdjfpb

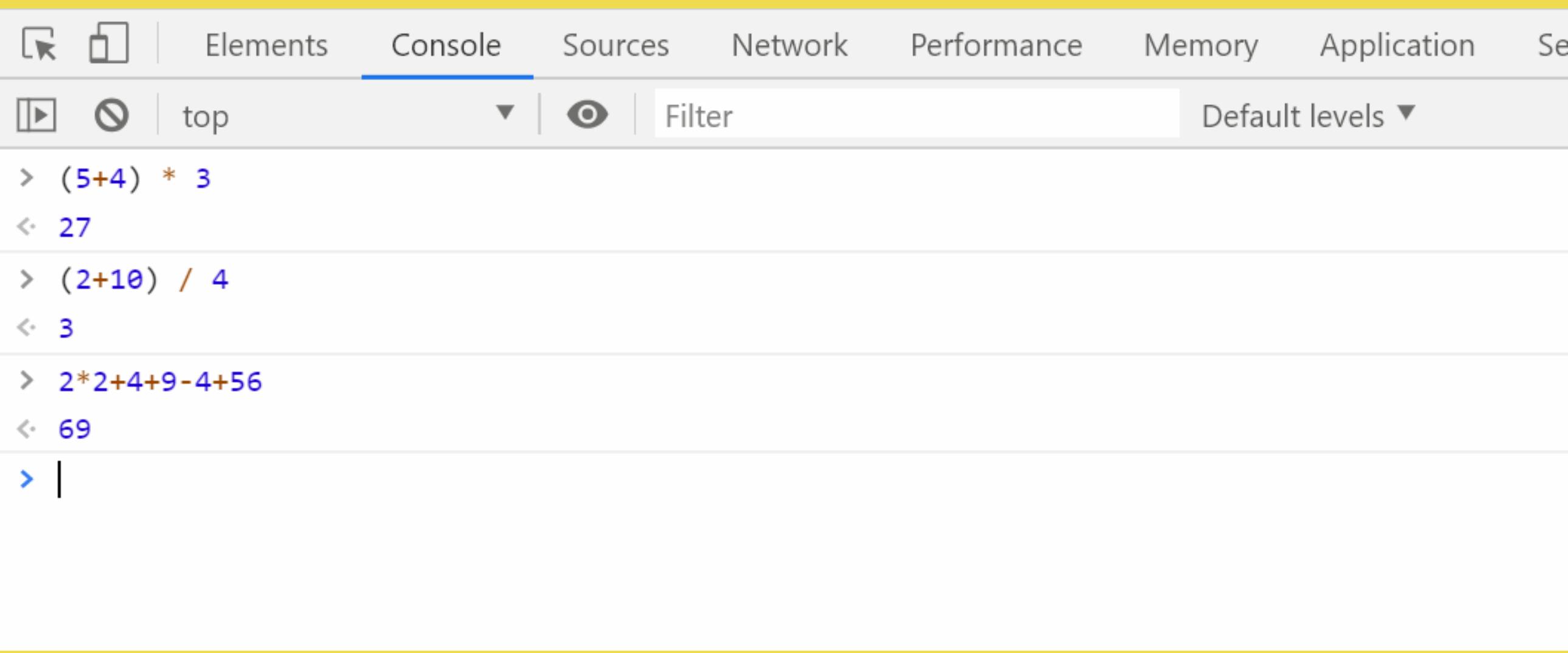
nerdjfpb

We can do all the arithmetic things here using javascript.



nerd_jfpb

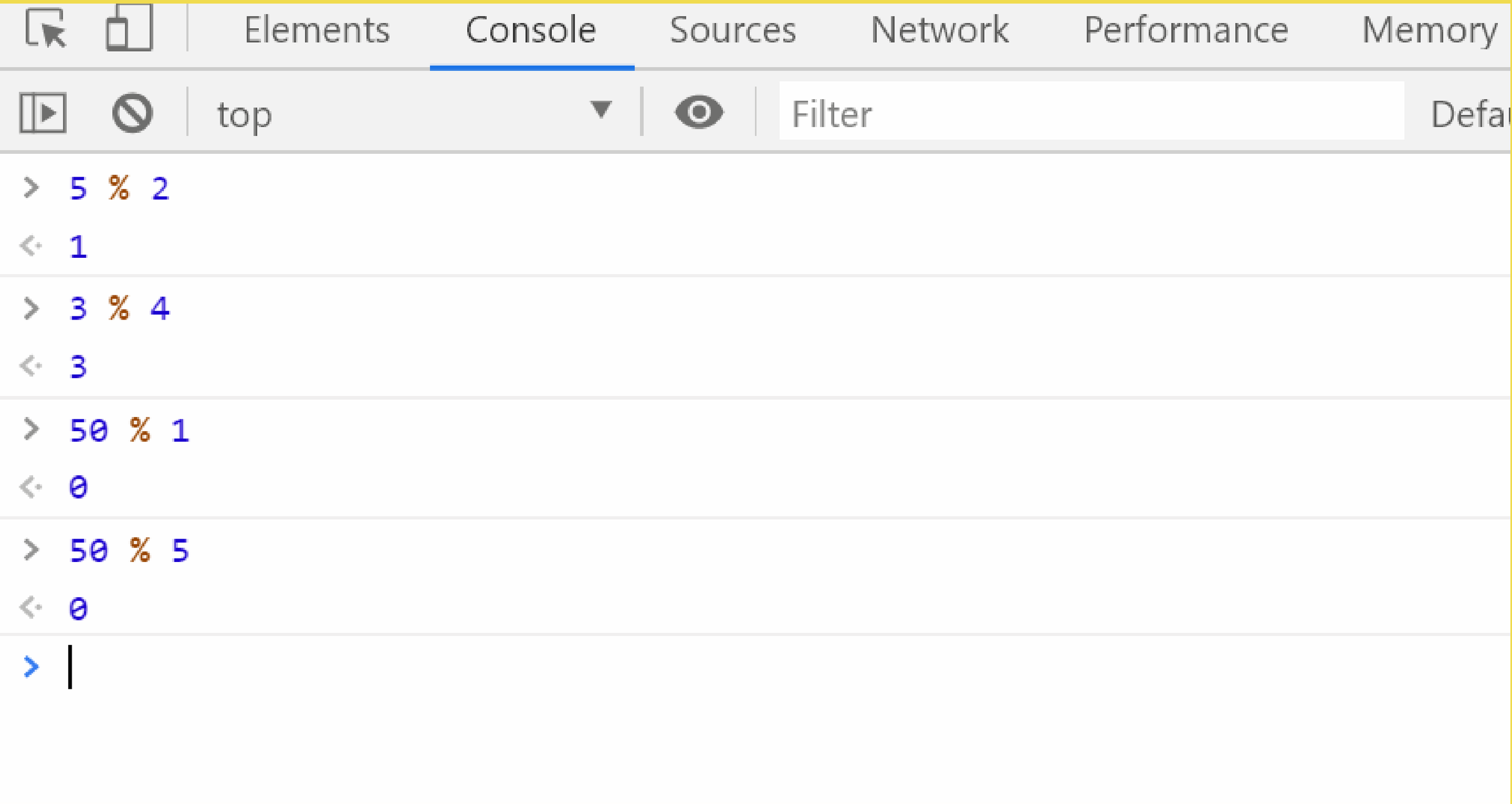
Working on some more arithmetic logics.



nerdjfpb

nerdjfpb

What do you think about this '%' ?



nerd_jfpb

nerdjfpb

This is actually 'modulo' this gives us the reminder!

nerdjfpb

nerdjfpb

Javascript Types Cont.

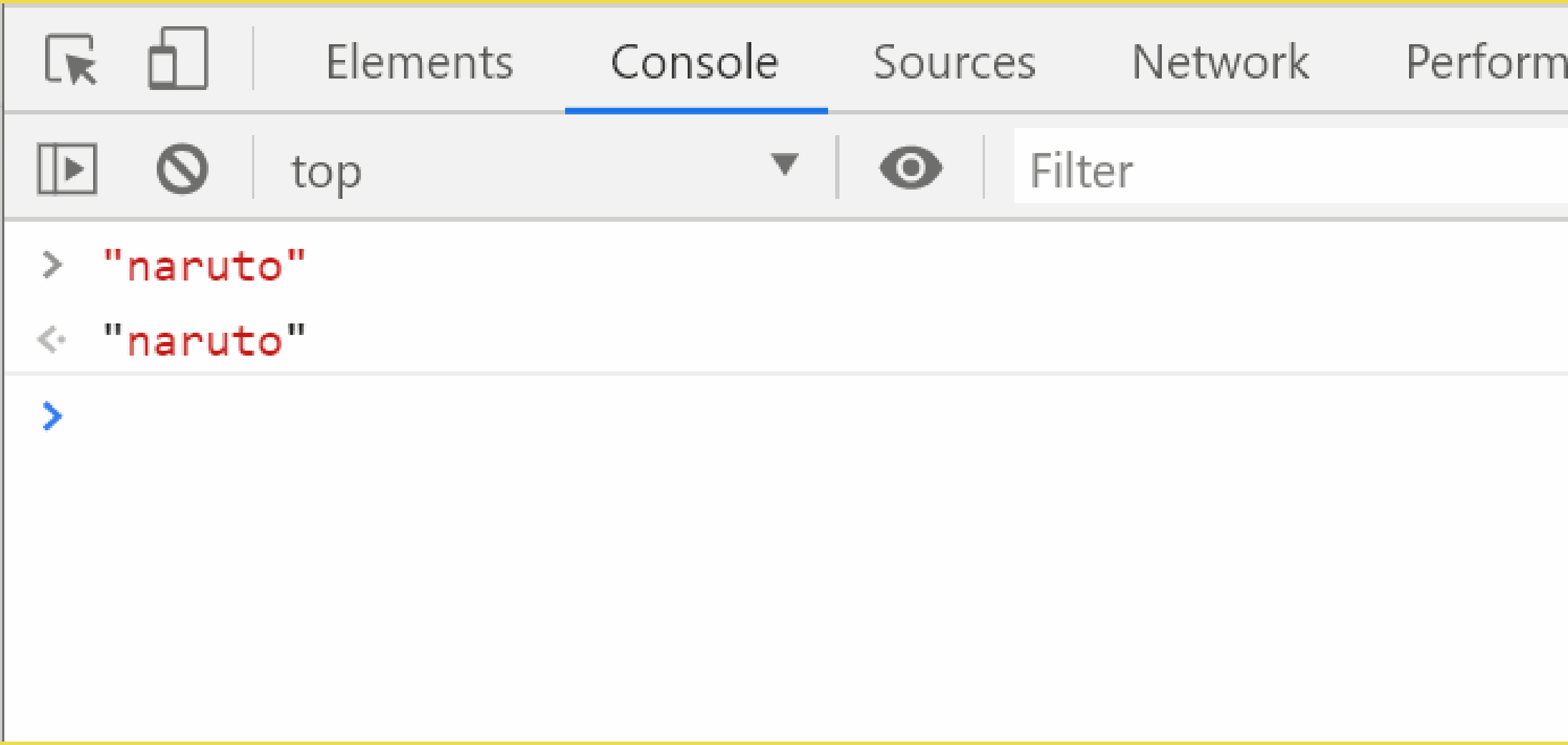
JS

nerd_jfpb

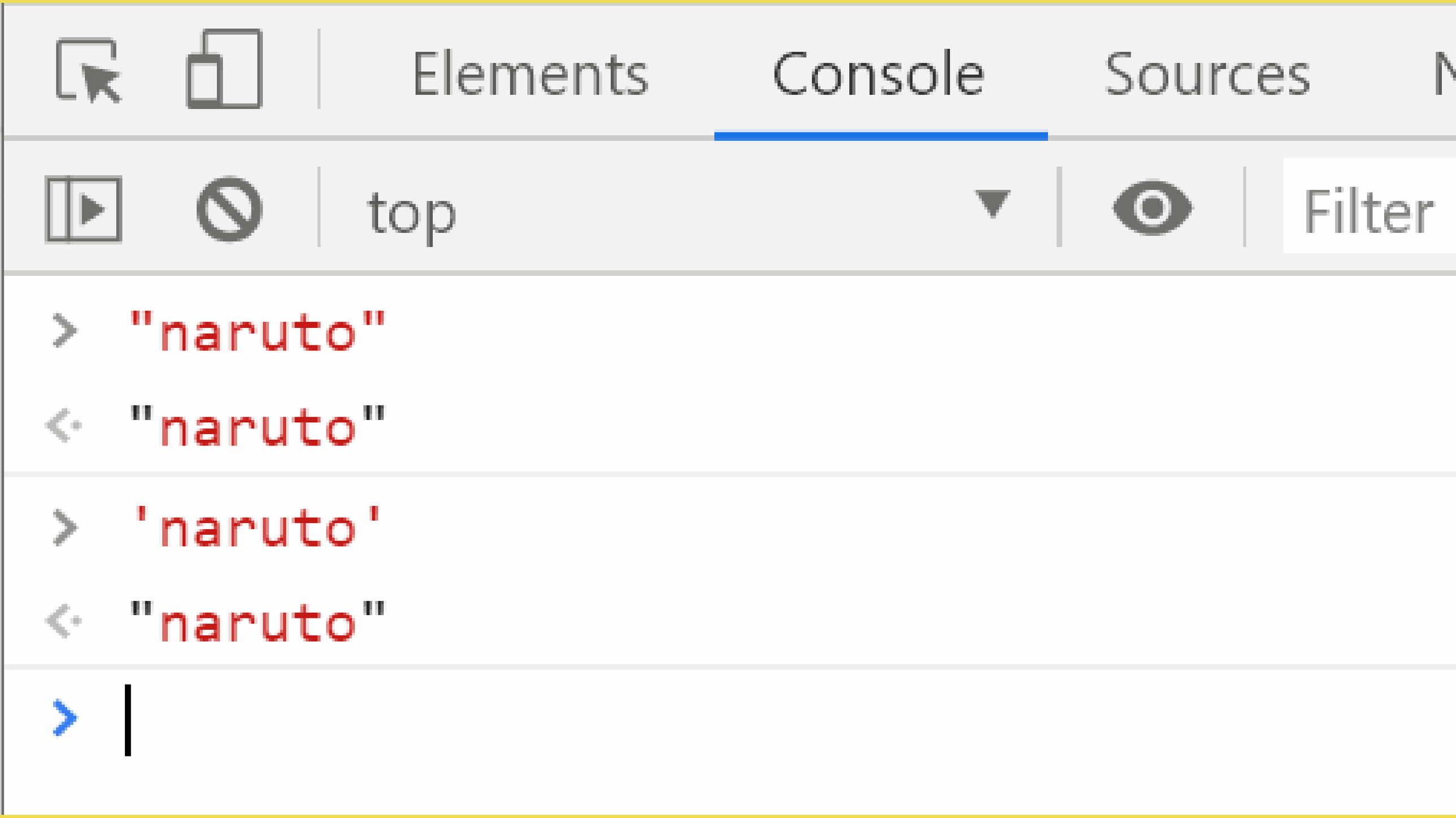
nerdjfpb

Today we're going to learn about String.

String is simple text type. We need a " for working with the string.



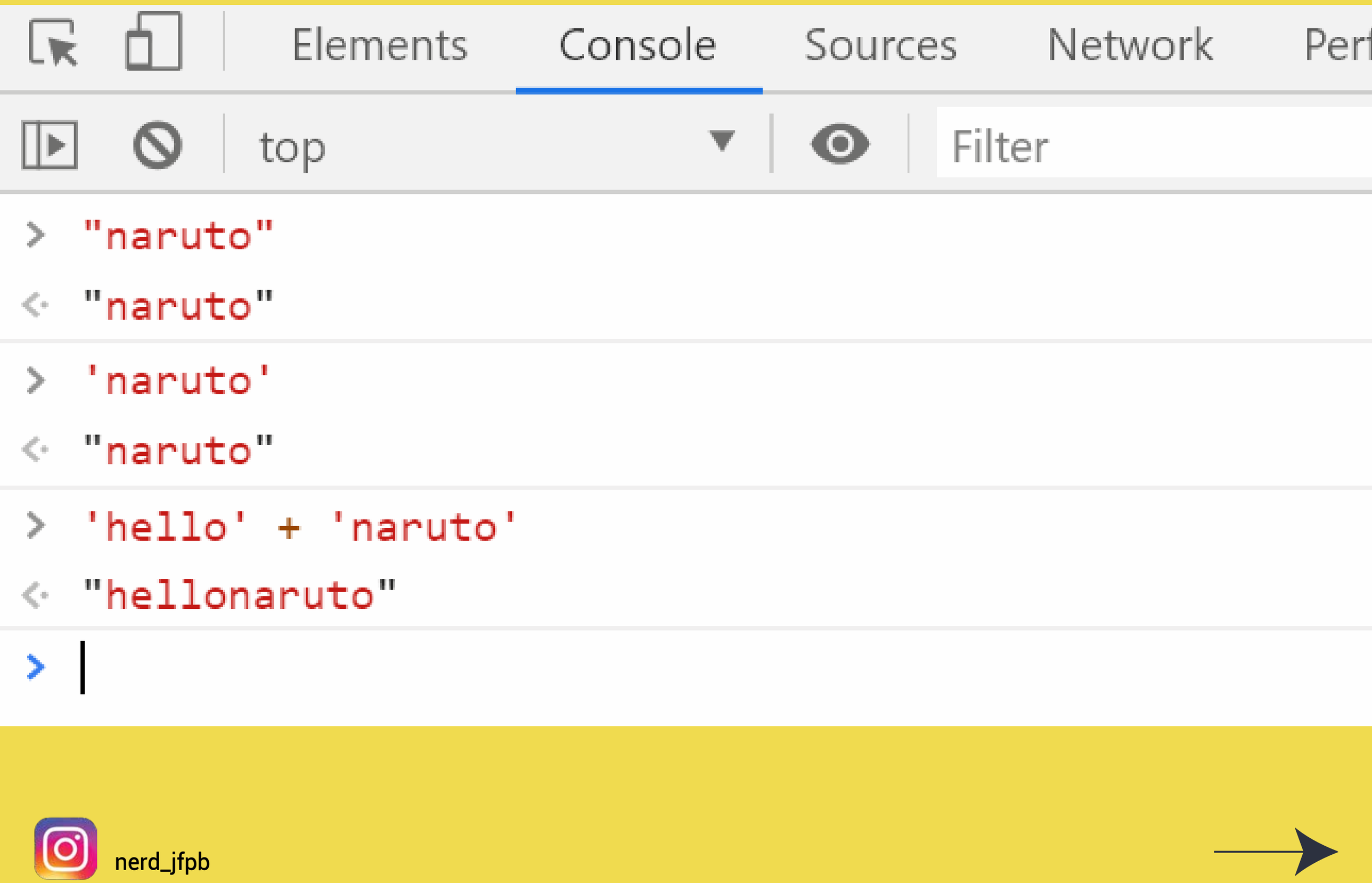
We can use single ' too.



nerdjfpb

nerdjfpb

We can add strings using + between two strings.



nerd_jfpb

nerdjfpb

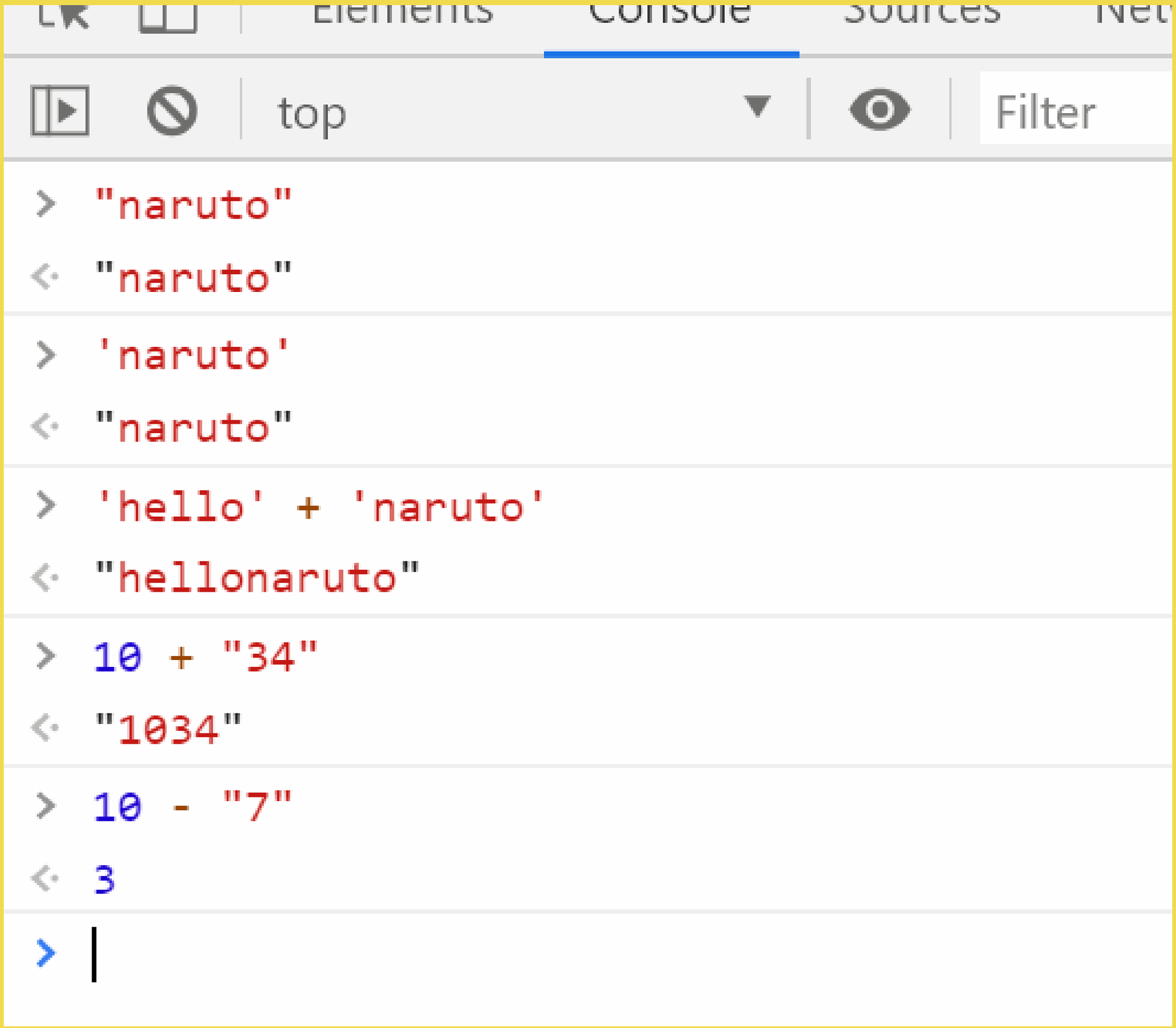
nerdjfpb

When we add number + string. Javascript made it to string.

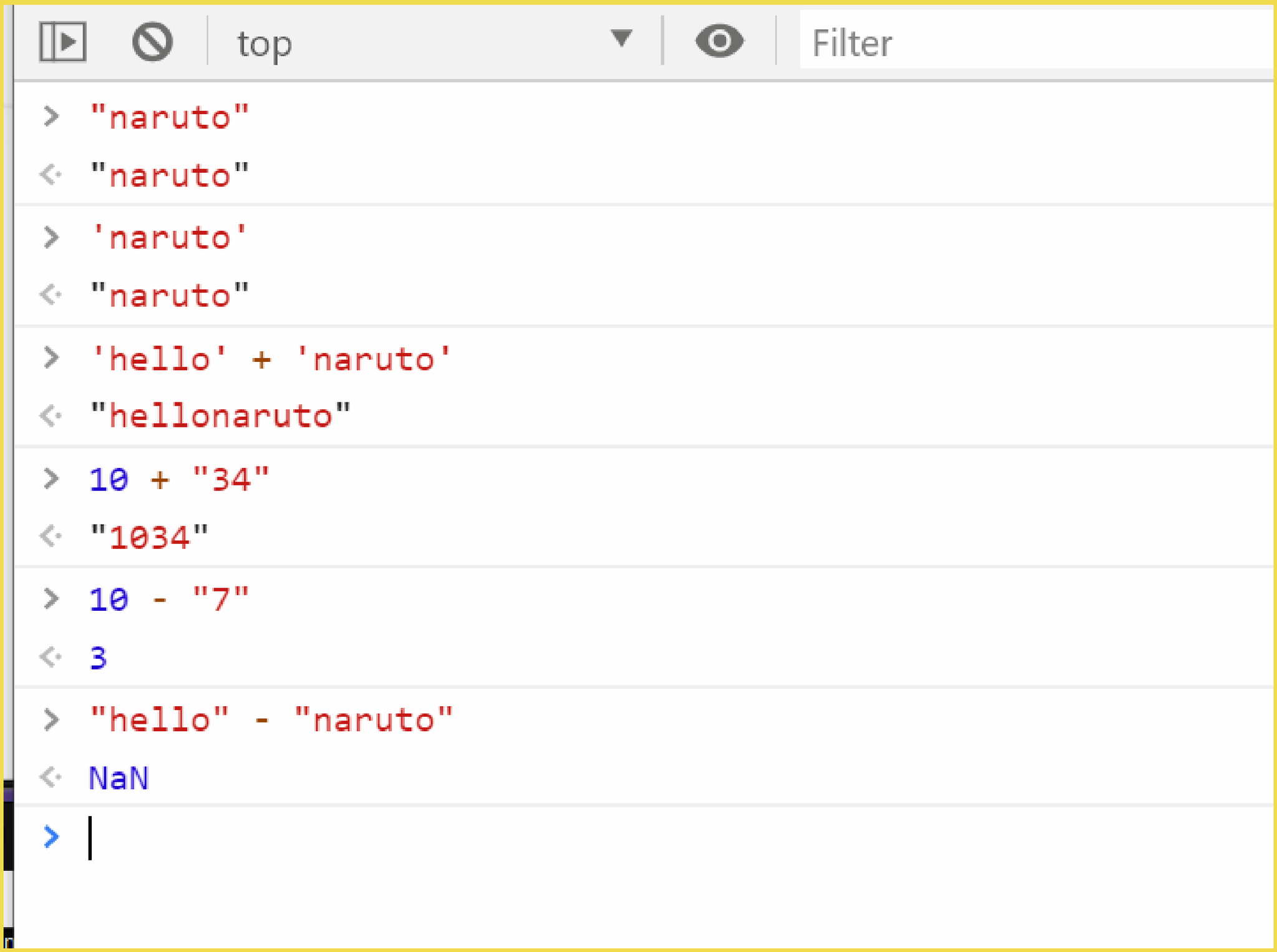
```
> "naruto"
<- "naruto"
> 'naruto'
<- "naruto"
> 'hello' + 'naruto'
<- "hellonaruto"
> 10 + "34"
<- "1034"
>
```

nerd_jfpb

But when we delete string from a number. It assume that string is a number.



When we try to delete string from string it give us NaN.



nerdjfpb

We normally don't do this, When we do any arithmetic operation, we don't contain any string in there. So that we're not facing any issue. .

nerdjfpb

nerdjfpb

Do you learned something about string today?

nerd_jfpb

nerdjfpb

nerdjfpb

Javascript Types Cont.

JS

nerd_jfpb

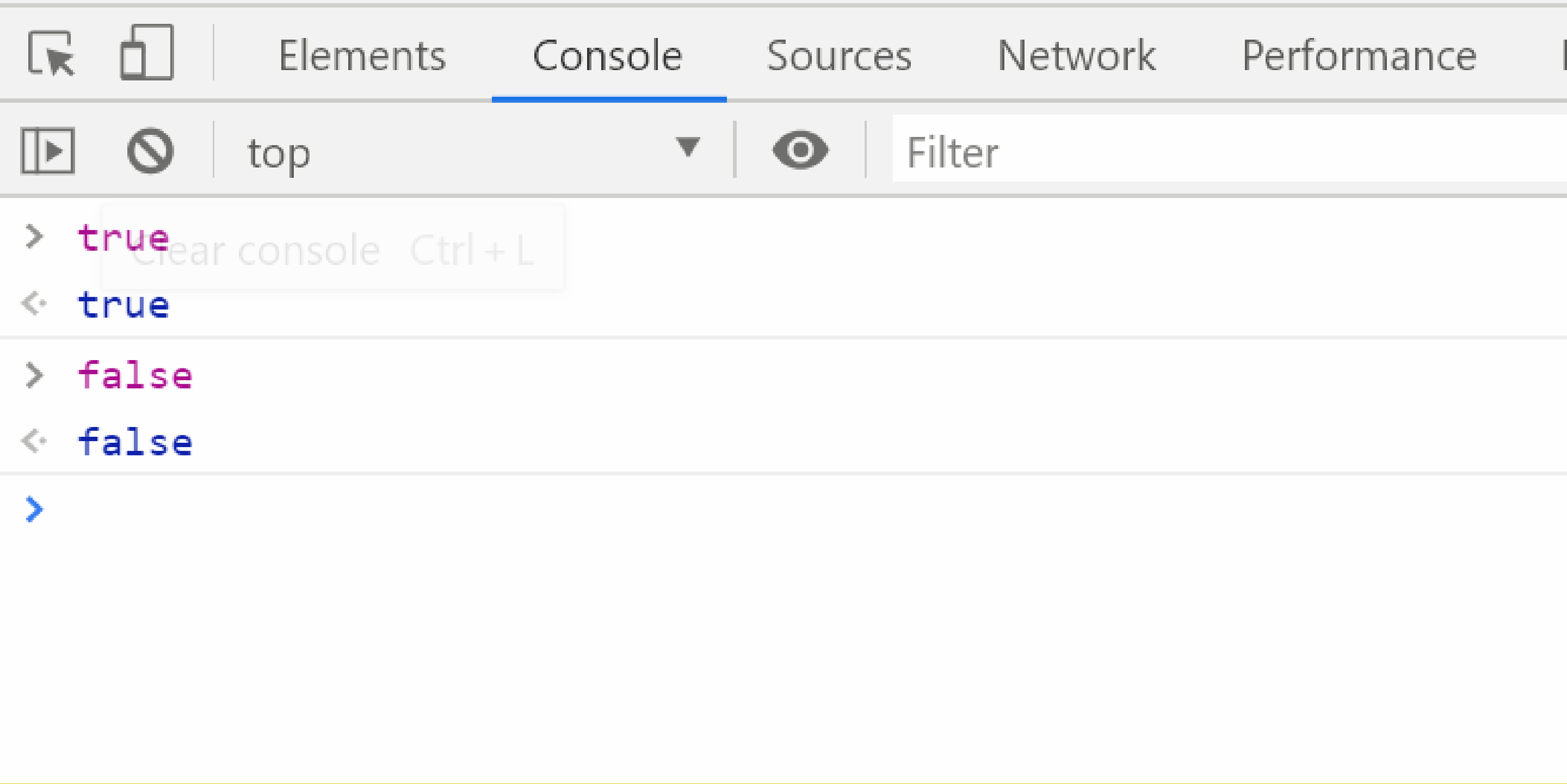
nerdjfpb

Today we're going to learn about Boolean.

nerdjfpb

Boolean is a pretty easy stuff.
Boolean is just about two states.
True & False.

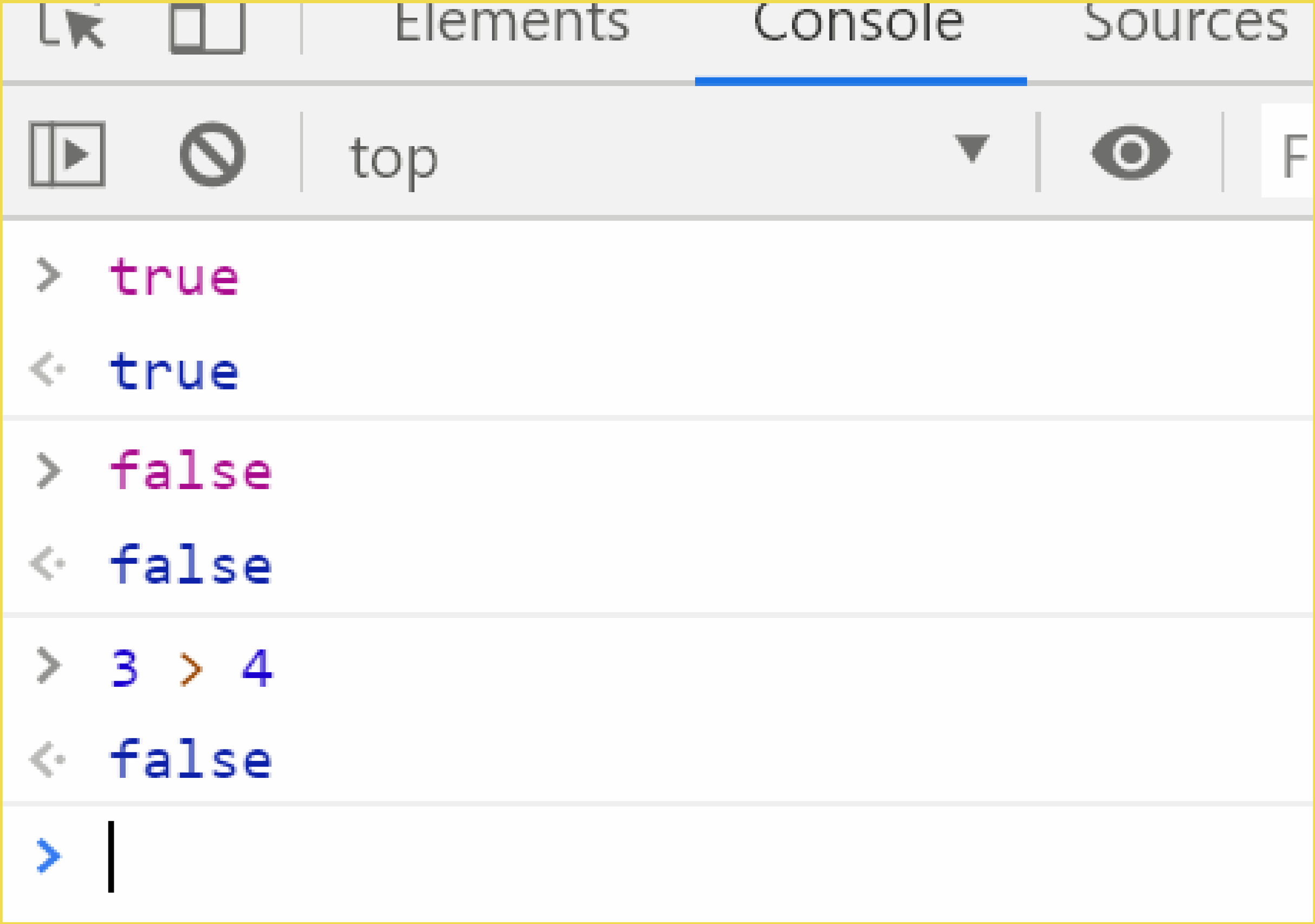
See an example -



nerdjfpb

Boolean is 1 and 0 actually!

Boolean is really useful in a code. Example -



nerdjfpb

In the 3 > 4 we get the false.
So this is how we can assume some logic is right or not.

Boolean data format is almost available in every programming language.

nerdjfpb

Do you want to use the 'true' and 'false' which is boolean or want to use 1 and 0 which also gave the same thing.

nerdjfpb

nerdjfpb

Let me know if you have any questions.

nerd_jfpb

nerdjfpb

nerdjfpb

Javascript Comparisons

JS

nerd_jfpb

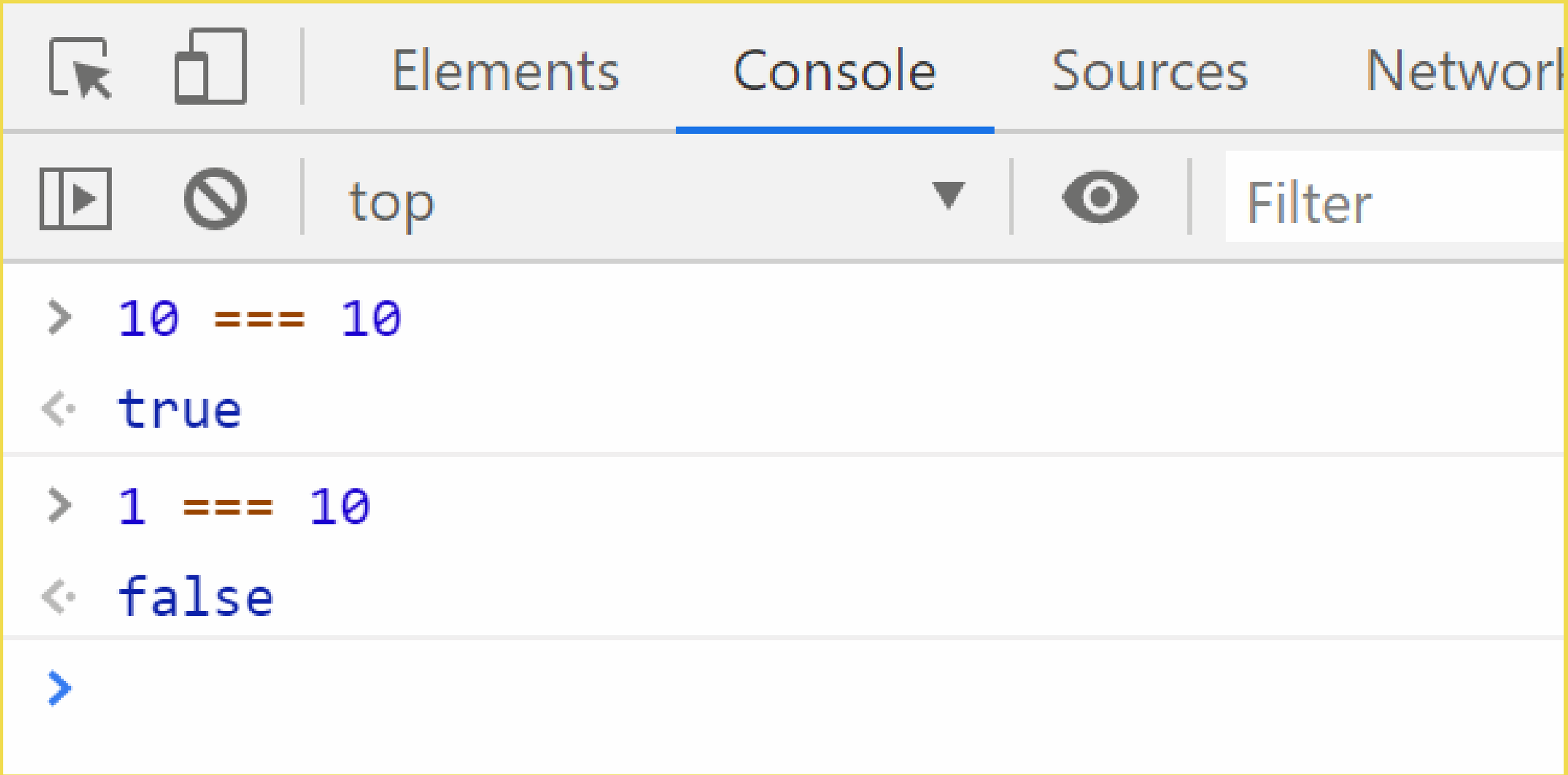
There are some JavaScript types that we'll discuss right now. We'll work on that later.

They are - Undefined, Null, Symbol, Object.

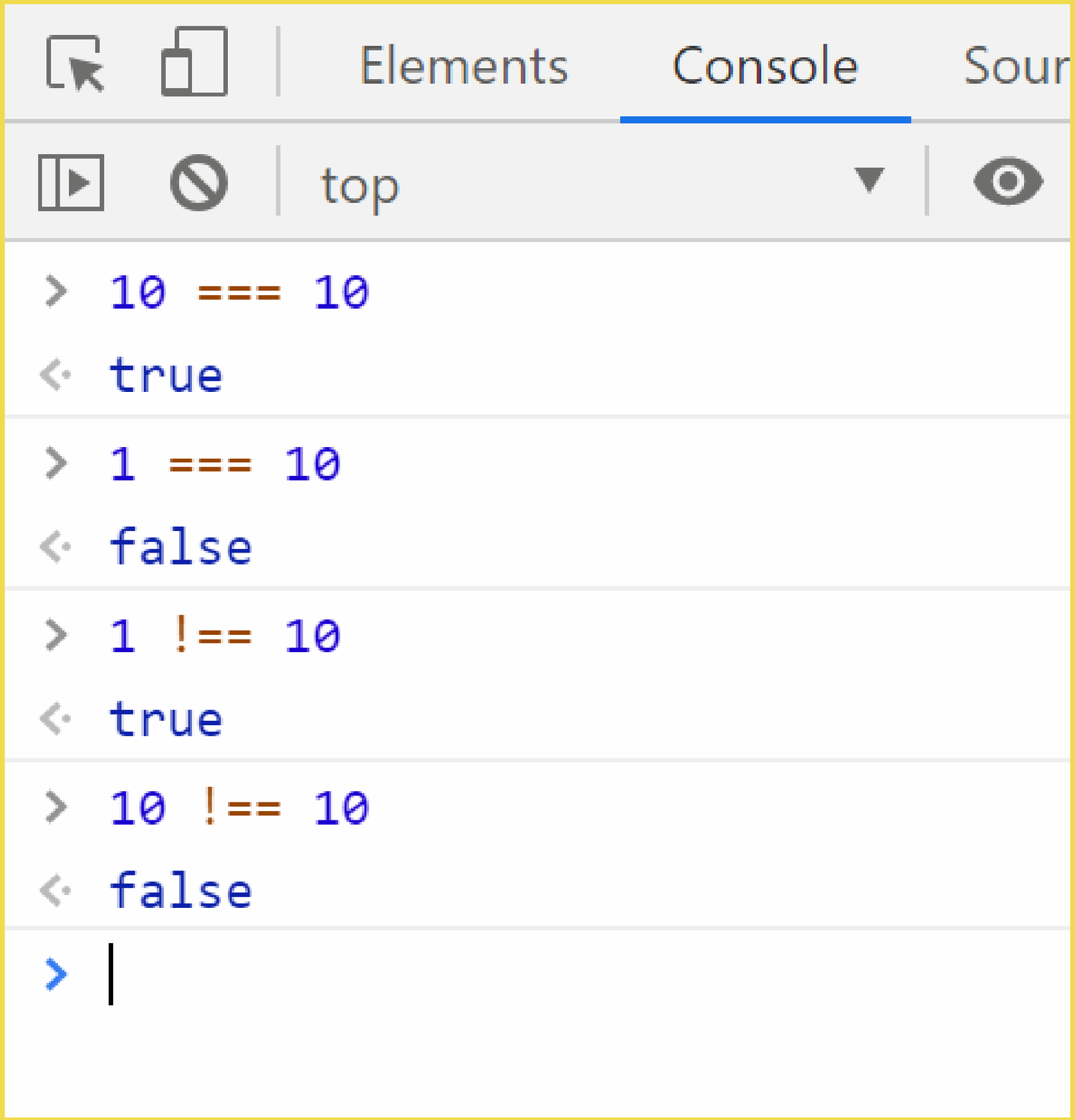
nerdjfpb

Today we are going to learn comparisons in JavaScript.

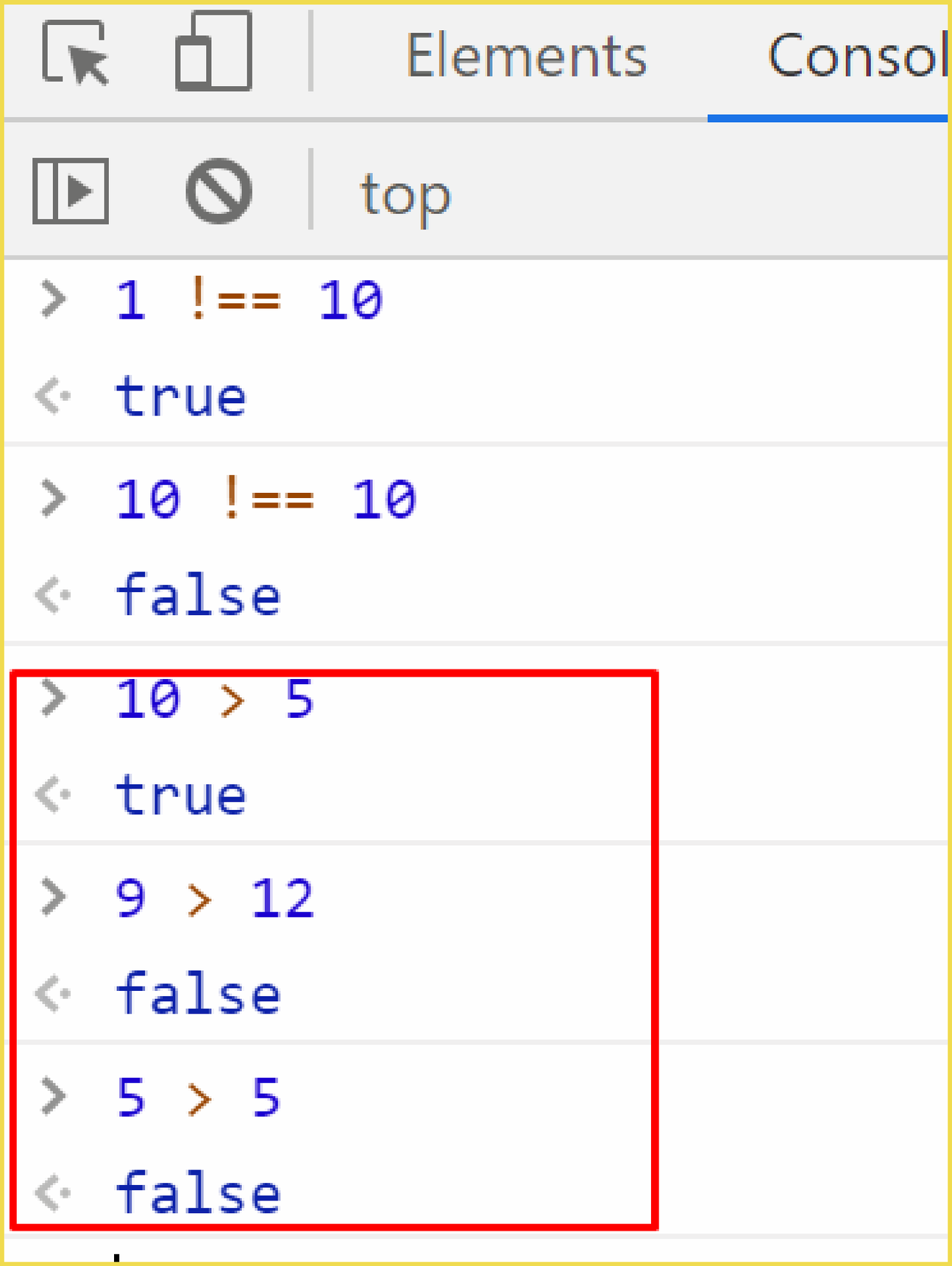
Lets start with checking equal. In javascript we need to use === for checking if they are same or not.



To find out if they are not same we use !==
This gives us the boolean result

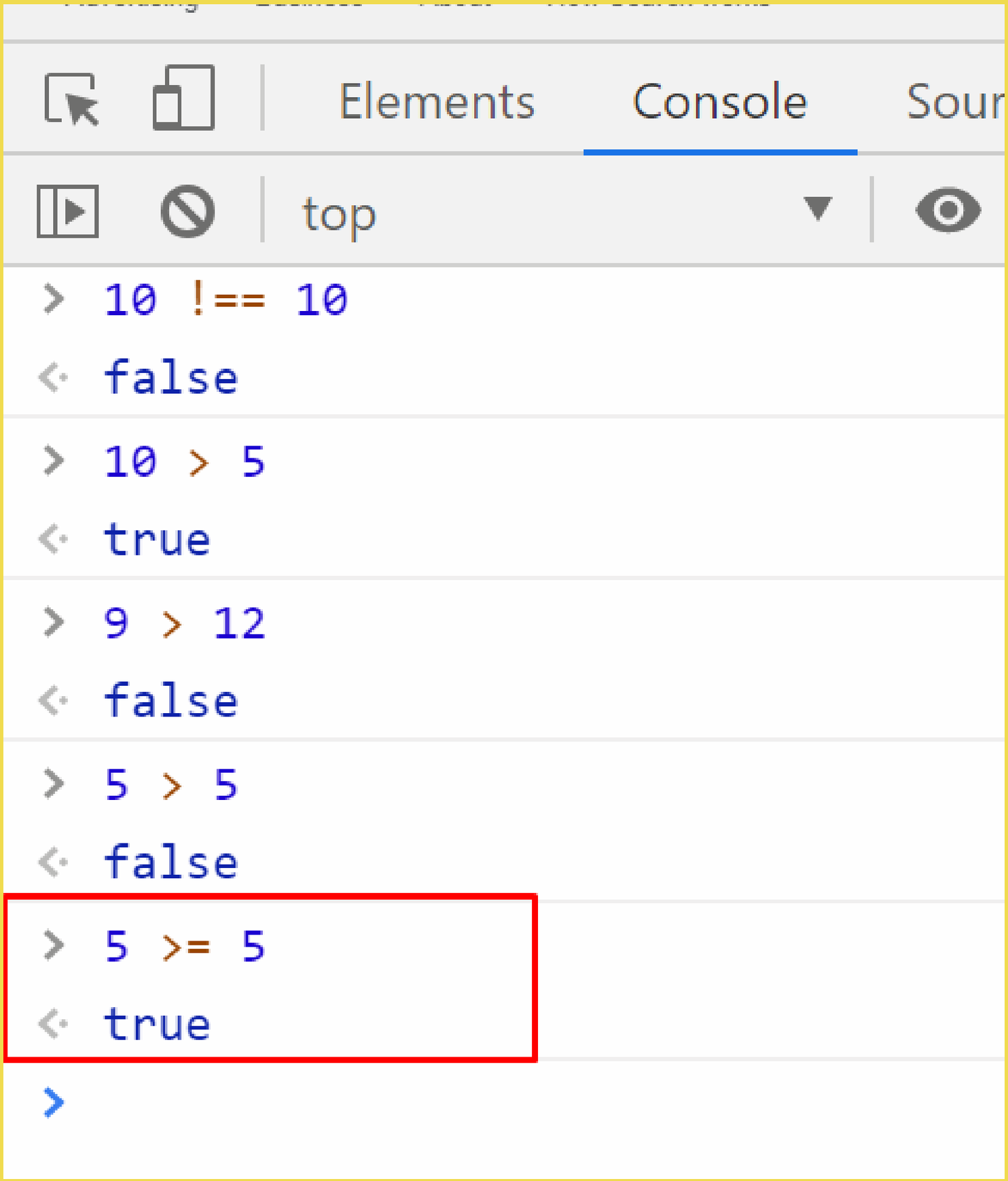


To find some value is greater than some value we use >



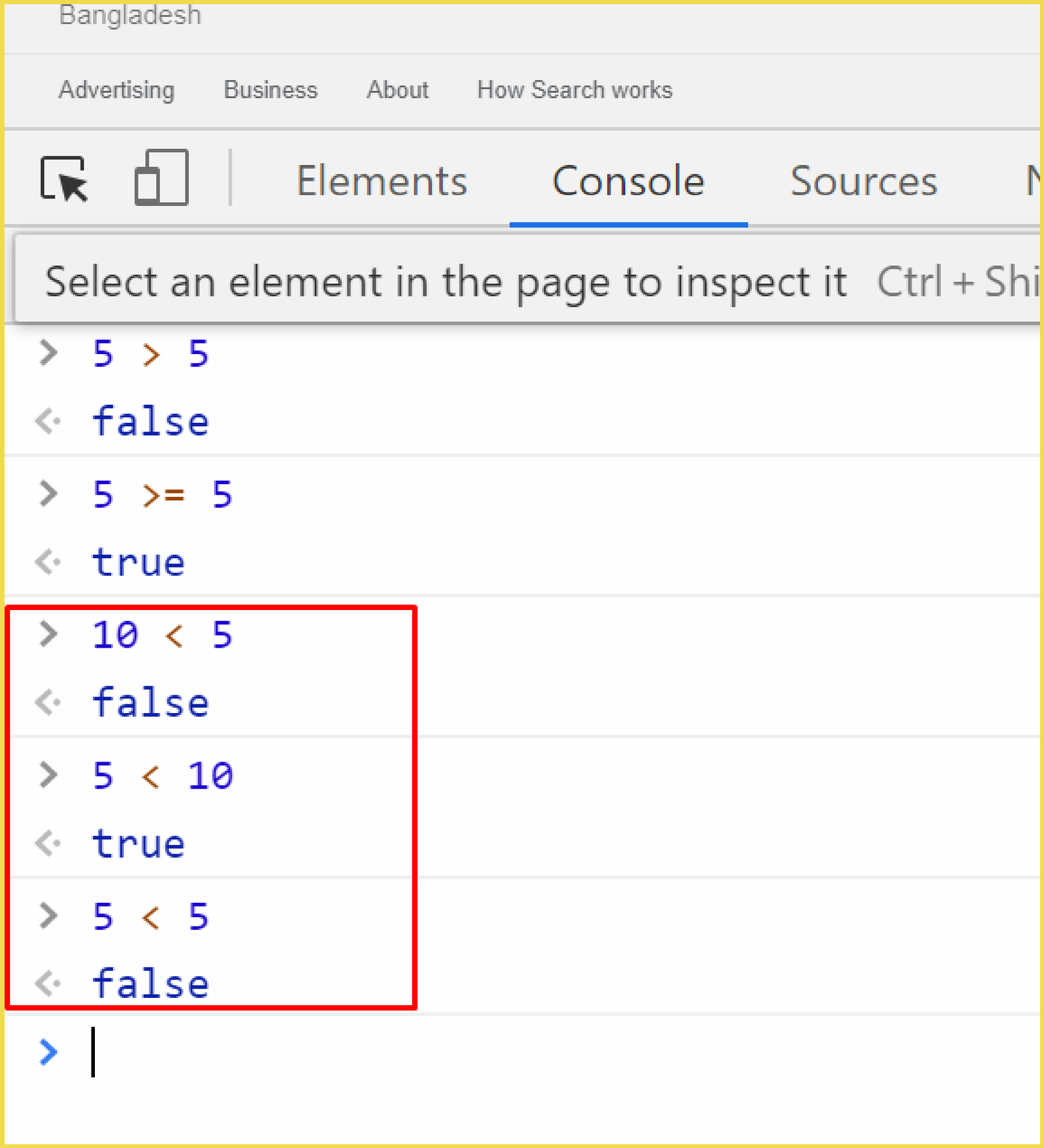
nerdjfpb

To find some value is greater than and equal to some value we use >=



nerdjfpb

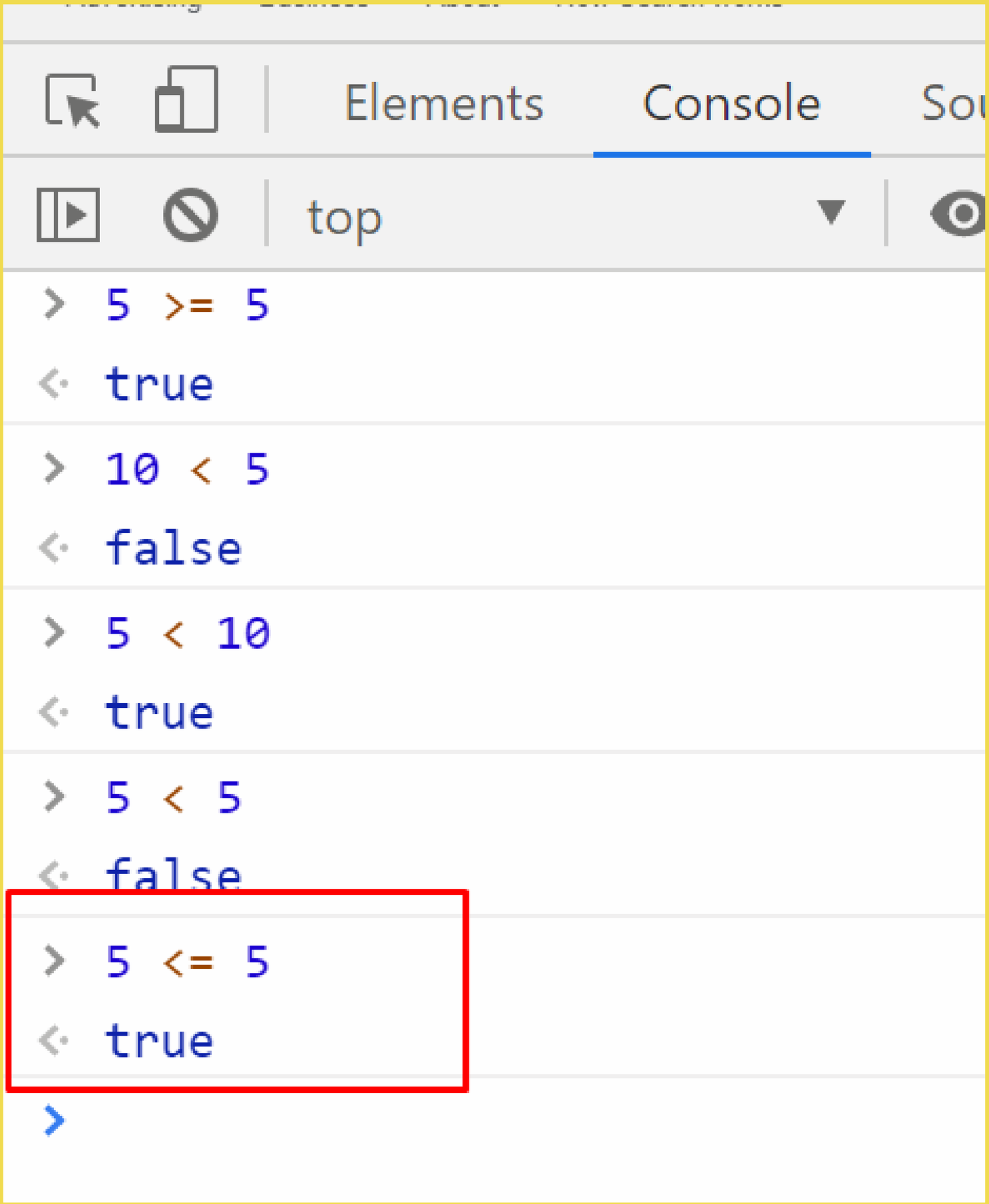
To find some value is smaller than some value we use <



nerd_jfpb

nerdjfpb

To find some value is smaller than and equal to some value we use >=



Do you understand the comparisons signs now in JavaScript ?

nerdjfpb

nerdjfpb

Javascript Variables

JS

nerd_jfpb

nerdjfpb

**Up until now,
we are not making to remember something. Everything wasn't stored.
But from today we'll store some value in some variables.
It's like x = 5 in math!**

nerdjfpb

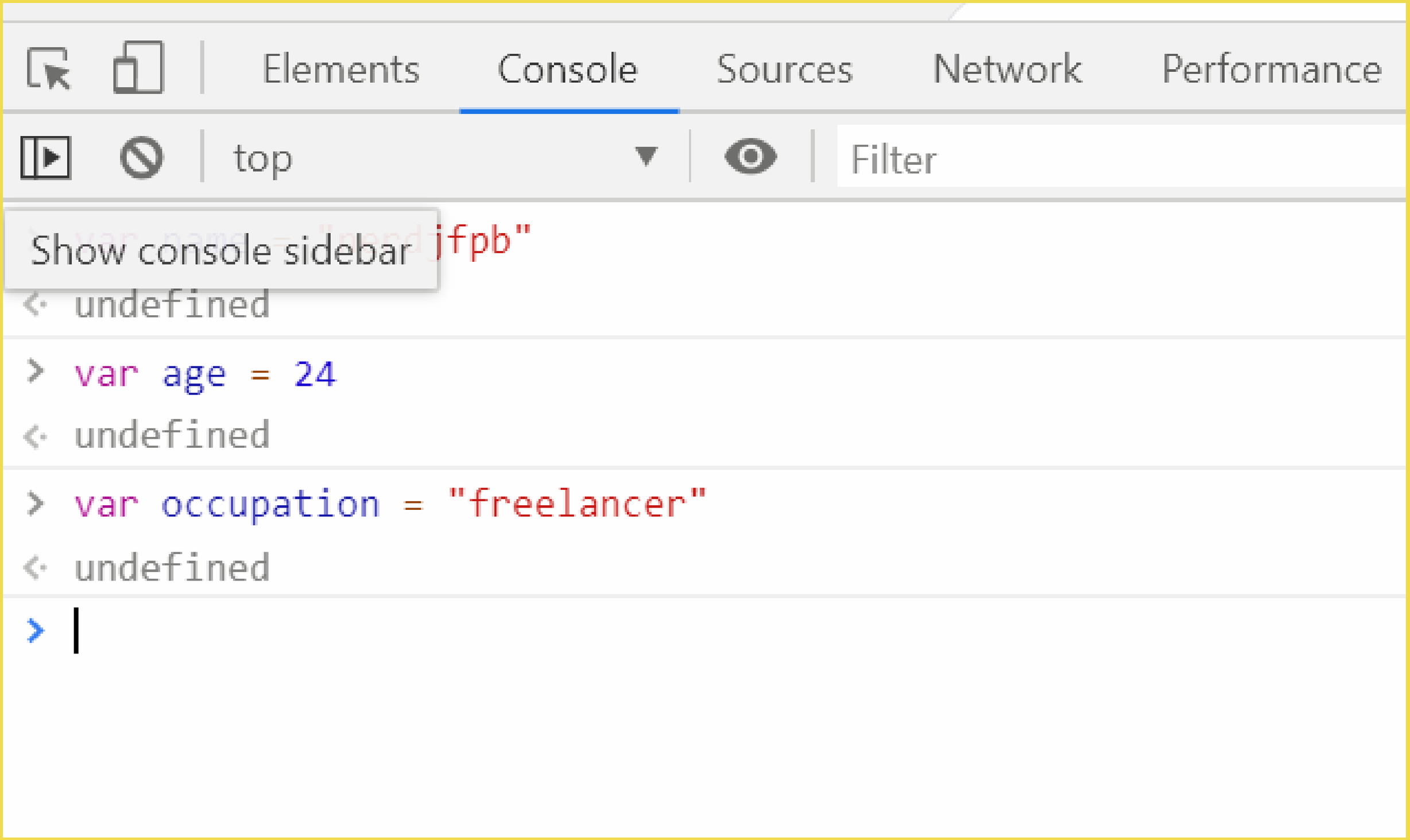
We can do different things with the variables.

Like add 5 on the variable or change the string stored there.

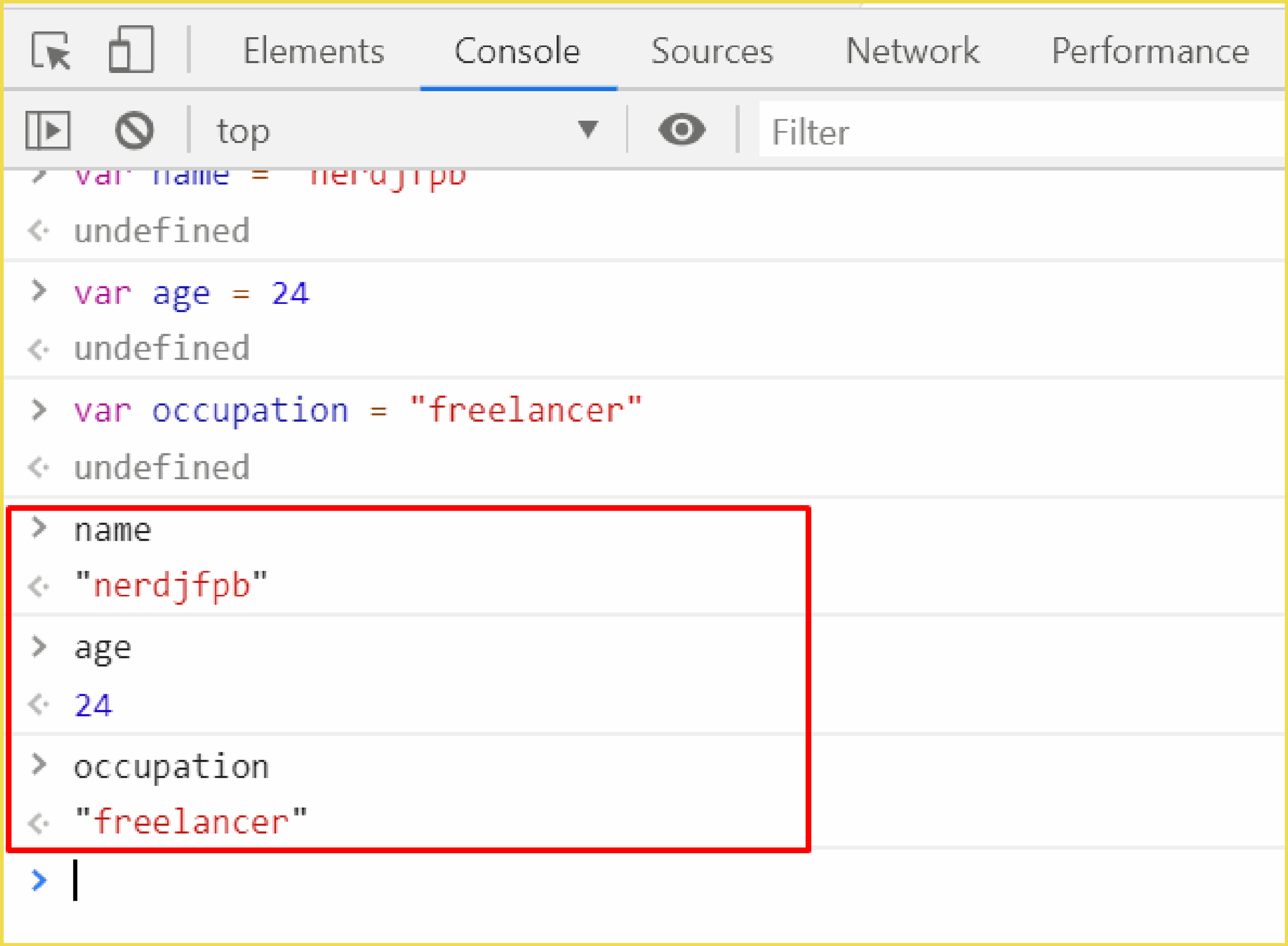
nerdjfpb

In JS we use `var` to initial a variable.
There are some other methods also, We'll learn those method later. Because currently we are working with basic.

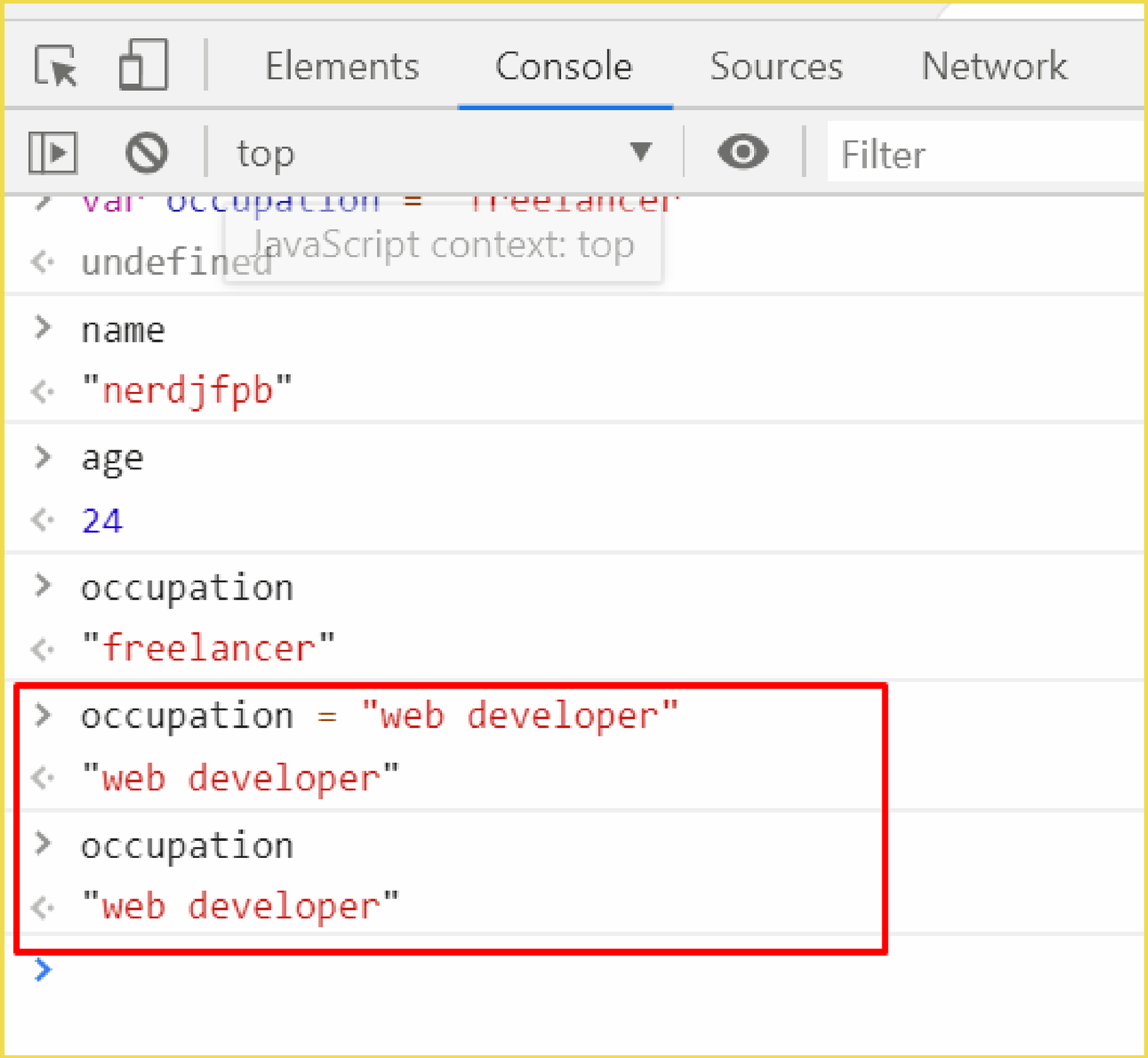
Now we can do `var name = "nerdjfpb"` and some other codes. For now don't think about the undefined.



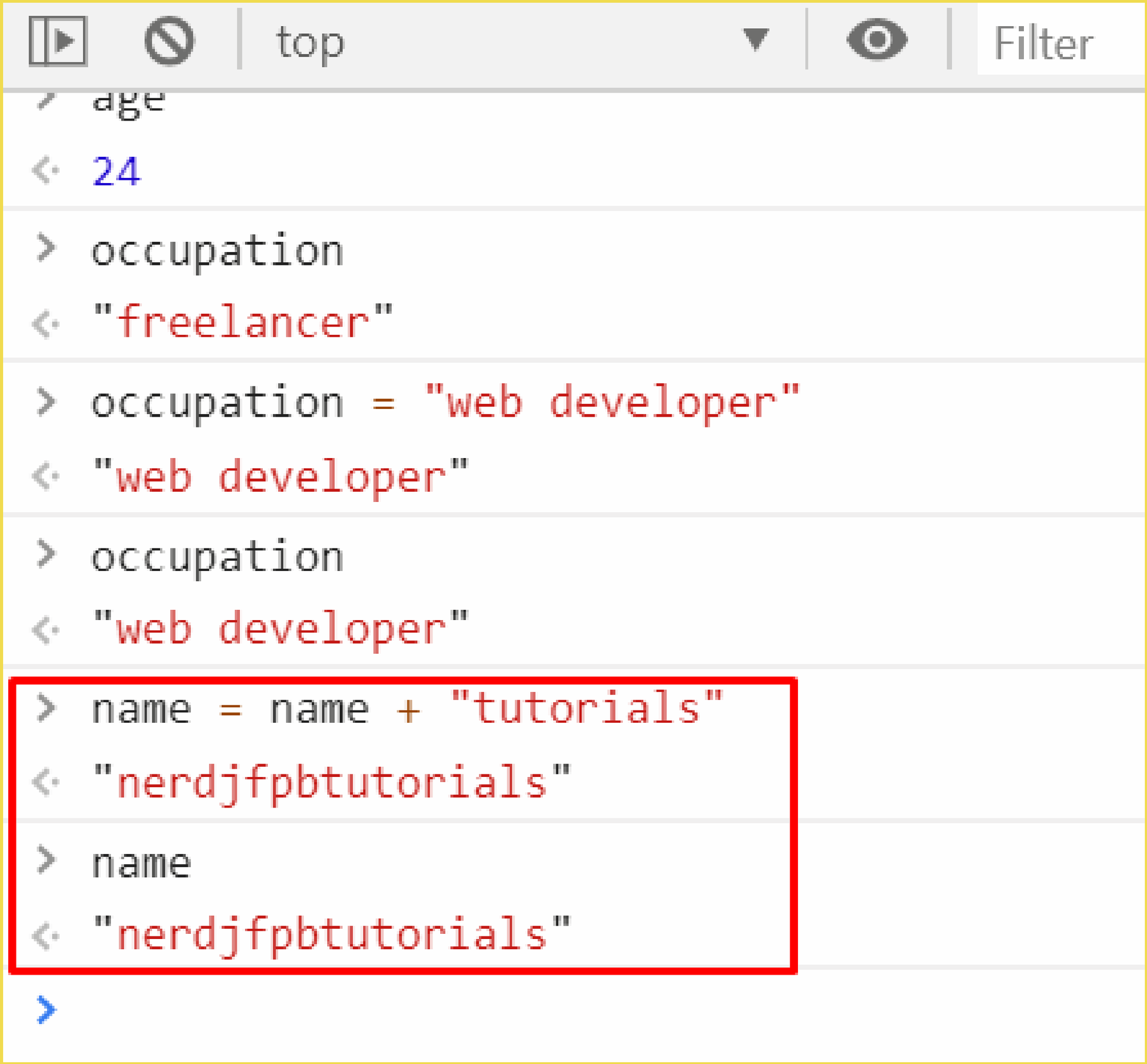
Now we can get value by just calling the variable name.



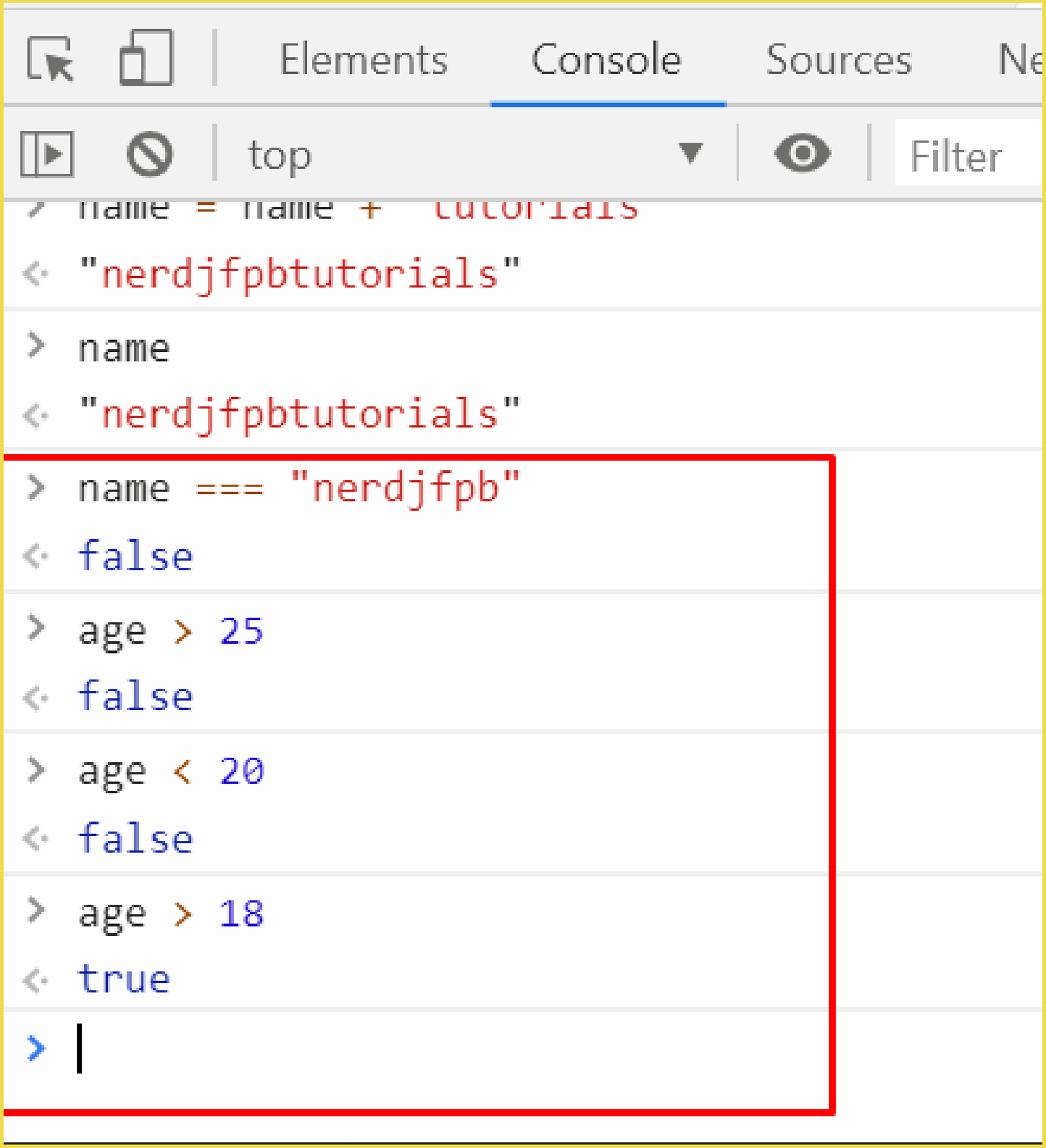
We can do different stuff with the variable, we can change a variable just to new value and it will work like -change the occupation to web developer.



We can add some value with occupation and save it inside of the same variable like-



Also we can do the old things we've done with values. example -



nerdjfpb

nerdjfpb

Keep practicing the stuff to learn more!

nerd_jfpb

nerdjfpb

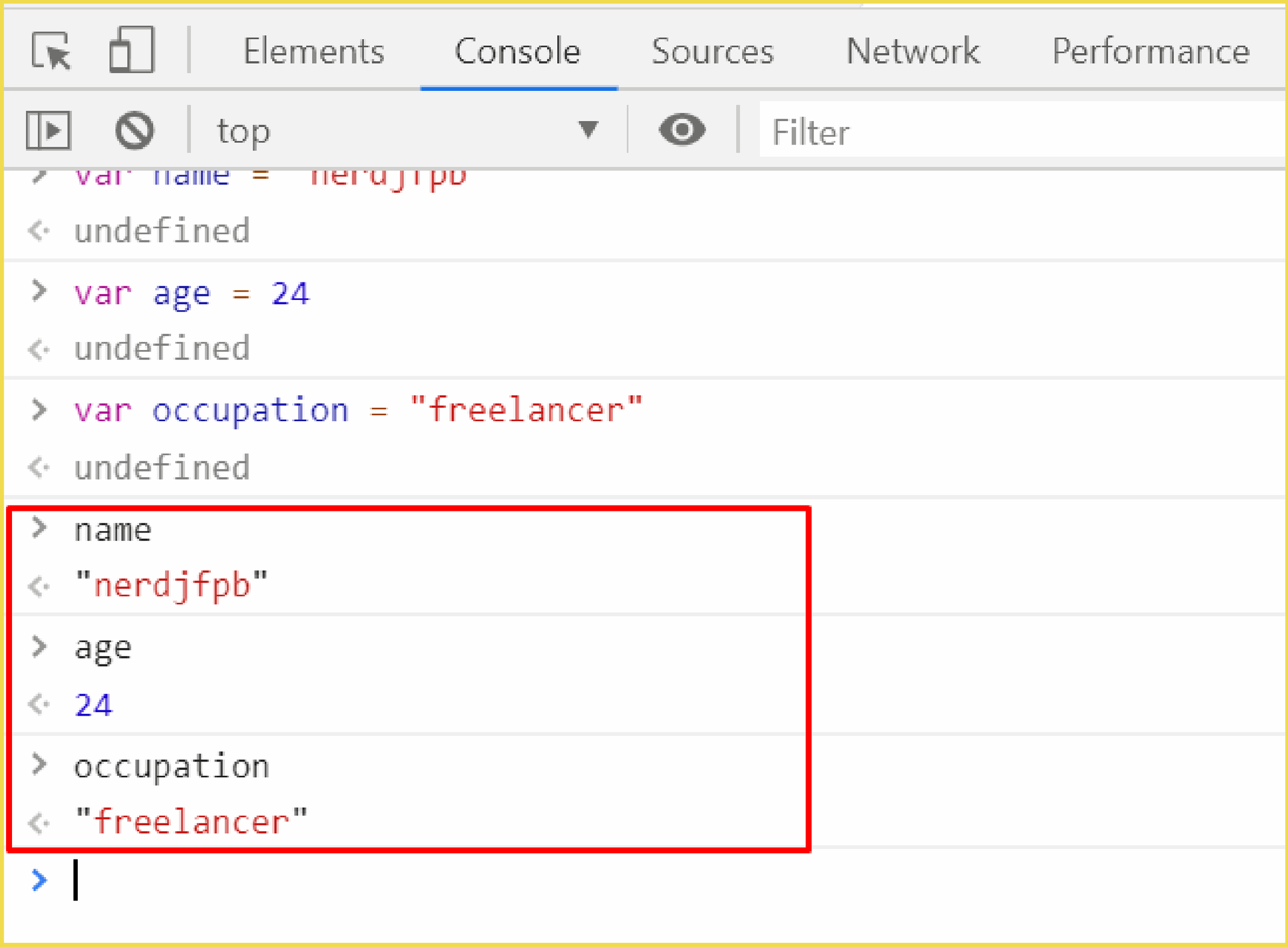
nerdjfpb

More About Variables - JavaScript Series -

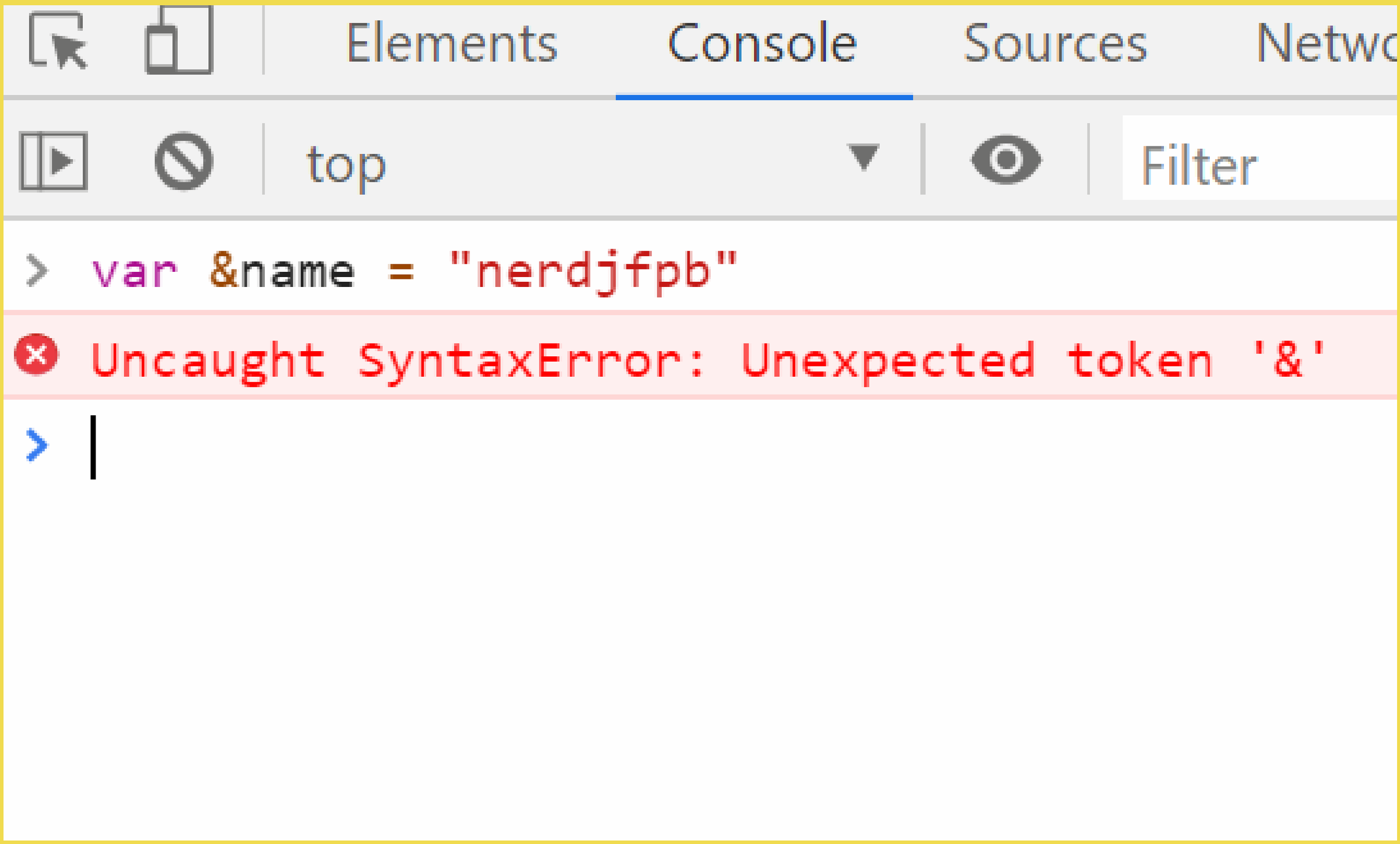
Part 7

nerd_jfpb

In last part we learned about variables right ? Now today we are going to learn a bit more about variables.

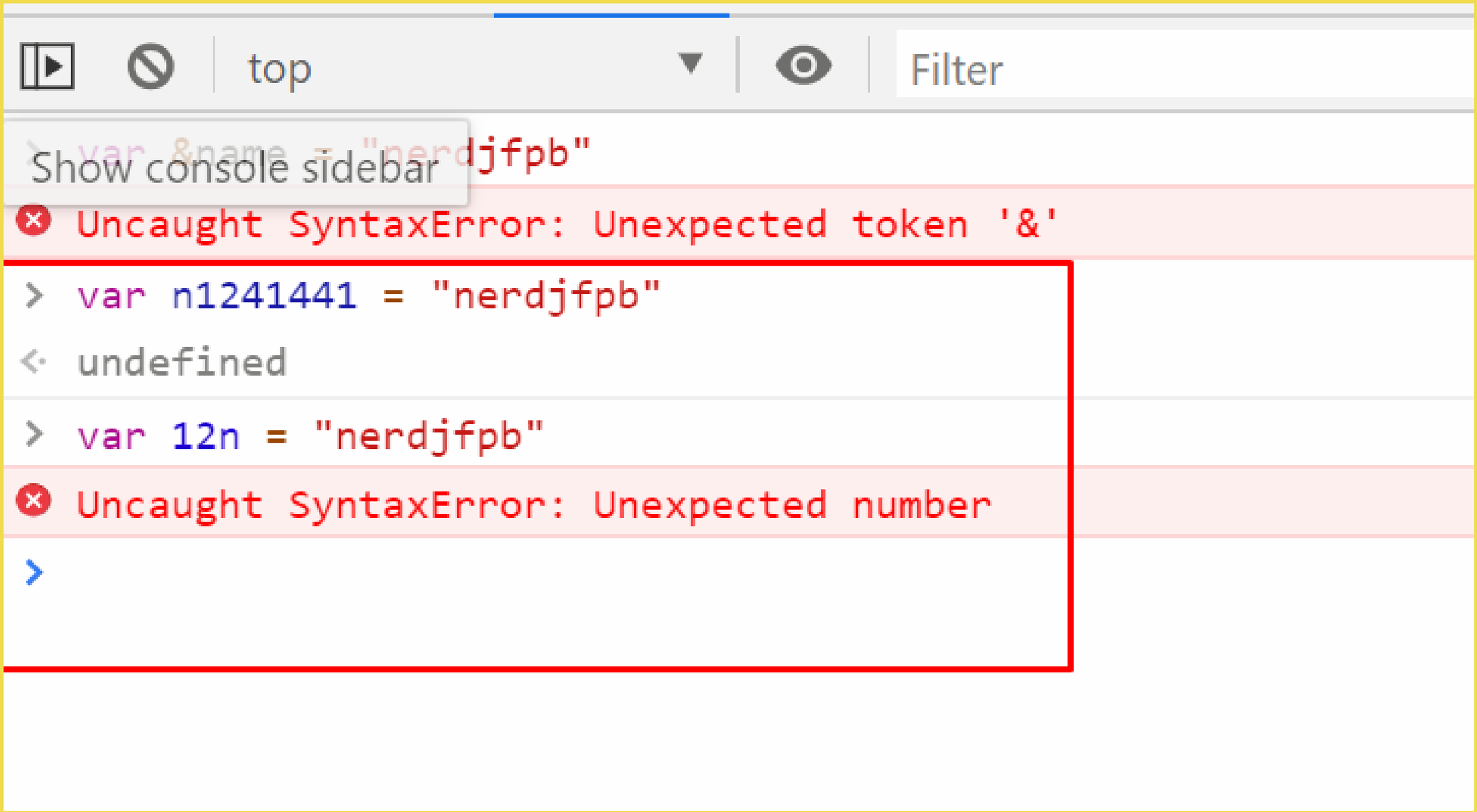


We can't start variable name with symbol. Like -



nerdjfpb

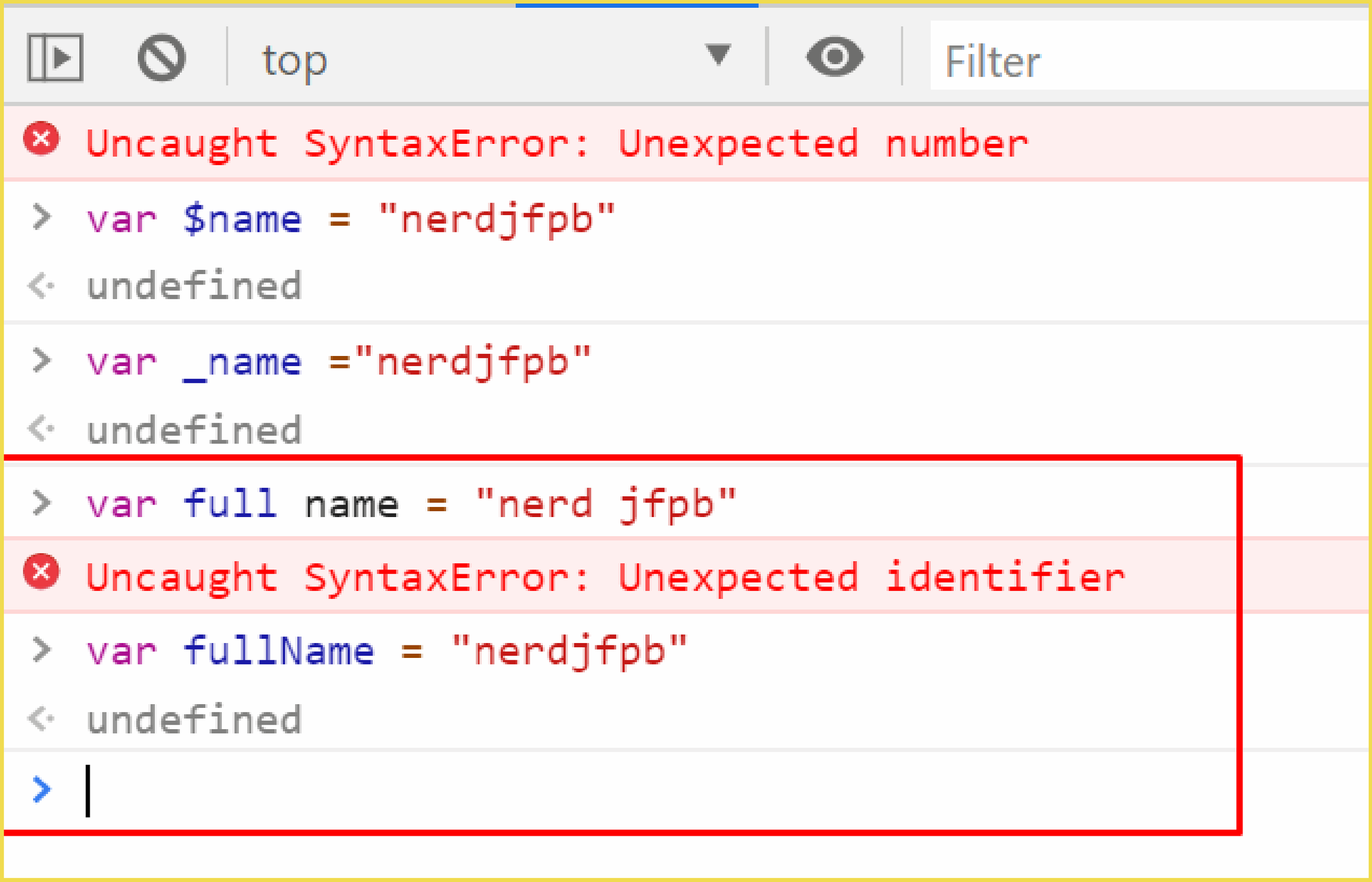
Variable can start with a letter than numbers, it's fine.
But no way to start numbers.



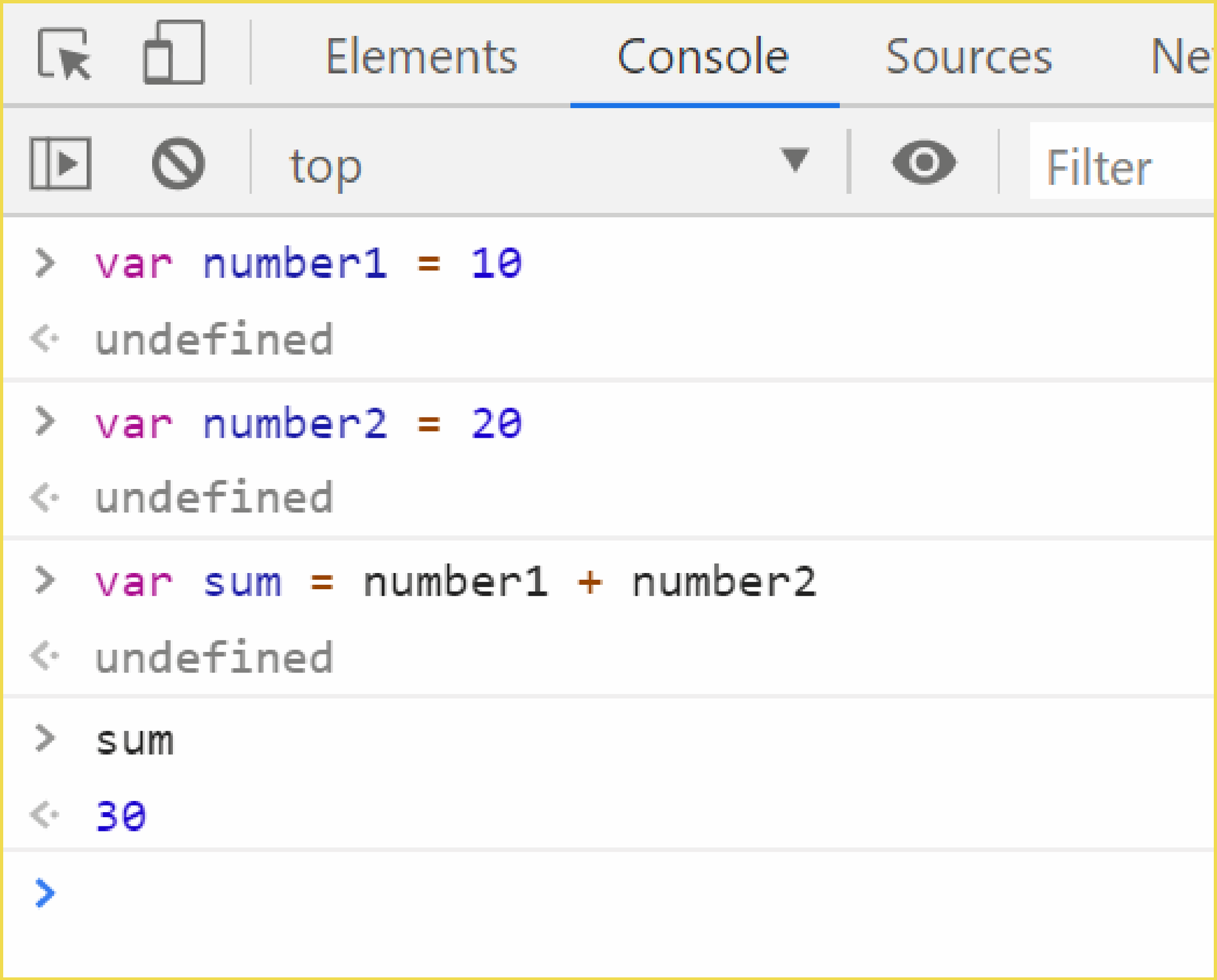
But you can start a variable with _ & $

nerdjfpb

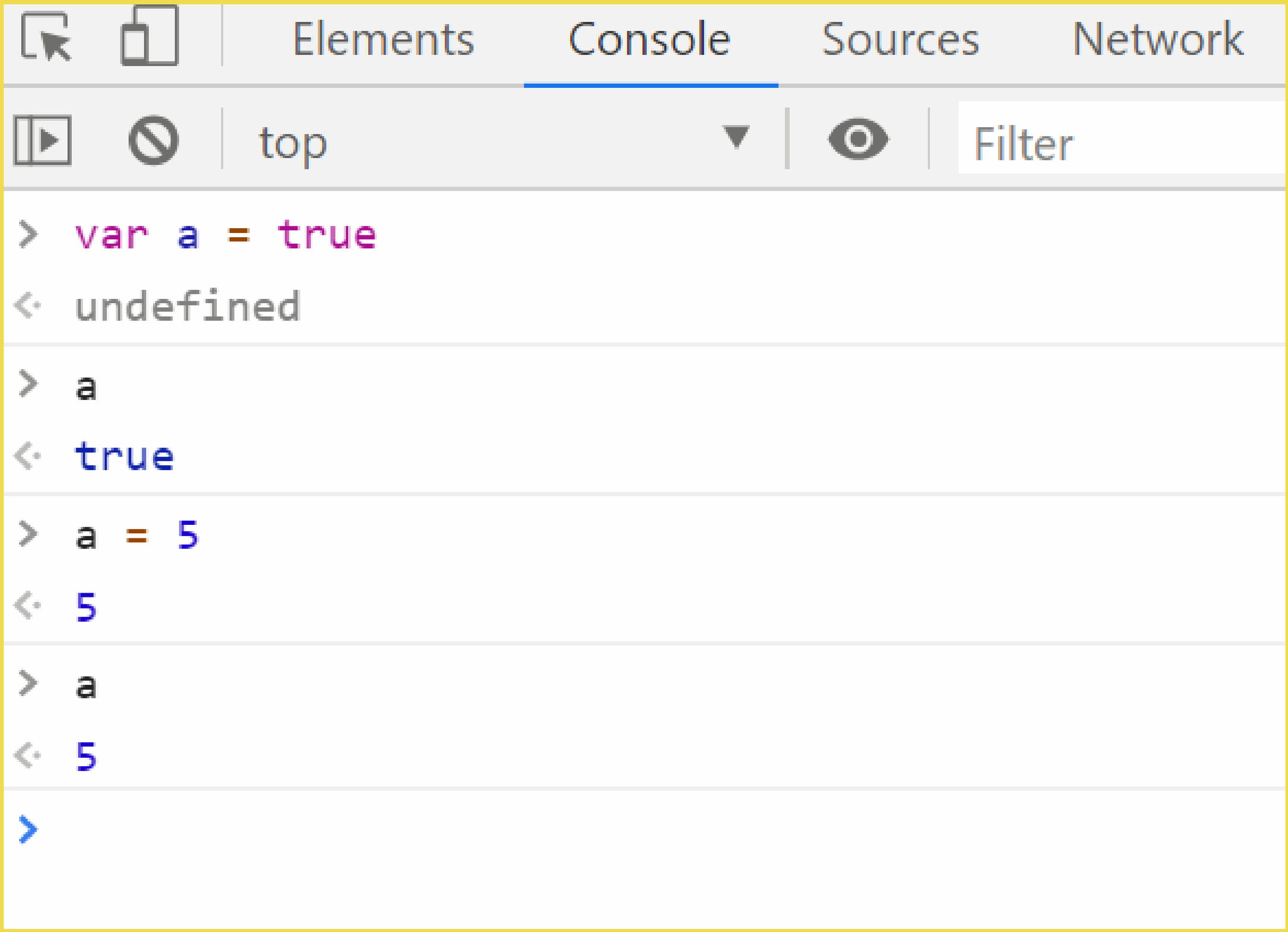
We can't have space in variable name and if we need a long variable name. Then we use camelCase. Which means you'll write in big letter the second word. example -



Let do a little easy trick with variable. First we'll get two variable and add them and store it on a new variable, then we'll see if we got the exact value.

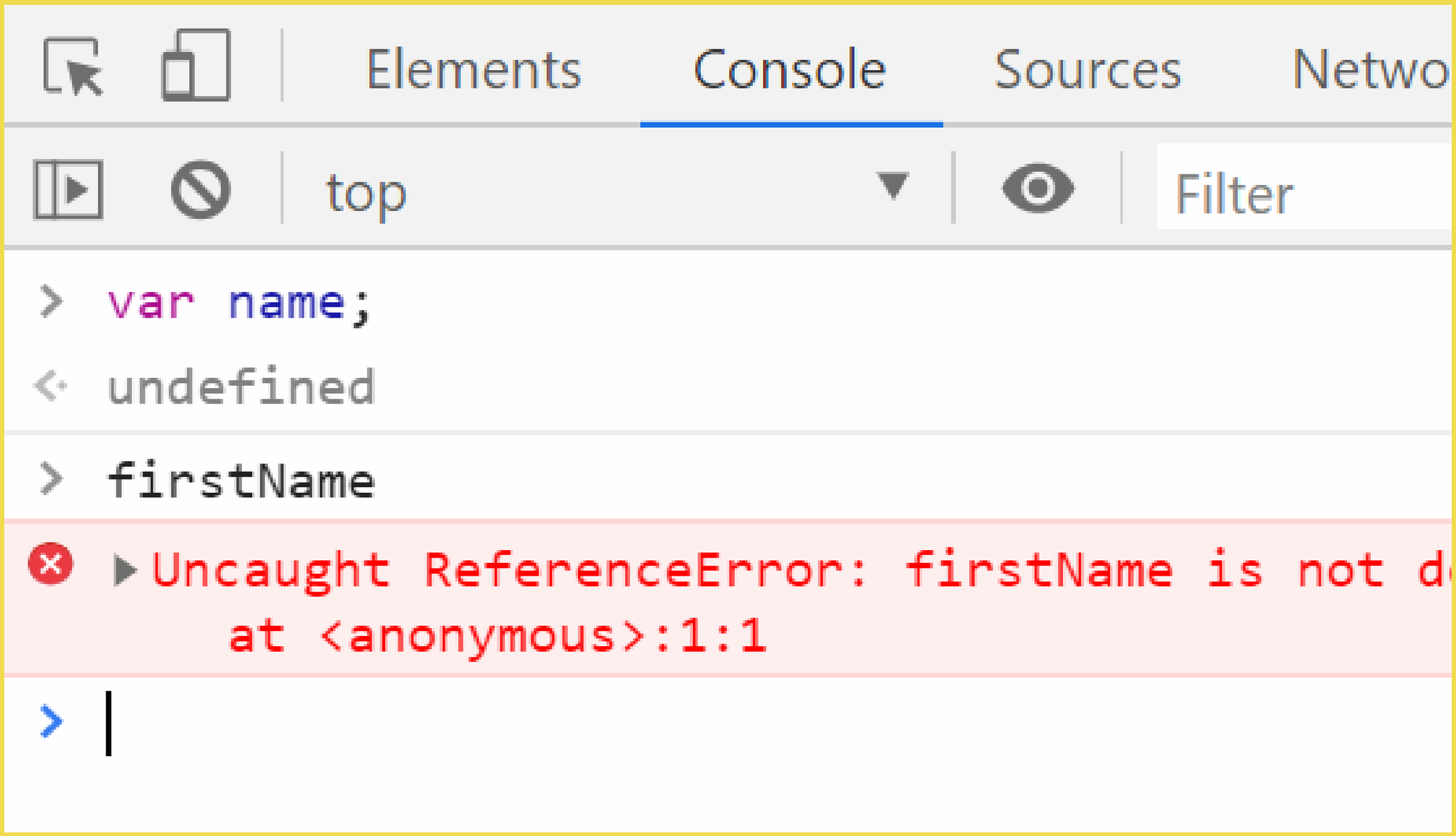


Another thing, we can change the types of variable just by storing that value on there, like -



nerdjfpb

What if we didn't mention any type while initialing a variable. It will be undefined. This is another type of JS, we didn't learn at first. If you never initialize a variable but trying to call it, then it will give you a reference error.



nerdjfpb

nerdjfpb

Do you learn some new stuff today ?

nerd_jfpb

nerdjfpb

nerdjfpb

Conditional Statement - JavaScript Series -

Part 8

nerd_jfpb

nerdjfpb

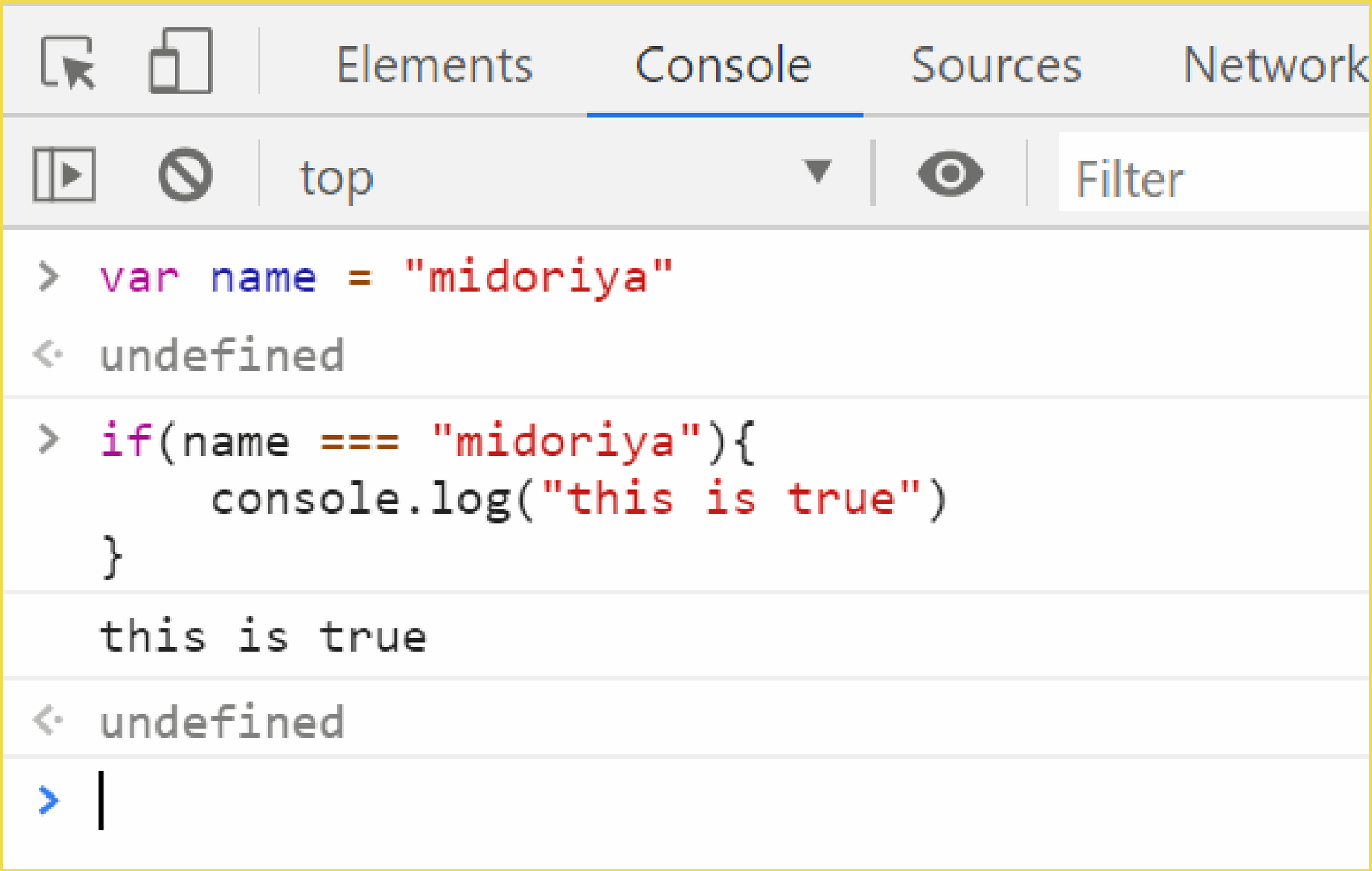
**Up until now we never wrote any condition in our code,
we just wrote one line code and executed those.
But now we are going to learn about conditional statements in Javascript.**

nerdjfpb

There are
if
else
else if &
switch

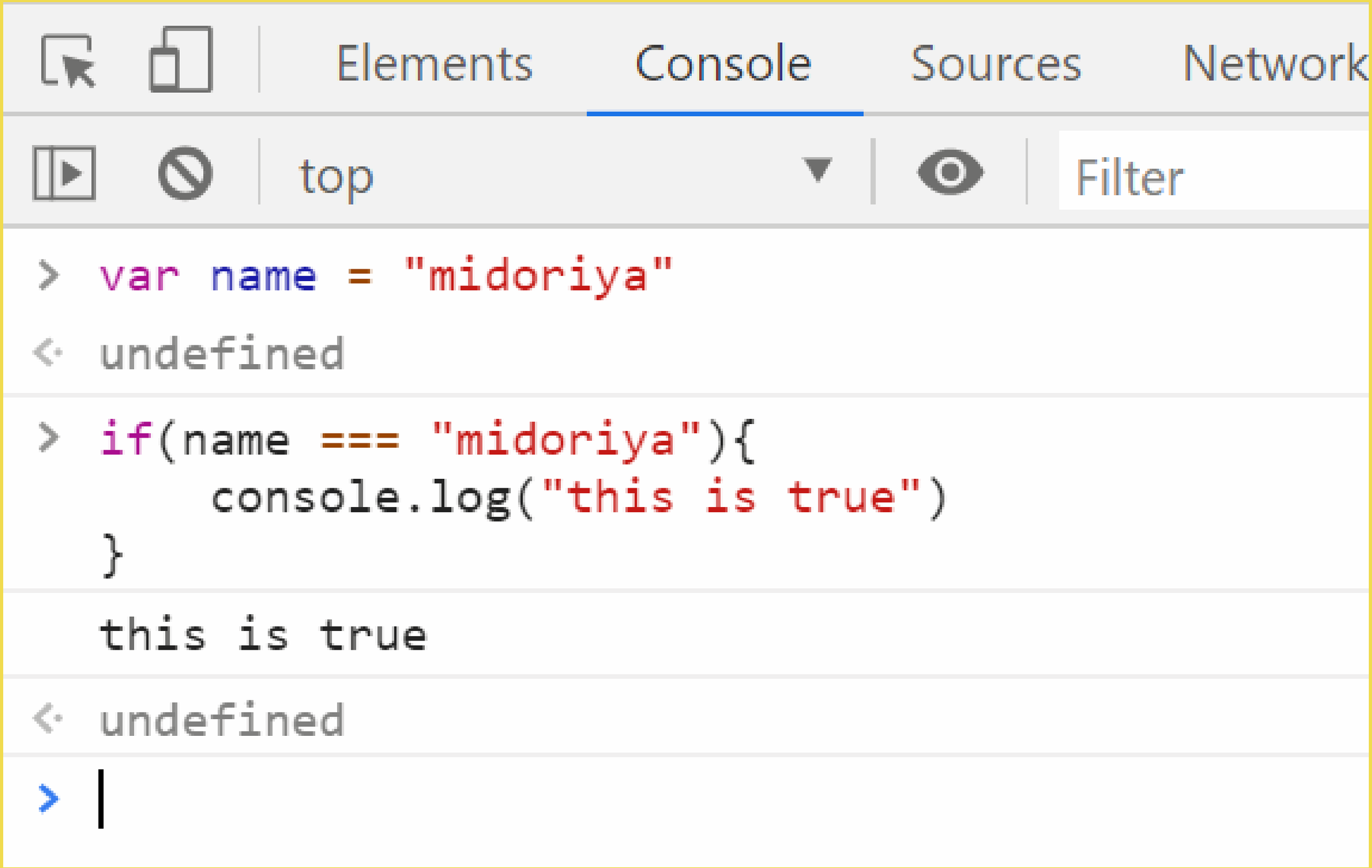
these are available in javascript to write conditions.
You can write ternary operators too.
But for now let's just focus on
if statement.

This is like english. We'll write if first then first brackets and in the brackets we'll write the condition and finally there will be second brackets where we'll write what we want to do. Look at this code -



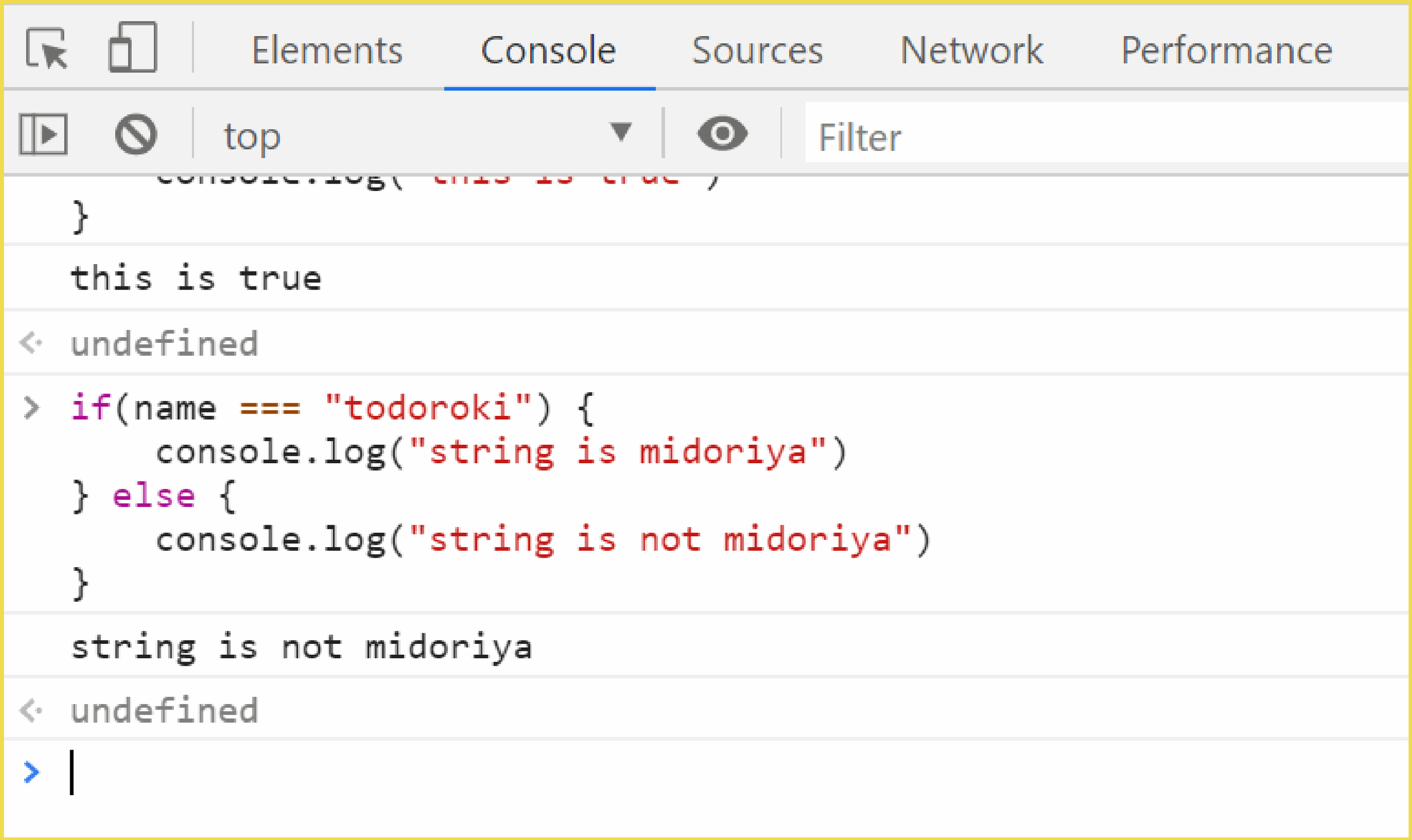
nerdjfpb

Now let's break it down from the way. First we store midoriya string into the name, then we checked if the name is equal to midoriya then we just print in console a new string. This is easy right ?
Just it's almost like plain english.



nerdjfpb

Else part is easier also. If the condition doesn't match then else part will work. Look here -



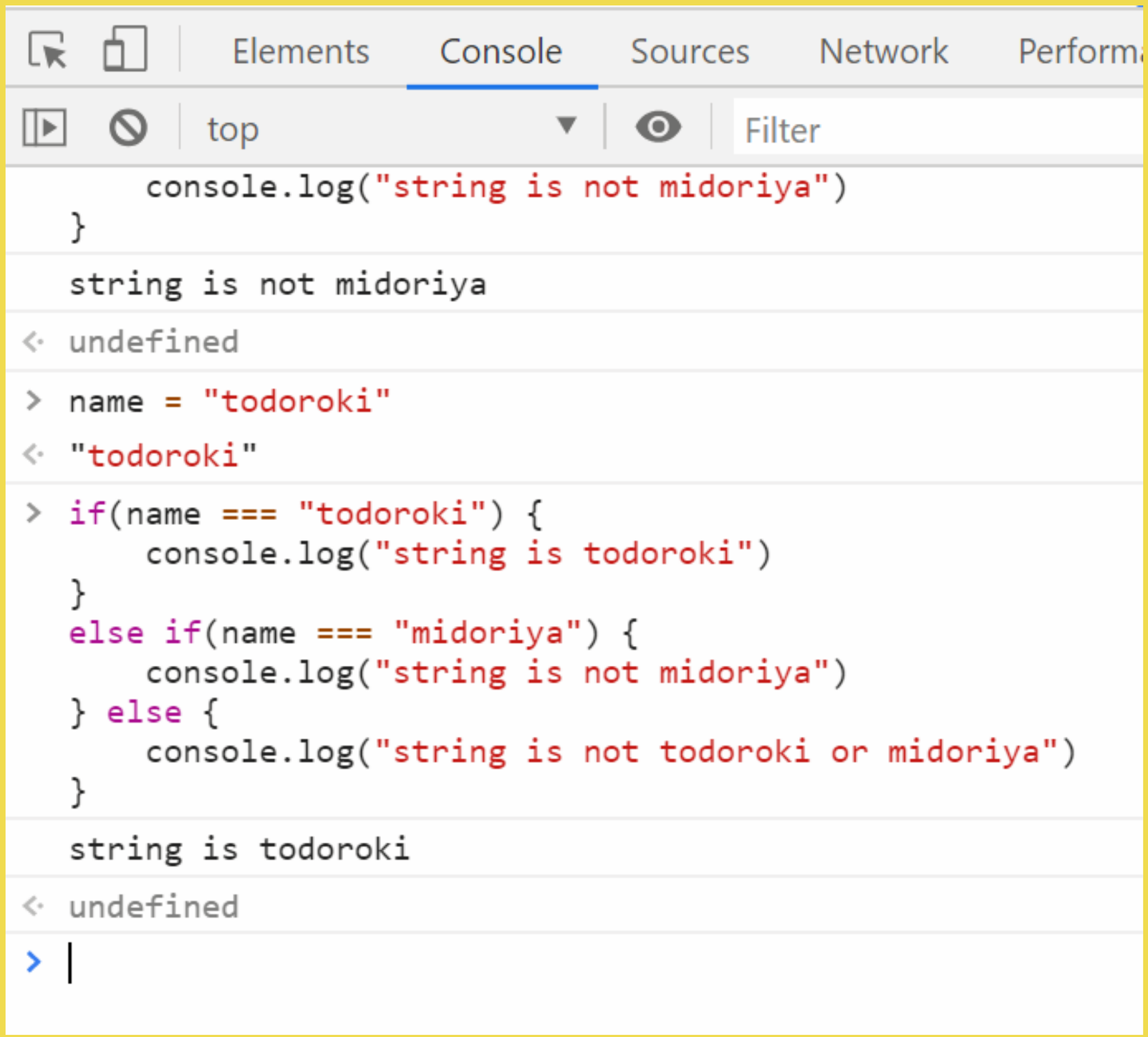
nerd_jfpb

nerdjfpb

Now we'll learn about the else if. But let's change the name to "todoroki" first.

nerdjfpb

Like before if midoriya doesn't match then always the else part was printed, but now we want to check if the string is todoroki or midoriya or none of them. We can do it easily by just using a new else if(conditions) -



So finally what is our code ?

```
if(condition) {
block of code
} else if (condition) {
block of code
} else {
block of code
}
```

Let me know if you understand this part clearly or not!

nerdjfpb

nerdjfpb

More Conditional Statement - JavaScript Series -

Part 9

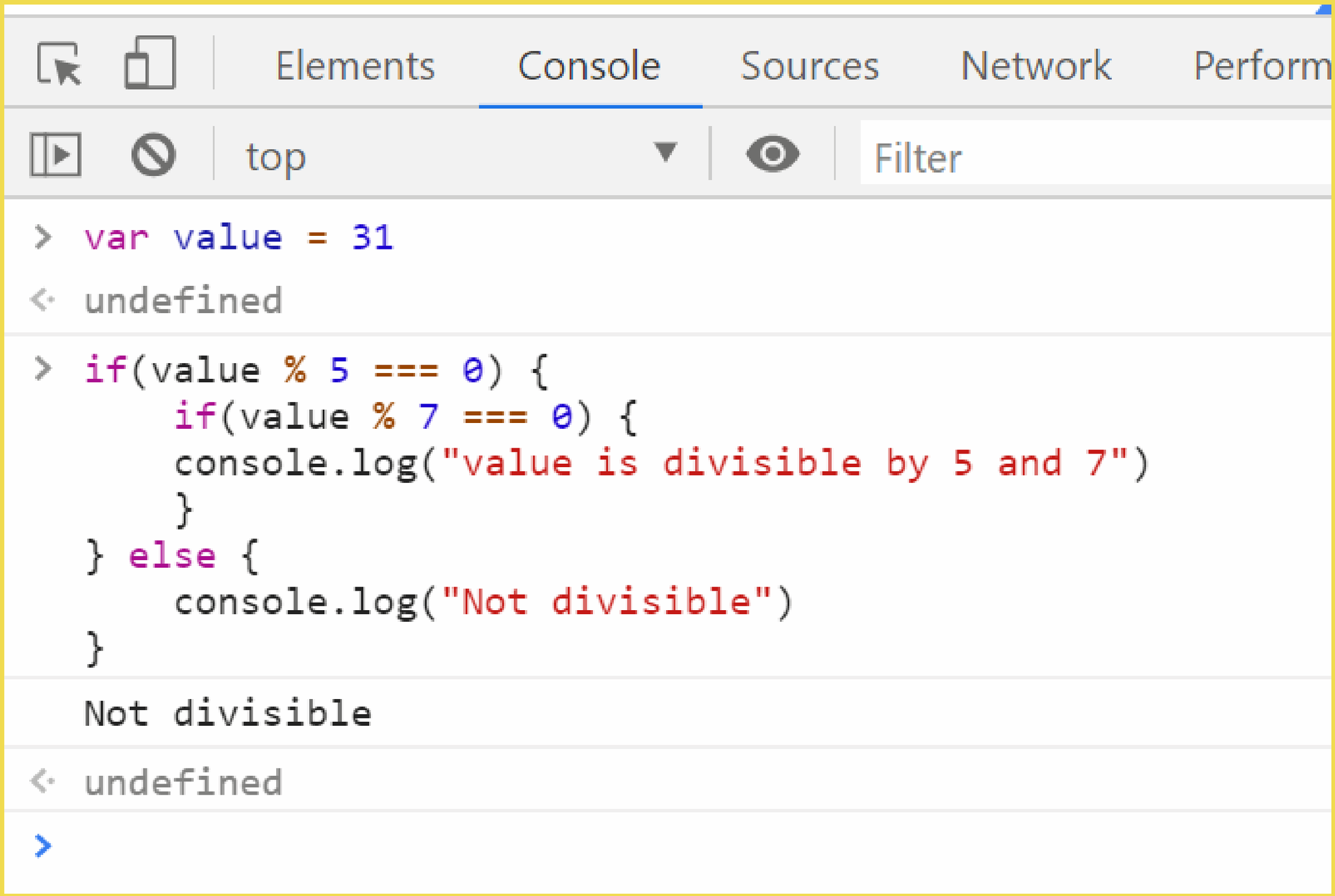
nerd_jfpb

nerdjfpb

In last part we didn't talk much about conditions.

In this part we are going to talk about the conditions.

Suppose we need to found a value which is divisible by 5 and 7 both. How can we do this? The straight forward way will be just write one if inside of another one. Example -



nerdjfpb

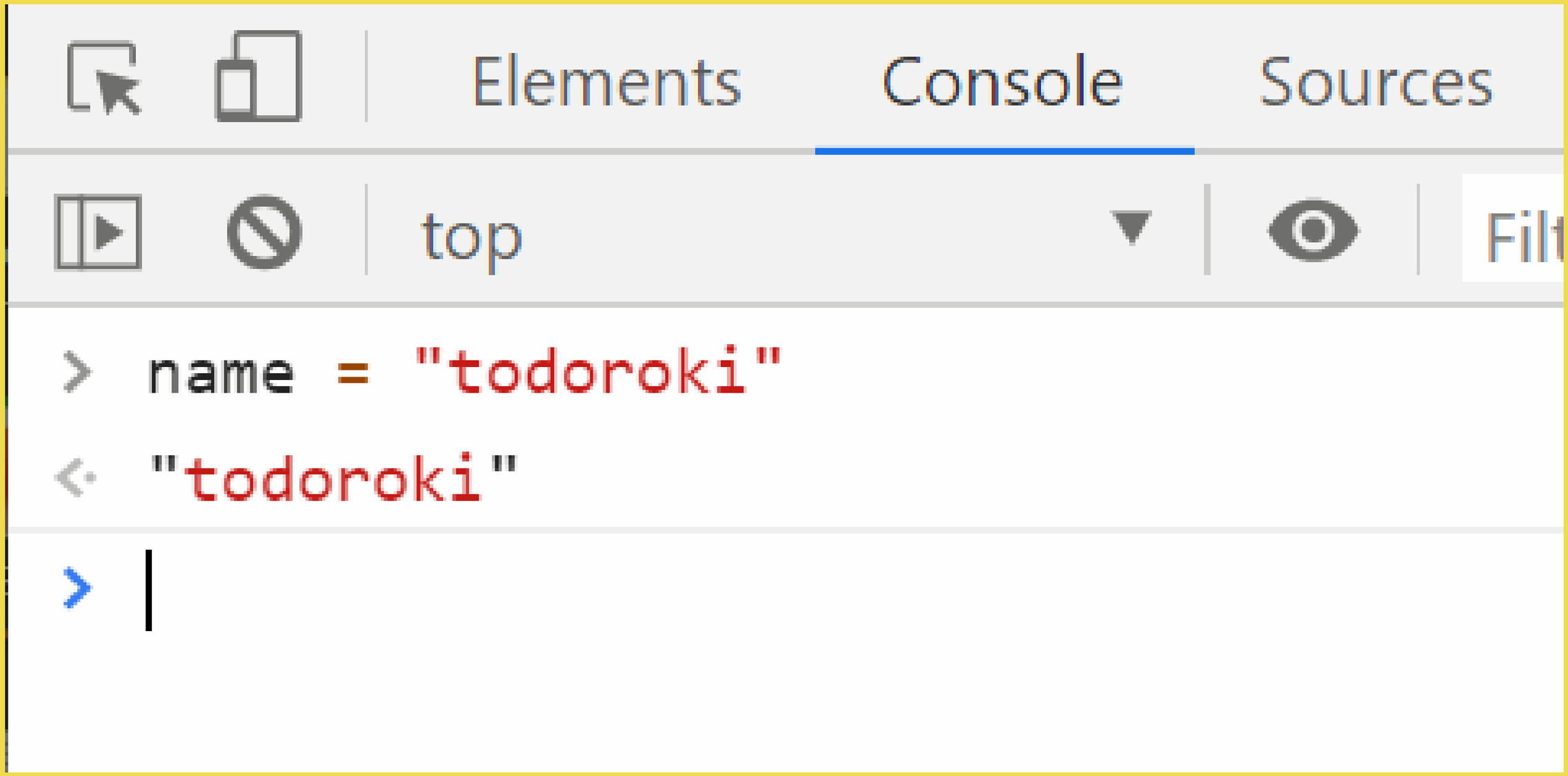
Now we can do two conditions in one if.
By adding && we can do it.
Adding && means we need to satisfy both conditions to go that block.
Example -

```
> if(value % 5 === 0 && value % 7 === 0) {
      console.log("value is divisible by 5 and 7")
  } else {
      console.log("Not divisible")
  }
  Not divisible
< undefined
>
```

nerdjfpb

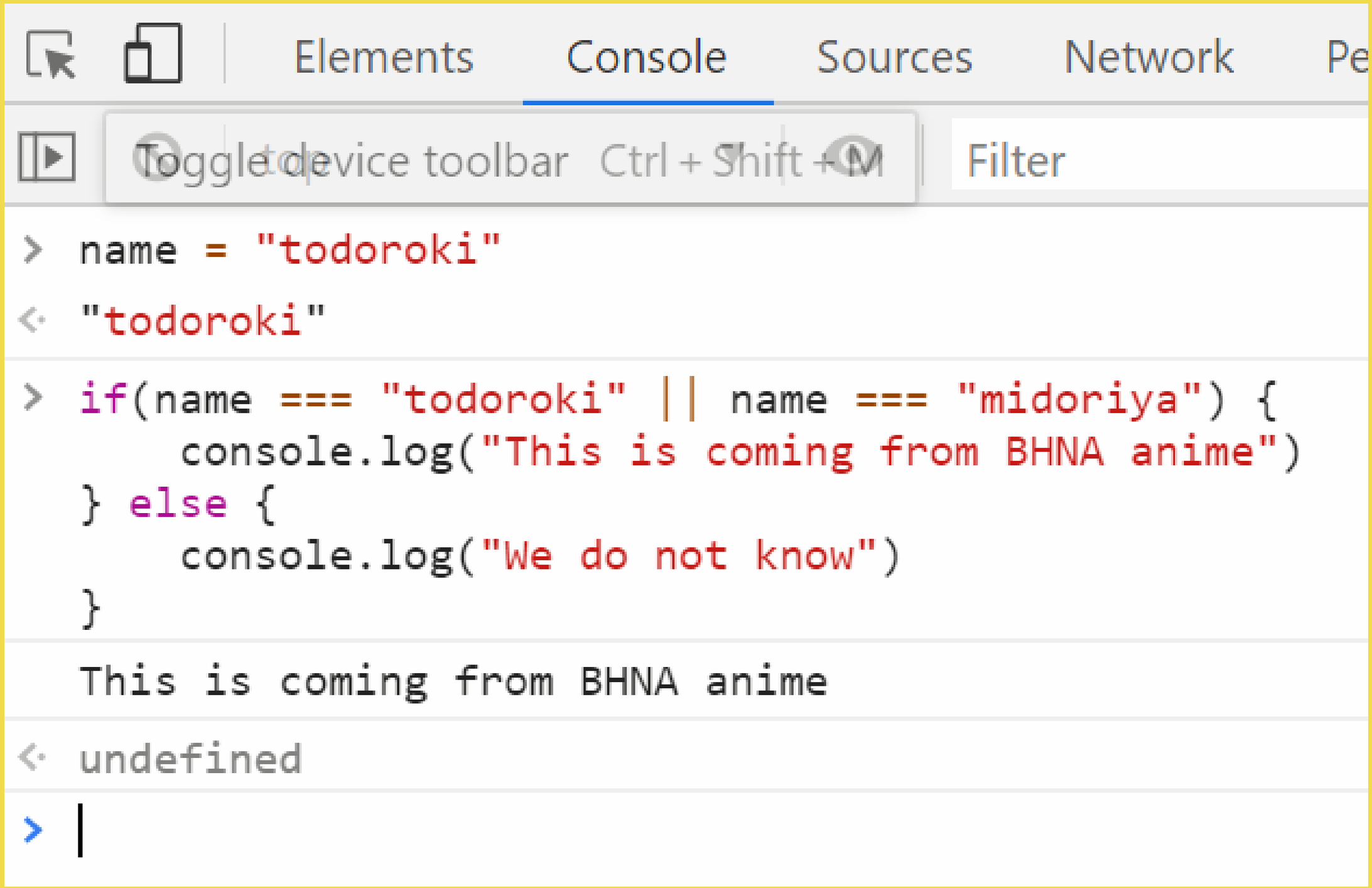
**Suppose we have string which can contain midoriya or todoroki.
If we get any of these, then we'll tell this is coming from BHNA anime or else we'll print - we don't know the source. Let's do it now!**

First we need a name variable and store the value in it -



nerdjfpb

Now lets check the both condition and put || between them. This means OR. If any of these condition approved, then if block will run.



nerd_jfpb

So this was easy right ?

We learned about && which means and.

Also we learn about || which means or.

nerdjfpb

nerdjfpb

Let me know if you have any questions.

nerd_jfpb

nerdjfpb

nerdjfpb

Switch - JavaScript Series -

Part 10

nerd_jfpb

Switch can be used instead of if else.
It's almost like if else too.

```
switch(expression) {
case x:
code block
break;
case y:
code block
break;
default:
code block
}
```

First we need to write the switch then expression, this is mean on which variable you are trying to apply.

nerdjfpb

Cases are the specific if.
Like case "todoroki" will be same of variable==="todoroki"

Let's try the last code in switch.

First we need the variable name. Then we'll write the switch for the name - `switch(name) {}` everything will go inside the second brackets.

Finally -

Elements Console Sources Network Performance

top Filter

```
> name = "todoroki"
"todoroki"
> switch(name){
      case "todoroki":
          console.log("This is coming from BHNA");
          break;
      case "midoriya":
          console.log("This is coming from BHNA");
          break;
      default:
          console.log("We do not know the anime name");
  }
  This is coming from BHNA
undefined
>
```

nerdjfpb

This is almost same right ?
So which one we should use ?

"As it turns out, the switch statement is faster in most cases when compared to if-else, but significantly faster only when the number of conditions is large. The primary difference in performance between the two is that the incremental cost of an additional condition is larger for if-else than it is for switch. Therefore, our natural inclination to use if-else for a small number of conditions and a switch statement for a larger number of conditions is exactly the right advice when considering performance. Generally speaking, if-else is best used when there are two discrete values or a few different ranges of values for which to test. When there are more than two discrete values for which to test, the switch statement is the most optimal choice."
[oreilly.com]

So which one you are goint to use ?

nerdjfpb

nerdjfpb

JavaScript Series

Setup Code Editor

nerd_jfpb

Part 11

Up until now we just worked with the console.

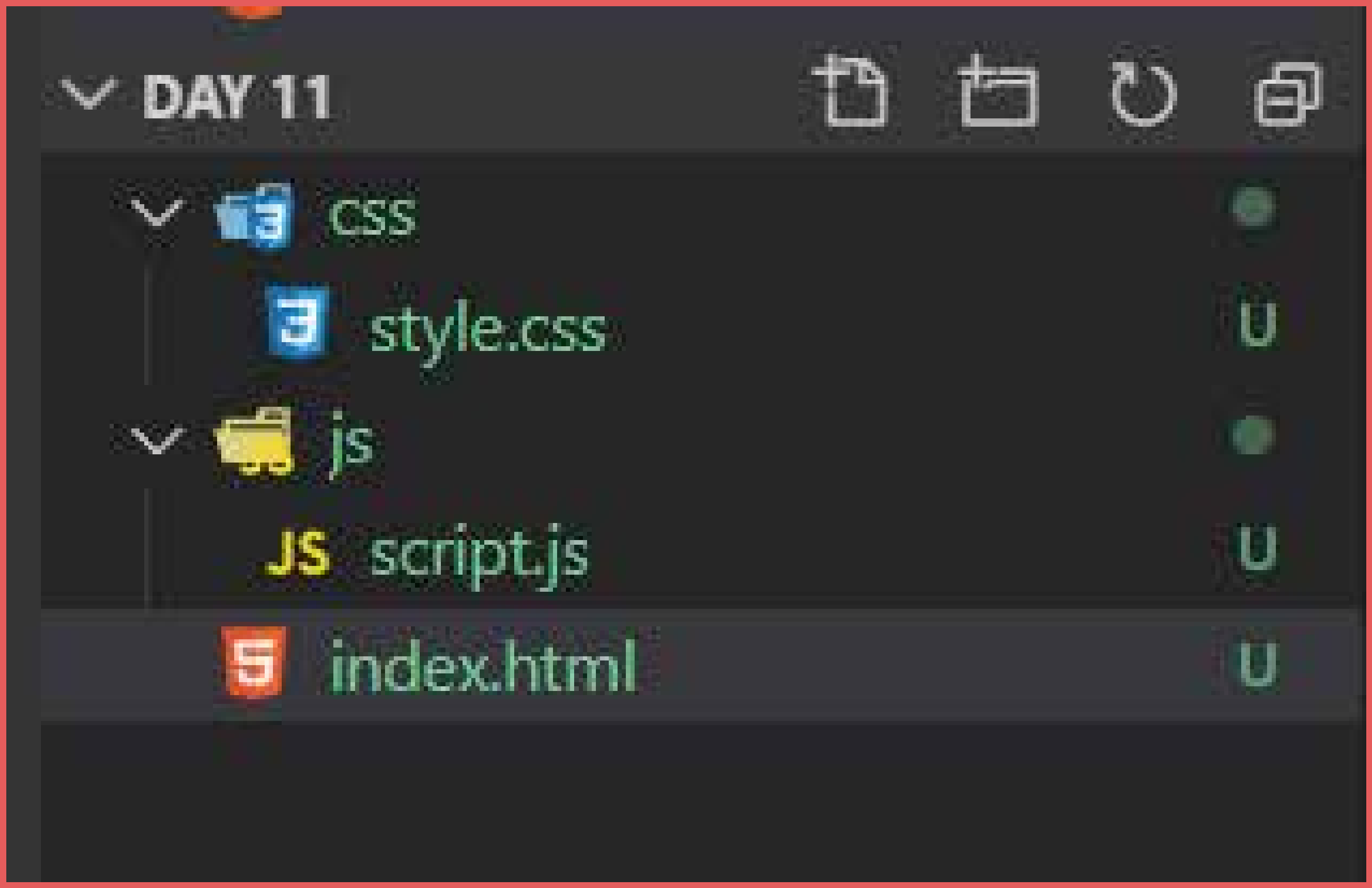
But now we'll work on a code editor.

nerdjfpb

I'm mostly a vscode fan so I'm going to use vscode.

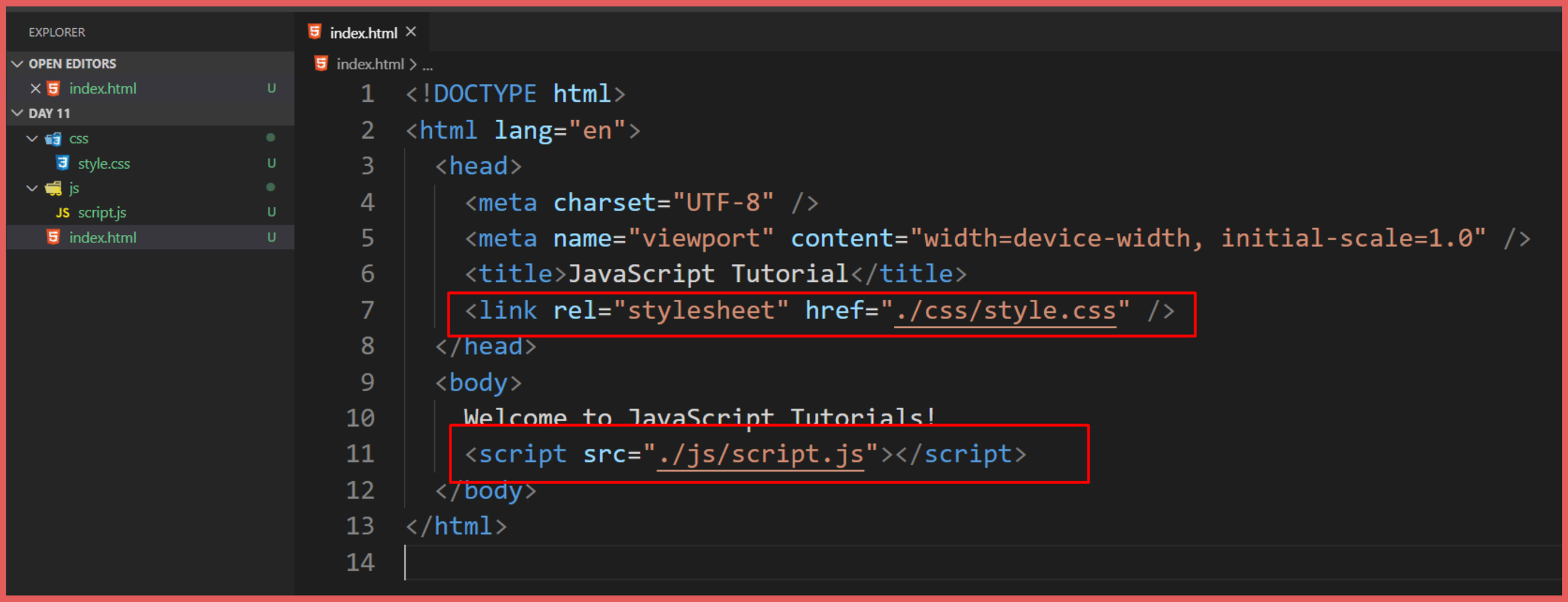
You can use whatever you want.

For folder I'll create like -



nerdjfpb

There is nothing in css or js file. Now we'll include this with the index.html

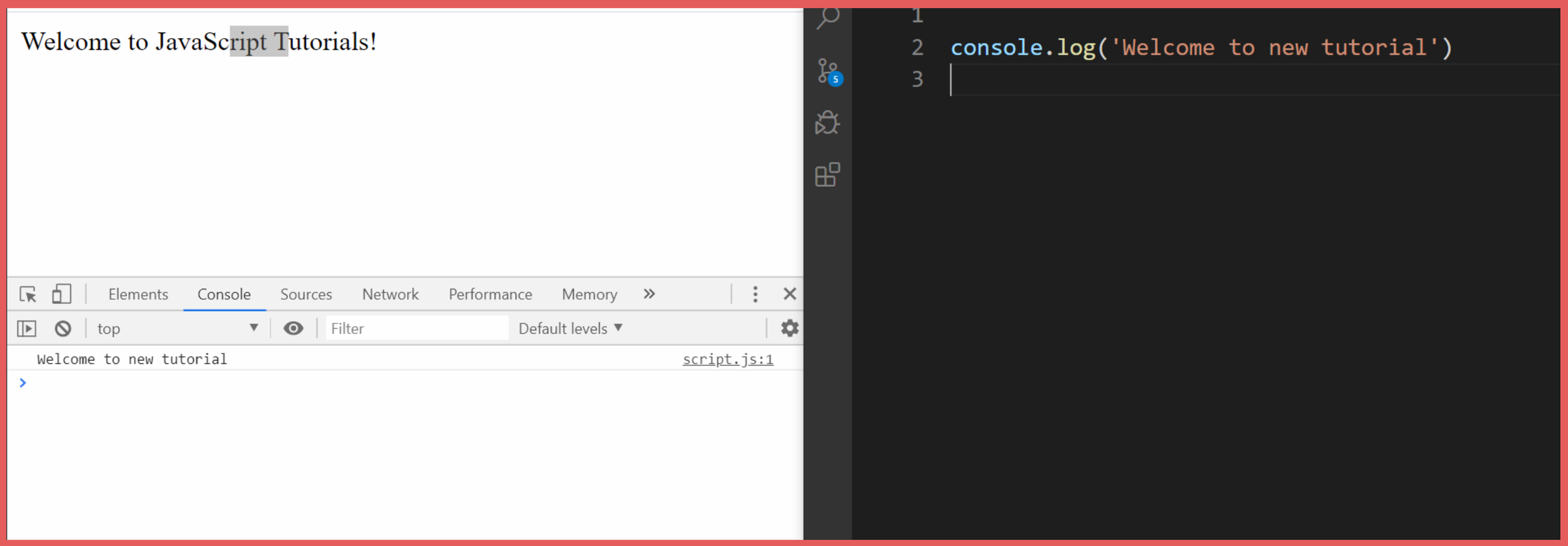


I'm using the live server extensions to run the files.

Welcome to JavaScript Tutorials!

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport"
    content="width=device-width,
    initial-scale=1.0" />
    <title>JavaScript Tutorial</title>
    <link rel="stylesheet" href="./css/style.
    css" />
  </head>
  <body>
    Welcome to JavaScript Tutorials!
    <script src="./js/script.js"></script>
  </body>
</html>
```

Now we'll just console in our script.js file and we can see the result on our console of the website.



I am using the live server extension which give me the way to run live the code

Live Server ritwickdey.liveserver

Ritwick Dey | 3,851,221 | ★★★★½ | Repository | License

Launch a development local Server with live reload feature for static & dynamic pages

Disable ▾ Uninstall *This extension is enabled globally.*

Details Contributions Changelog

[I'm sorry but I'm super busy now. If you want to be a maintainer of the project, please feel free to contact me! You've to be passionate about programming]

Live Server

Live Server loves 💔 your multi-root workspace

> **Live Server for server side pages like PHP. Check Here**

> ***[For 'command not found error' #78]***

vscode marketplace v5.6.1 | downloads 7M | rating 4.5/5 (222)

travis branch passing | appveyor branch failing | license MIT

Launch a local development server with live reload feature for static & dynamic pages.

nerdjfpb

I use the prettier which helps me to easily arrange the code, so I don't have any messy codes.

index.html | JS script.js | *Extension: Prettier - Code formatter* ×

Prettier - Code formatter esbenp.prettier-vscode

Esben Petersen | 4,894,653 | ★★★★☆ | Repository | License

Code formatter using prettier

Disable ▾ | Uninstall | *This extension is enabled globally.*

Details | Contributions | Changelog

Prettier Formatter for Visual Studio Code

Prettier is an opinionated code formatter. It enforces a consistent style by parsing your code and re-printing it with its own rules that take the maximum line length into account, wrapping code when necessary.

JavaScript · TypeScript · Flow · JSX · JSON
CSS · SCSS · Less
HTML · Vue · Angular
GraphQL · Markdown · YAML
Your favorite language?

Installation

Install through VS Code extensions. Search for `Prettier - Code formatter`

Visual Studio Code Market Place: Prettier - Code formatter

Can also be installed in VS Code: Launch VS Code Quick Open (Ctrl+P), paste the following command, and press enter.

```
ext install esbenp.prettier-vscode
```

Is this lot to take or this is easy ?

nerdjfpb

nerdjfpb

JavaScript Series

Loop

nerd_jfpb

Part 12

nerdjfpb

We are going to learn a really important topic today.
I know this one is overwhelming at first,
but if you practice then you'll get this properly!
Don't just see the post of today, work on it too!

nerd_jfpb

nerdjfpb

There are some different kind of loop in js but all they do the same work.

for - loops through a block of code a number of times

for/in - loops through the properties of an object

for/of - loops through the values of an iterable object

while - loops through a block of code while a specified condition is true

do/while - also loops through a block of code while a specified condition is true (w3schools)

Most popular is for and while.

We'll learn both in your practice.

nerdjfpb

Sometimes we need to do the same task again and again.
Like for adding sum of 10, 12 and 30 we need to add all these.
First we need to work 10 + 12 then 22 + 30 again right ?
Or we can add all three at a time, but suppose there is huge numbers like 500 numbers.
What to do ?

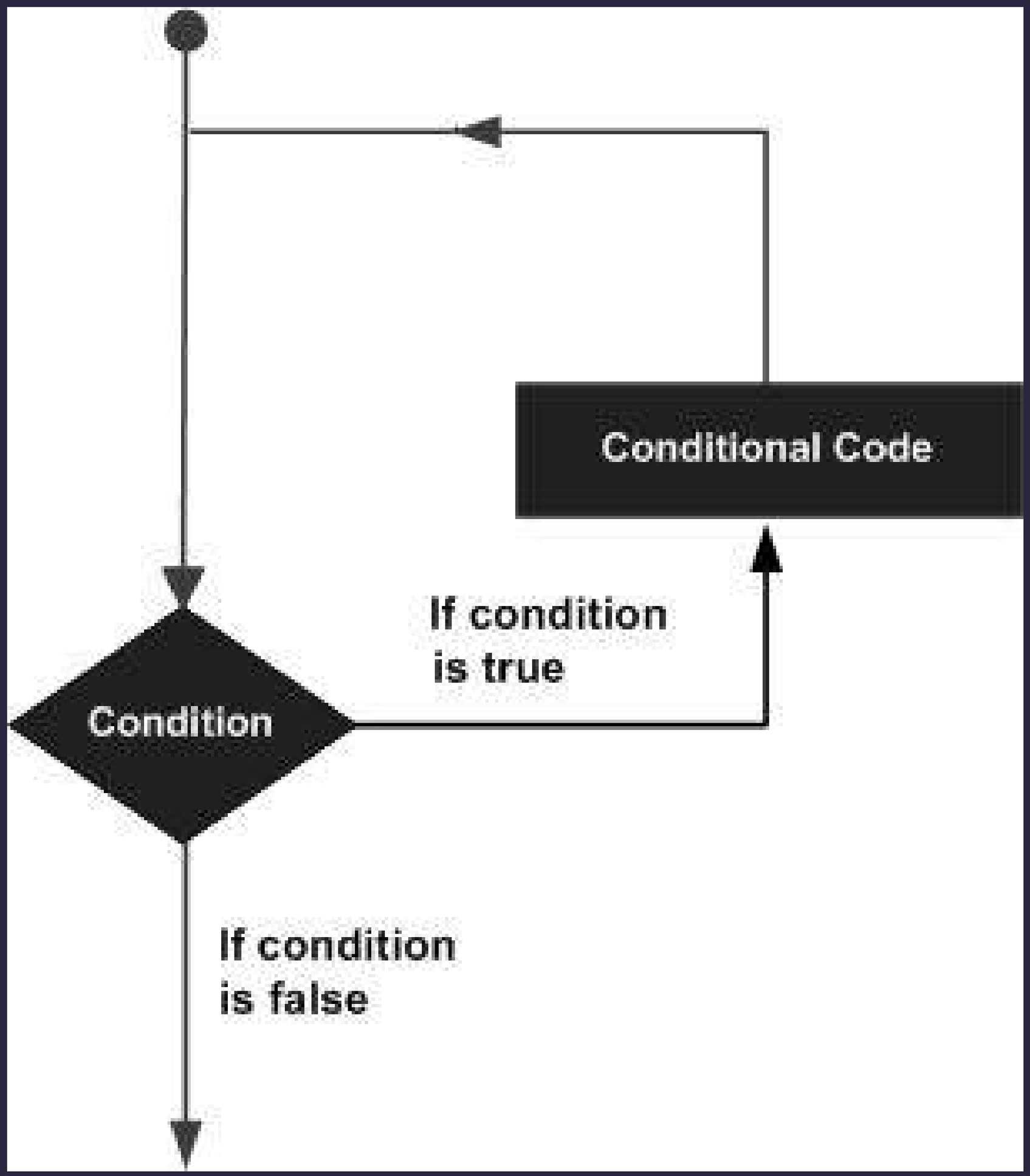
nerdjfpb

We need to go through each and every number and add that to our total value.

Which is going to start from zero right?

We can do this work by loop.
We can terminate doing the same task again and again by using loop.

Lets see this



Understand something ?

We'll work on loop code next tutorial.

nerdjfpb

nerdjfpb

JavaScript Series

While Loop

nerd_jfpb

Part 13

nerdjfpb

Suppose we are going to print 1 to 10 on the console.
But how can we do it ?
We can do it easily

```
`console.log(1)
console.log(2)
...
console.log(9)
console.log(10)
`
```

But this is not a good way to do it.
Currently we can do it because it's only ten times.

But suppose we need to print 1 – 100.
How to do it ?

This is where we use loops. We're going to use `while` loop today!

nerdjfpb

While is easy.
Just remember what we have learned in our last tutorial.
While loop syntax is -

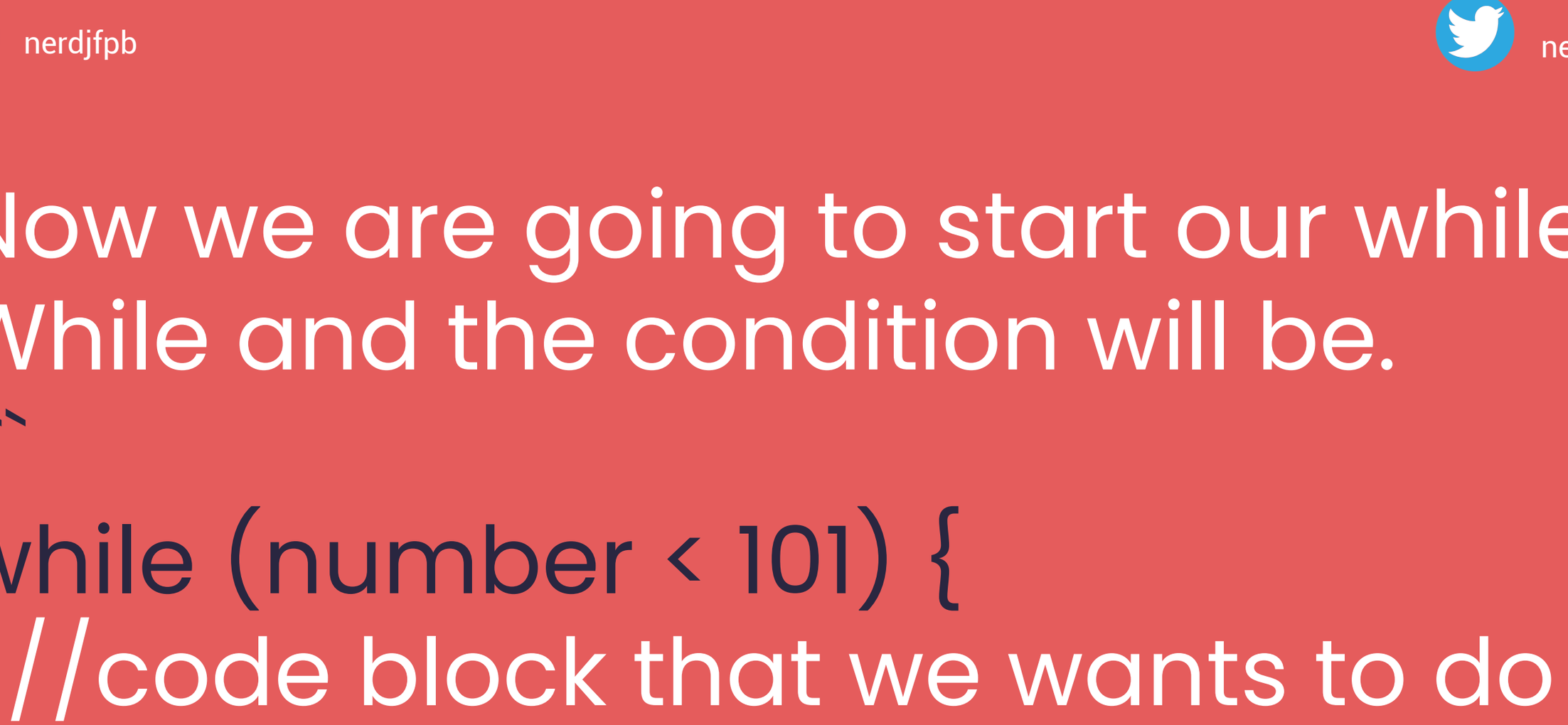
```
while (condition) {
  // code block to be executed
}
```

nerd_jfpb

Let's write some real codes now.
If we want to print 1 - 100 then we are going to store the values in variable and we are going to start from 1.
So `var number = 1`

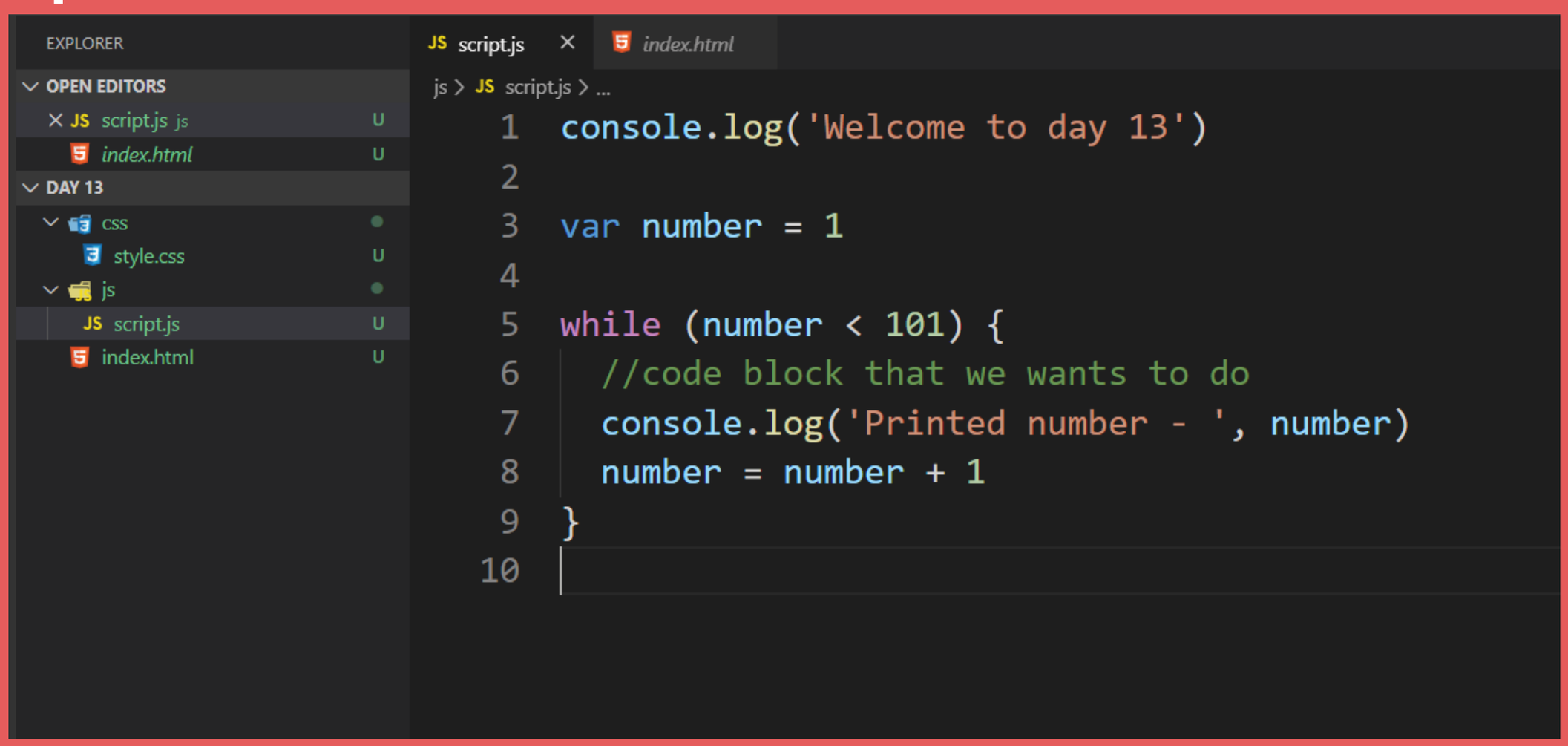
Now we are going to start our while.
While and the condition will be.

```
while (number < 101) {
  //code block that we wants to do
}
```



nerdjfpb

Now we are going to print the values from the first one. We need to increase the value of number to break the condition, if the condition doesn't break then it will stuck forever. So we always break the loop condition.



nerd_jfpb

See the result in browser.

Welcome to JavaScript Tutorials!

Elements Console Sources Network Performance Memory Application Security Audits

top Filter Default levels

```
Welcome to day 13                                   script.js:1
Printed number - 1                                  script.js:7
Printed number - 2                                  script.js:7
Printed number - 3                                  script.js:7
Printed number - 4                                  script.js:7
Printed number - 5                                  script.js:7
Printed number - 6                                  script.js:7
Printed number - 7                                  script.js:7
Printed number - 8                                  script.js:7
Printed number - 9                                  script.js:7
Printed number - 10                                 script.js:7
Printed number - 11                                 script.js:7
Printed number - 12                                 script.js:7
Printed number - 13                                 script.js:7
Printed number - 14                                 script.js:7
Printed number - 15                                 script.js:7
Printed number - 16                                 script.js:7
Printed number - 17                                 script.js:7
Printed number - 18                                 script.js:7
```

Do you understand the while loop ?

nerdjfpb

nerdjfpb

JavaScript Series

For Loop

nerd_jfpb

Part 14

nerdjfpb

In last lesson we learn about the while loop,
but there is another loop which is really popular and people use this.
So we're going to learn it too.

nerdjfpb

Suppose we are going to print 1 to 10 on the console. But how can we do it ? We can do it easily

```
console.log(1)
console.log(2)
...
console.log(9)
console.log(10)
```

But this is not a good way to do it. Currently we can do it because it's only ten times.
But suppose we need to print 1 - 100. How to do it ?

nerdjfpb

nerdjfpb

This is where we use loops.

We're going to use `For` loop today!

nerd_jfpb

nerdjfpb

For is easy.
Just remember what we learned in our last tutorial.
For loop syntax is -

```
for (initial, condition, increment/
      decrement/rulesBreaking)
{
  // code block to be executed
}
```

nerdjfpb

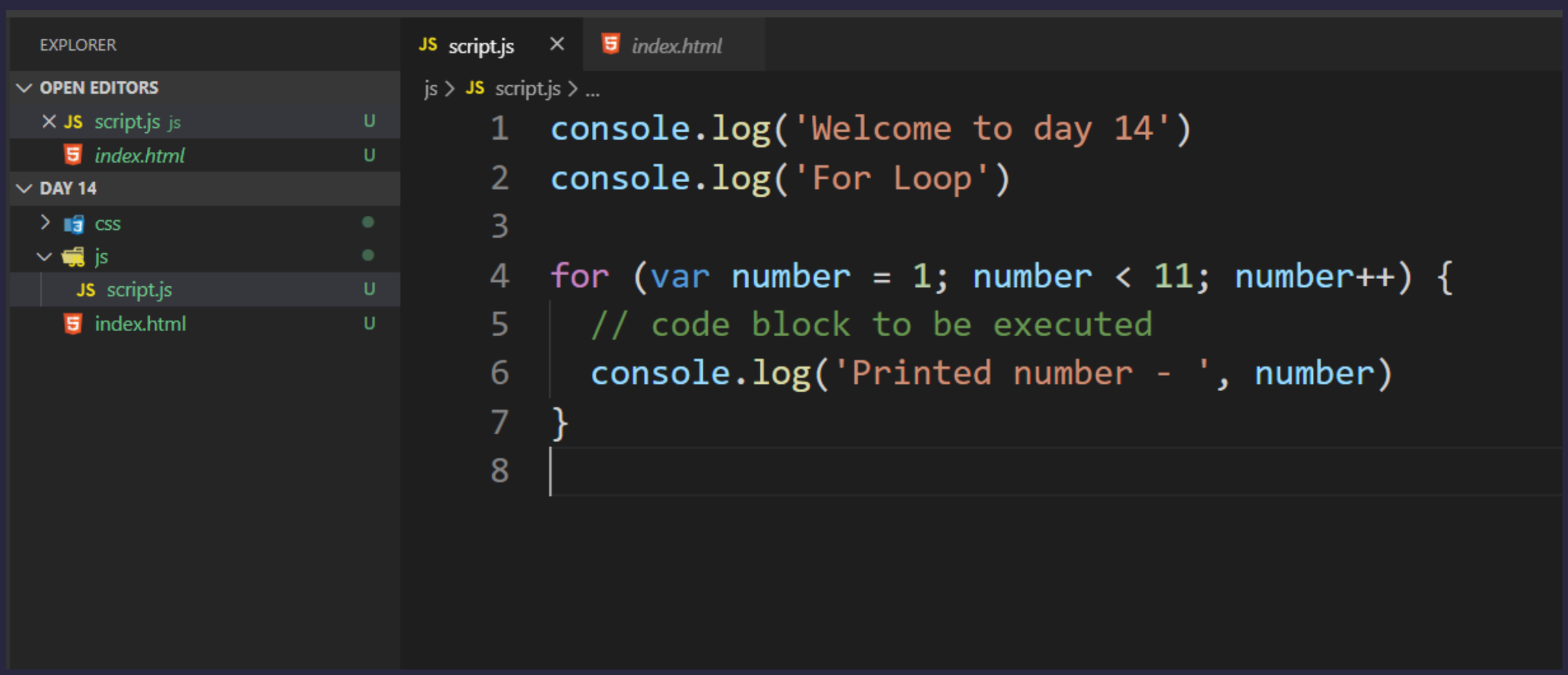
Let's write some real codes now. If we want to print 1 - 10 then we are going to store the values in variable and we are going to start from 1.

We can directly do this part in `for` first part!

nerdjfpb

Now we are going to start our for. For then first part will be initial of the number, then condition for breaking the loop and final part will be how we can break the loop.

```for
(var number = 1; number < 11; number++)
{
  // code block to be executed
  console.log('Printed number - ', number)
}
```



nerdjfpb

By the way did you notice that I used number++ ?
What that's mean ?
number++ = (number = number + 1)
same thing.
number++ is just a shortcut way to write it.

See the result in browser.

Welcome to JavaScript Tutorials!

Elements Console Sources Network Performance Memory Application Security Audits

top Filter Default levels

```
Welcome to day 14                                   script.js:1
For Loop                                            script.js:2
Printed number -  1                                 script.js:6
Printed number -  2                                 script.js:6
Printed number -  3                                 script.js:6
Printed number -  4                                 script.js:6
Printed number -  5                                 script.js:6
Printed number -  6                                 script.js:6
Printed number -  7                                 script.js:6
Printed number -  8                                 script.js:6
Printed number -  9                                 script.js:6
Printed number -  10                                script.js:6
>
```

Do you understand the For loop ?

nerdjfpb

nerdjfpb

JavaScript Series

For Vs While

nerd_jfpb

Part 15

nerdjfpb

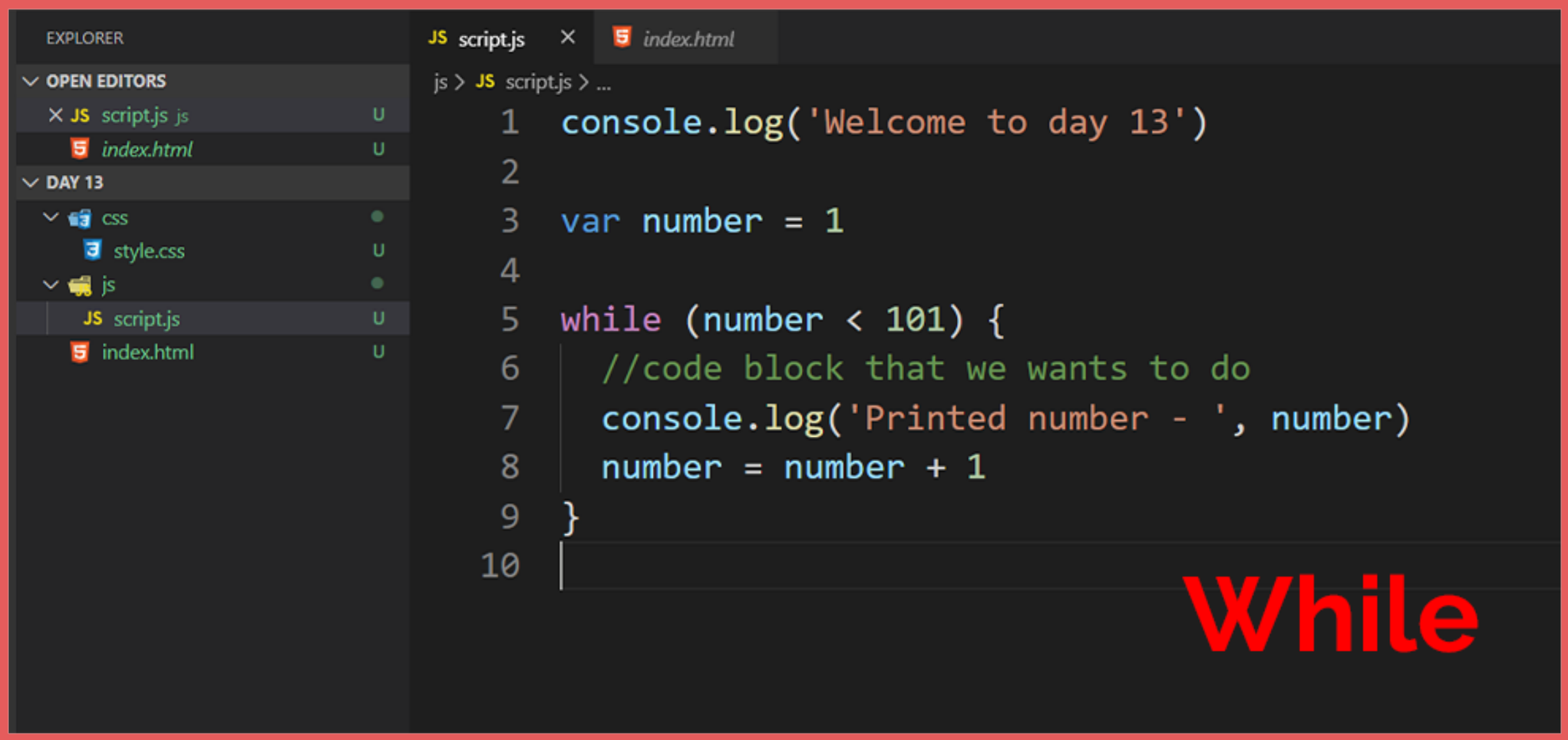
nerdjfpb

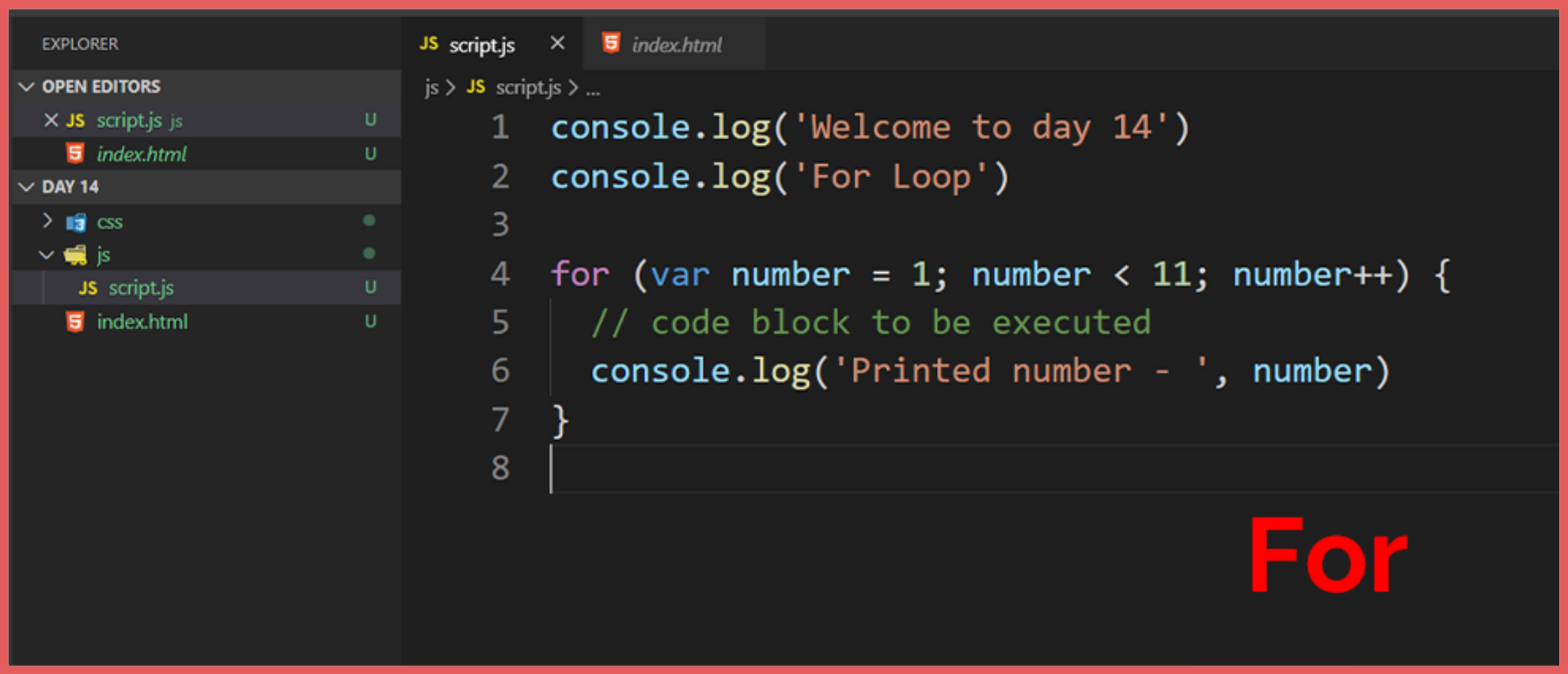
We have learned about both for and while.

So let's get a comperasion today!

nerd_jfpb

See both in one go.





nerdjfpb

In while Normally we have

```
while (condition) {
  // code block to be executed
}
```

nerdjfpb

In For Normally we have

```
for (initial, condition, increment
      decrement/rulesBreaking)
{
  // code block to be executed
}
```

nerd_jfpb

So basically both do the same work in different style right ?

Yaah that's true.
Now it's your choice which one you want to use!

nerdjfpb

For loop is more readable in my sense, so I use for loop more.

nerdjfpb

nerdjfpb

Which loop you like most and why ?

nerd_jfpb

nerdjfpb

nerdjfpb

Are you enjoying this series ?

nerd_jfpb

nerdjfpb

nerdjfpb

JavaScript Series

Functions

nerd_jfpb

Part 16

nerdjfpb

Function is a block of code where we write some tasks or a single task to do.

Function run only when we call it.

Sound confusing ?

nerdjfpb

If no then just skip next two slides/lines.

nerdjfpb

Do you remember using console.log?
What do you think this is ?
When we pass something on the "" in console.log then we can see it is printing on the console right ?
How this happens?

Actually there is some native parts of JavaScript.
Where there is function and we call it using it and pass the value to show in the console.
So we are already using the function right ?
Do you understand now ?

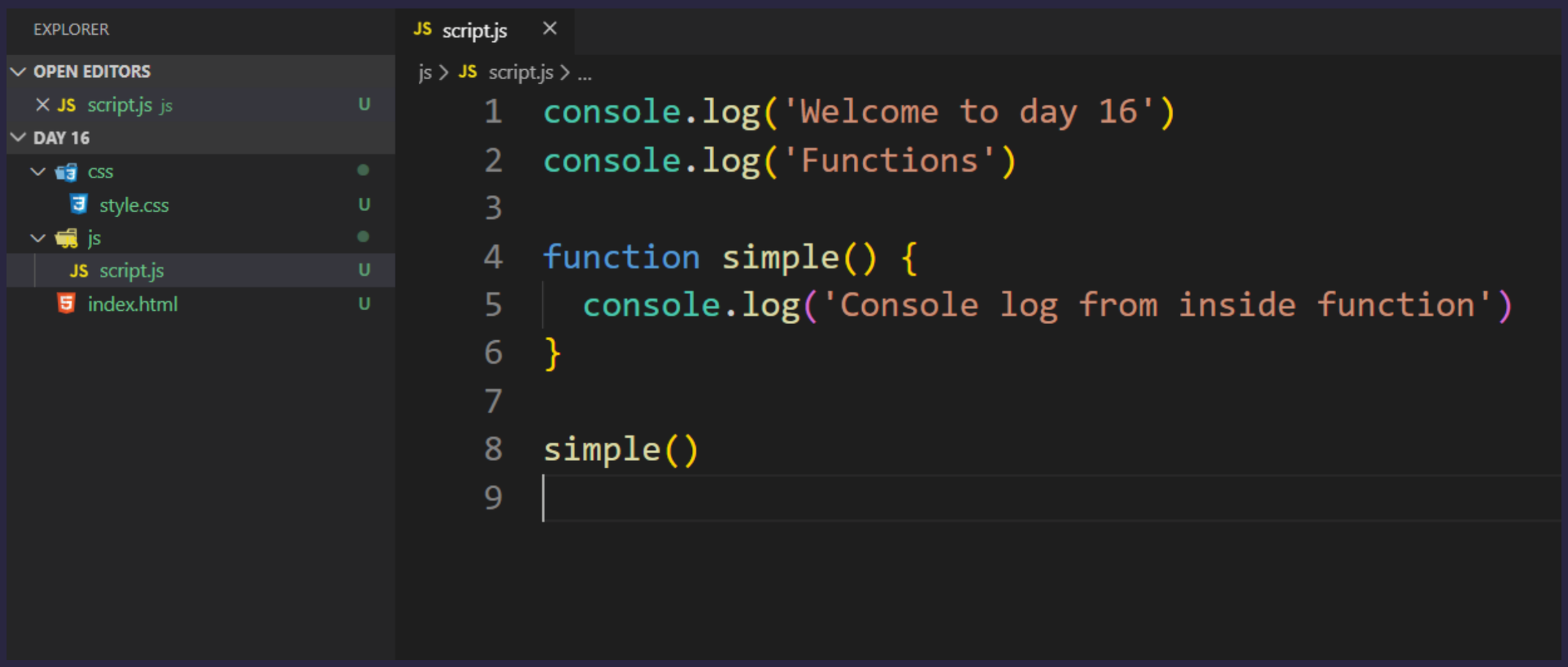
Now writing function in JavaScript is really easy.
Just start with function keyword.
This is reserve keyword for initialing a function like we initial variable using var.

nerdjfpb

It is like giving name of variable.
We'll give a name after function keyword.
Then function need some parameter to pass.
We can leave it blank for now and our code will be -

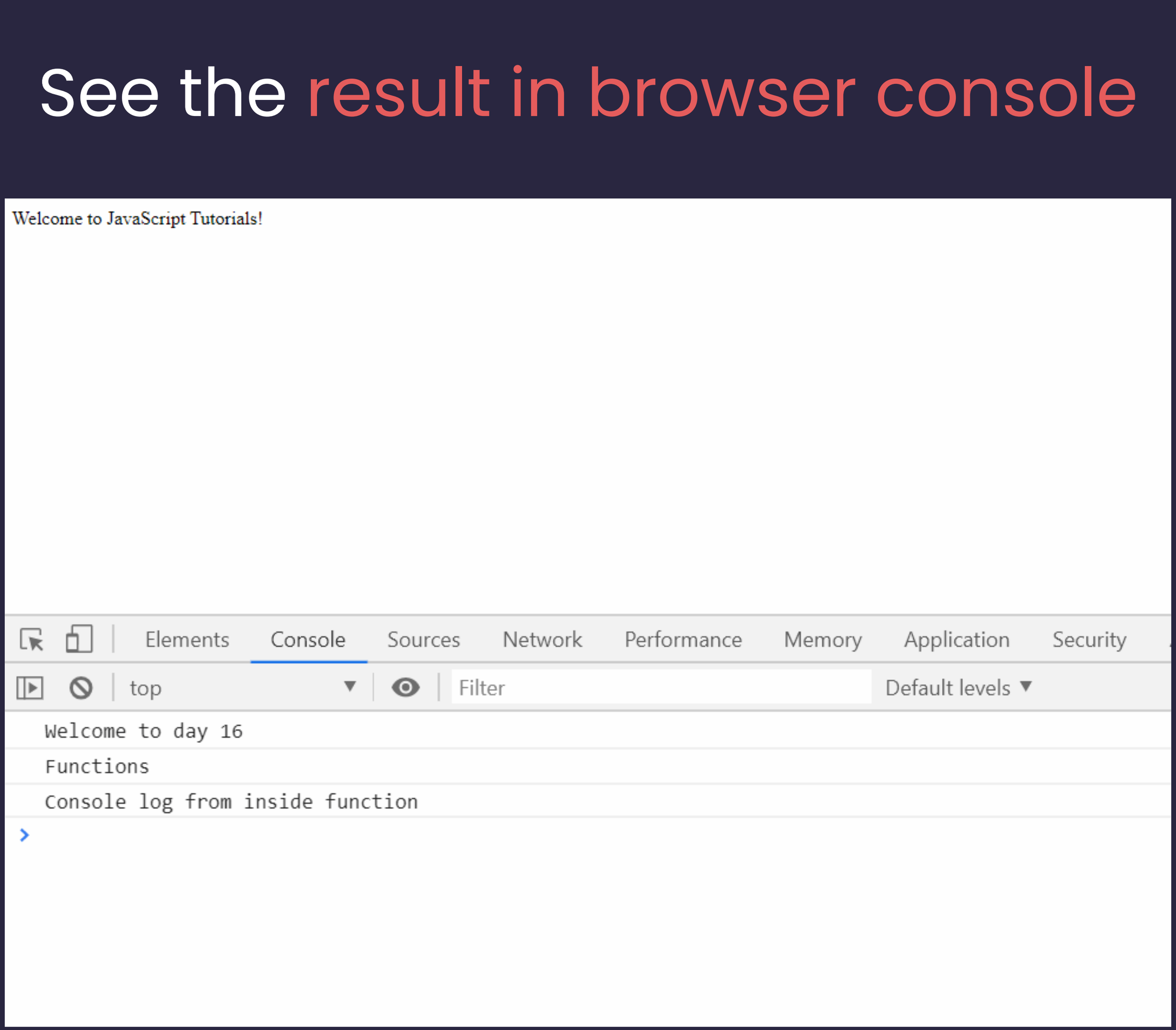
```
function nameOfFunction() {
//Block of code
}
```

Now lets write a function -



See the result in browser console

Welcome to JavaScript Tutorials!



Can you write some basic functions now ?

nerdjfpb

nerdjfpb

JavaScript Series

More About Functions

nerd_jfpb

Part 15

nerdjfpb

In last part of function we learn about a basic function and we just console.log inside the function but today we are going to learn more about it. We'll learn how we can add two numbers using a function!

nerdjfpb

In last part we didn't use any parameters right?

Now we'll use some parameters.

We'll pass the parameters inside of "()".
We can pass variables here, even function too.
But for now just think of variable.

Today we're going to create a plus function which will give us the result of two number.

nerdjfpb

First we're going to write function keyword and the name of function and we'll pass two numbers.

```
function plus(number1,number2){
  console.log(number1)
  console.log(number2)
}
plus(10,12)
```

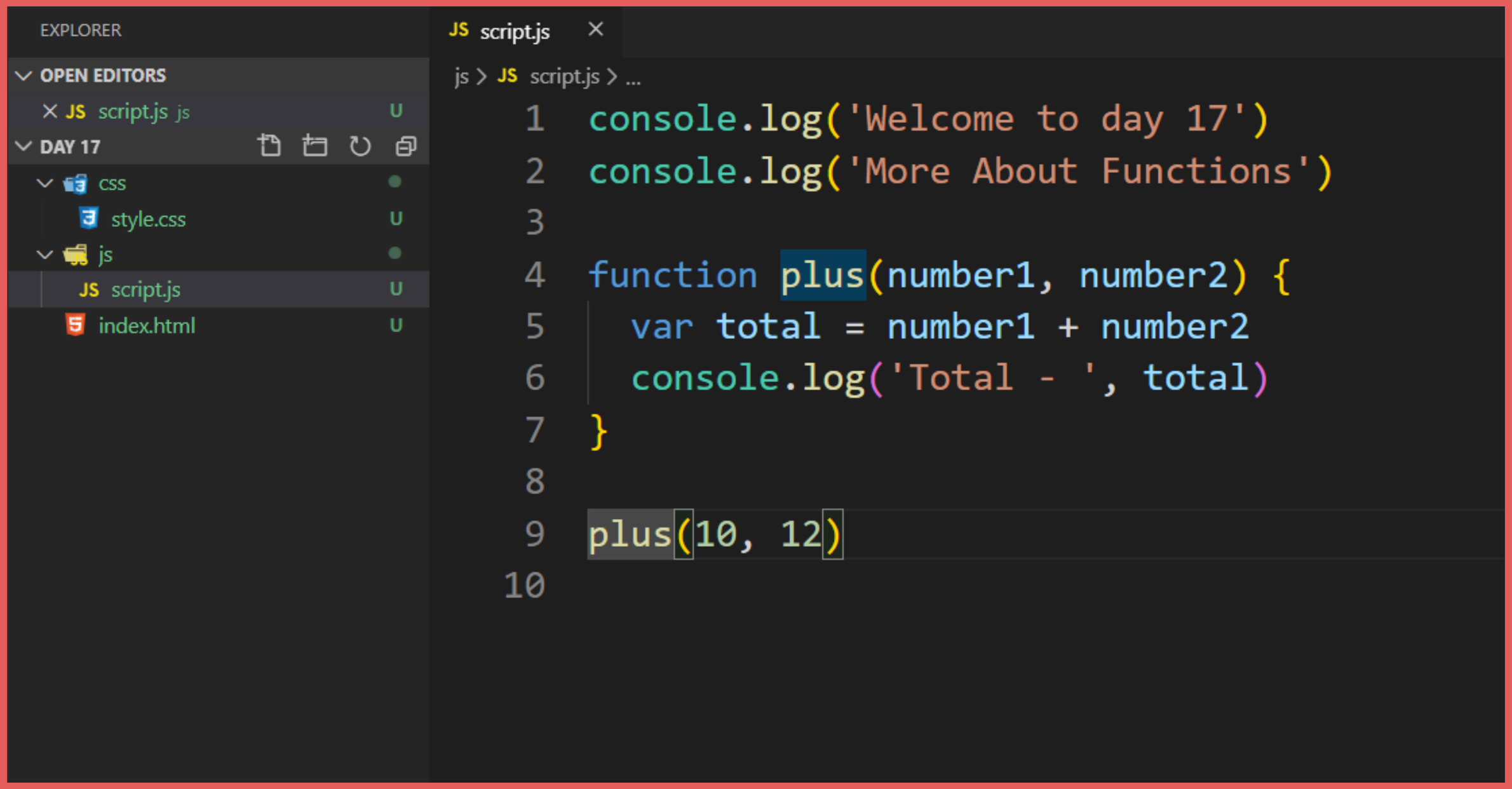
nerdjfpb

We can easily access the values from the parameters.
Now we can store both in a variable like -

```
function plus(number1,number2){
  var total = number1 + number2
}
```

nerd_jfpb

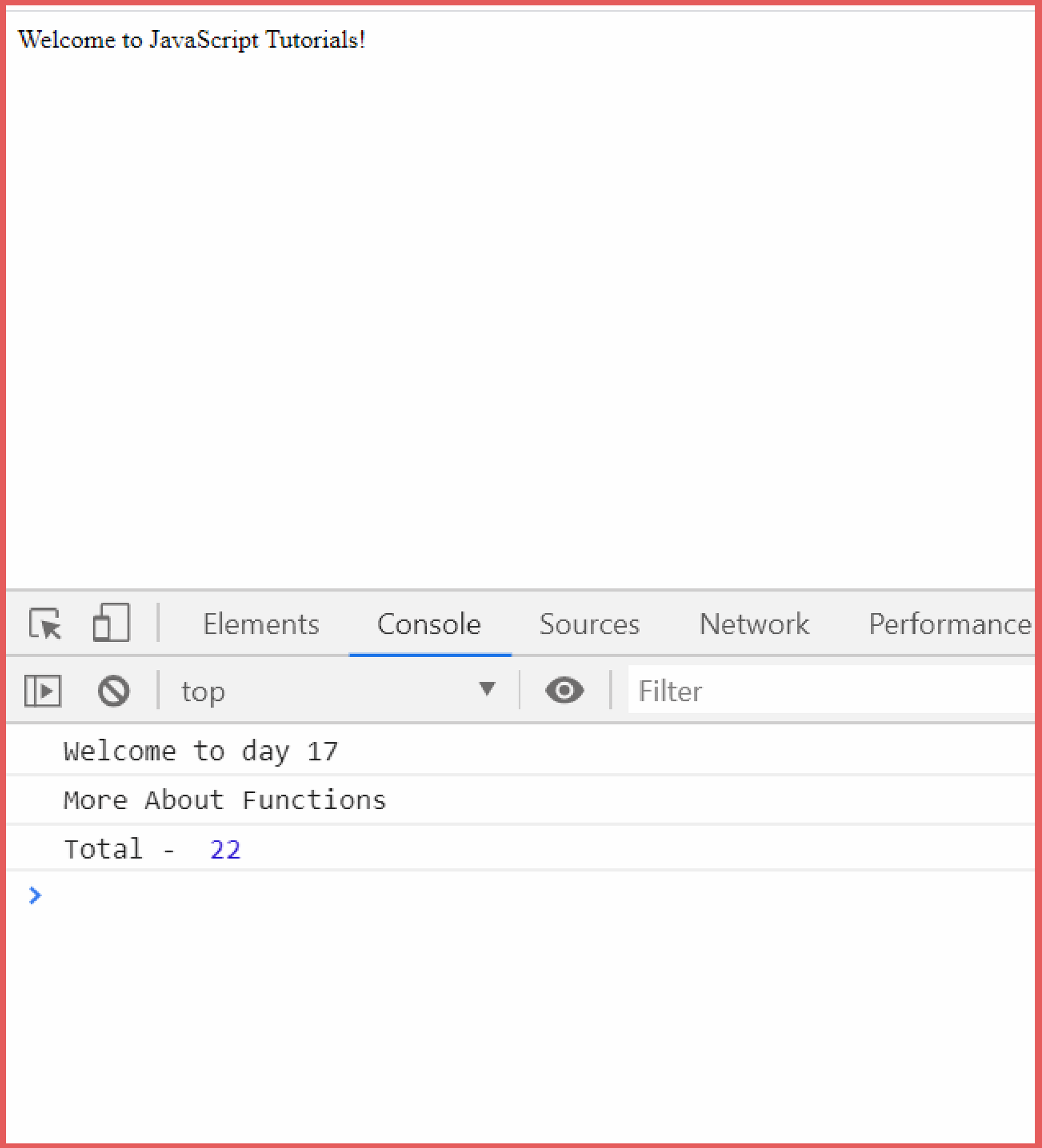
Now we can console log it on our console.



nerdjfpb

See the result

Welcome to JavaScript Tutorials!



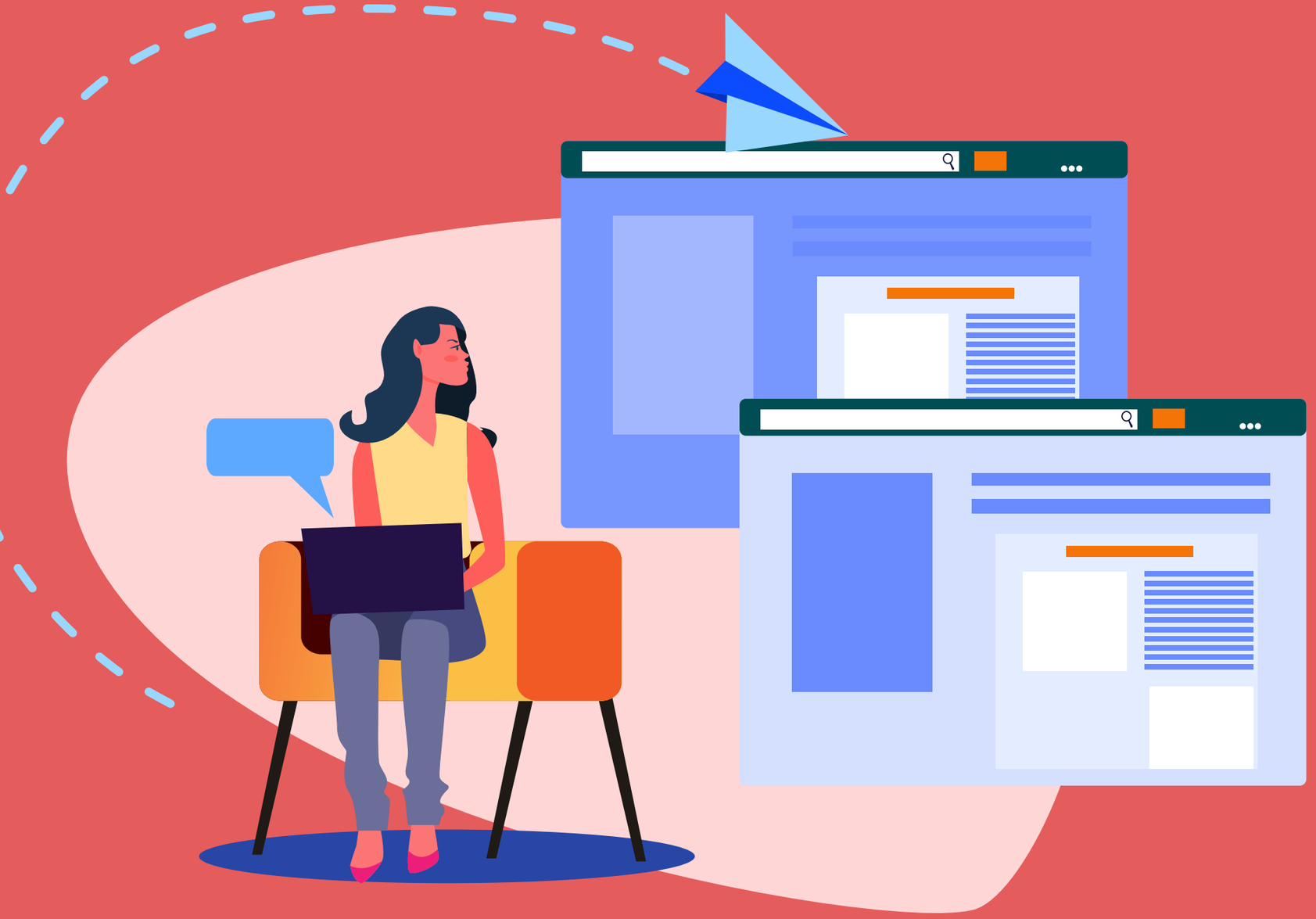
Can you write the minus function now ?

nerdjfpb

nerdjfpb

JavaScript Series

Different types of function



nerd_jfpb

Part 18

nerdjfpb

In last two part we declare the function in a same style, but there is another way declare a variable.

We are going to learn it today.

nerdjfpb

Up until now we wrote the functions declaration.

Now we are going to learn about function expression.

nerdjfpb

Here we don't have a name for the function.

So it also called anonymous function.

nerdjfpb

nerdjfpb

Let's write one to understand more about it.

nerd_jfpb

```
''
function() {
  console.log('How you doin ?')
}
```

this will be function and we'll store it on a variable and then we'll call the function using variable()

Example-

```
10
11  //function expression
12  const joeyTalking = function() {
13      console.log('How you doin ?')
14  }
15
16  joeyTalking()
17
```

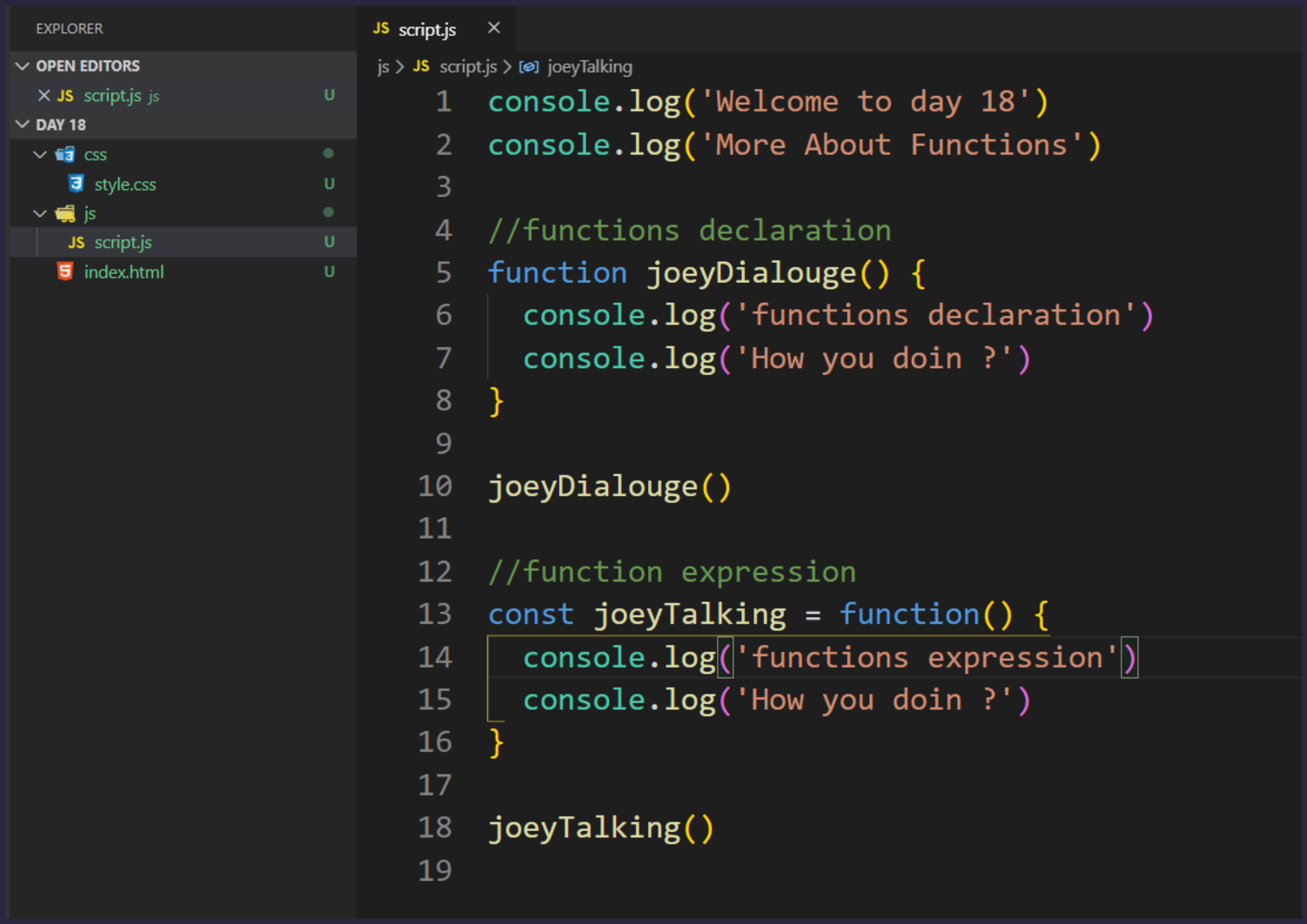
nerdjfpb

This exactly does the same thing as our old function.

This pattern is widely use in js, so it is better to know both.

nerdjfpb

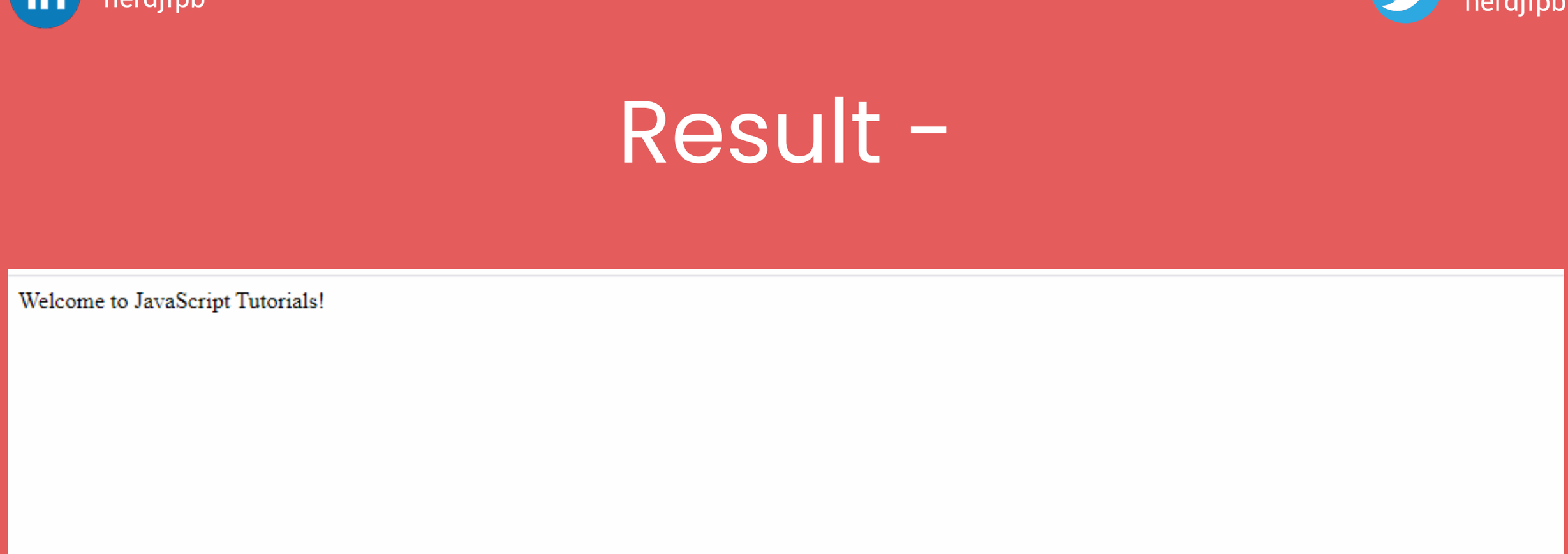
Let's have a look on the both type



nerdjfpb

Result -

Welcome to JavaScript Tutorials!



nerdjfpb

nerdjfpb

JavaScript Series

Return in functions

nerd_jfpb

Part 19

Until now we didn't worked with the return in functions.

But there is a return in function.

nerdjfpb

If we don't write anything in return then it just return undefined.

But we can return anything.
Like - integer, string etc.

nerdjfpb

nerdjfpb

Let's make our hand dirty by coding.

nerd_jfpb

Suppose in function we are going to two parameters.
One is name and another is dialouge.
Now we are going to print it on the console using return within a function.

Now get the values from params first and put it inside a new variable. Now we are going to return this variable.

```
function dialouge(name, dialouge) {
  const nameWithDialouge = name + ' ' + dialouge
  return nameWithDialouge
}
```

Now we need to change our function call.
Because this function is not doing anything.
It is returning something,
we are going to catch the value and print it.

This step is a pretty easy to. Just initial a variable by calling the function.

```
1  console.log('Welcome to day 19')
2  console.log('Return in functions')
3
4  function dialouge(name, dialouge) {
5    const nameWithDialouge = name + ' ' + dialouge
6    return nameWithDialouge
7  }
8
9  const dialougeForPrint = dialouge('Joey', 'How you doing?')
10 console.log(dialougeForPrint)
11
```

See the result in browser -

Welcome to JavaScript Tutorials!

Elements Console Sources Network Performance Memory Applicat

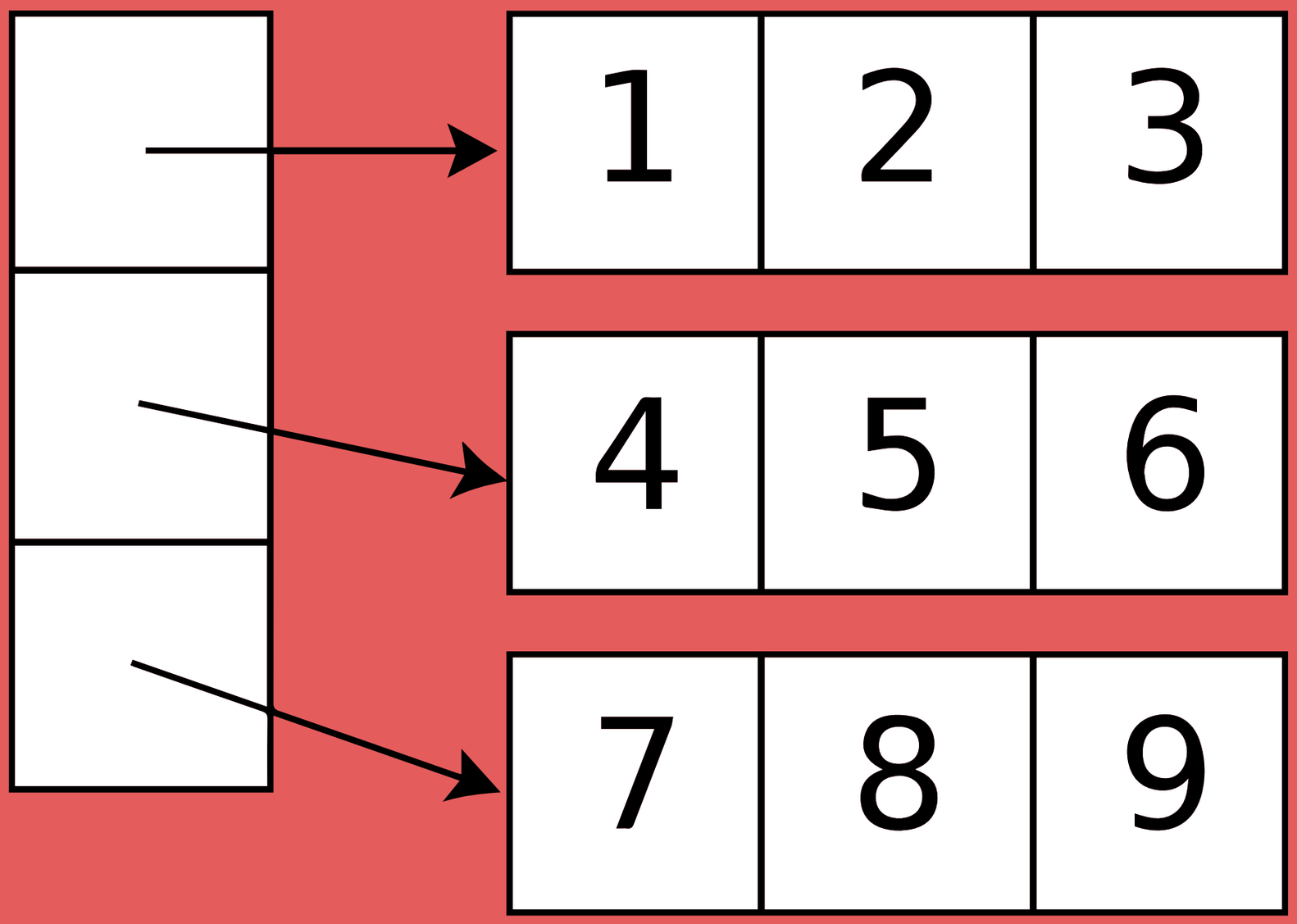
top Filter Default lev

```
Welcome to day 19
Return in functions
Joey How you doing?
>
```

Can you write some functions now ?

JavaScript Series

Array



We know about variable right?
Now we are going to learn about data structures.

nerdjfpb

In computer science, a data structure is a data organization, management and storage format that enables efficient access and modification.
More precisely, a data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data.

[wikipedia]

nerdjfpb

I think it's easy enough to understand.
Up until now we were just using the variable to store a single data, now we are going to learn about array where we can store multiple data.

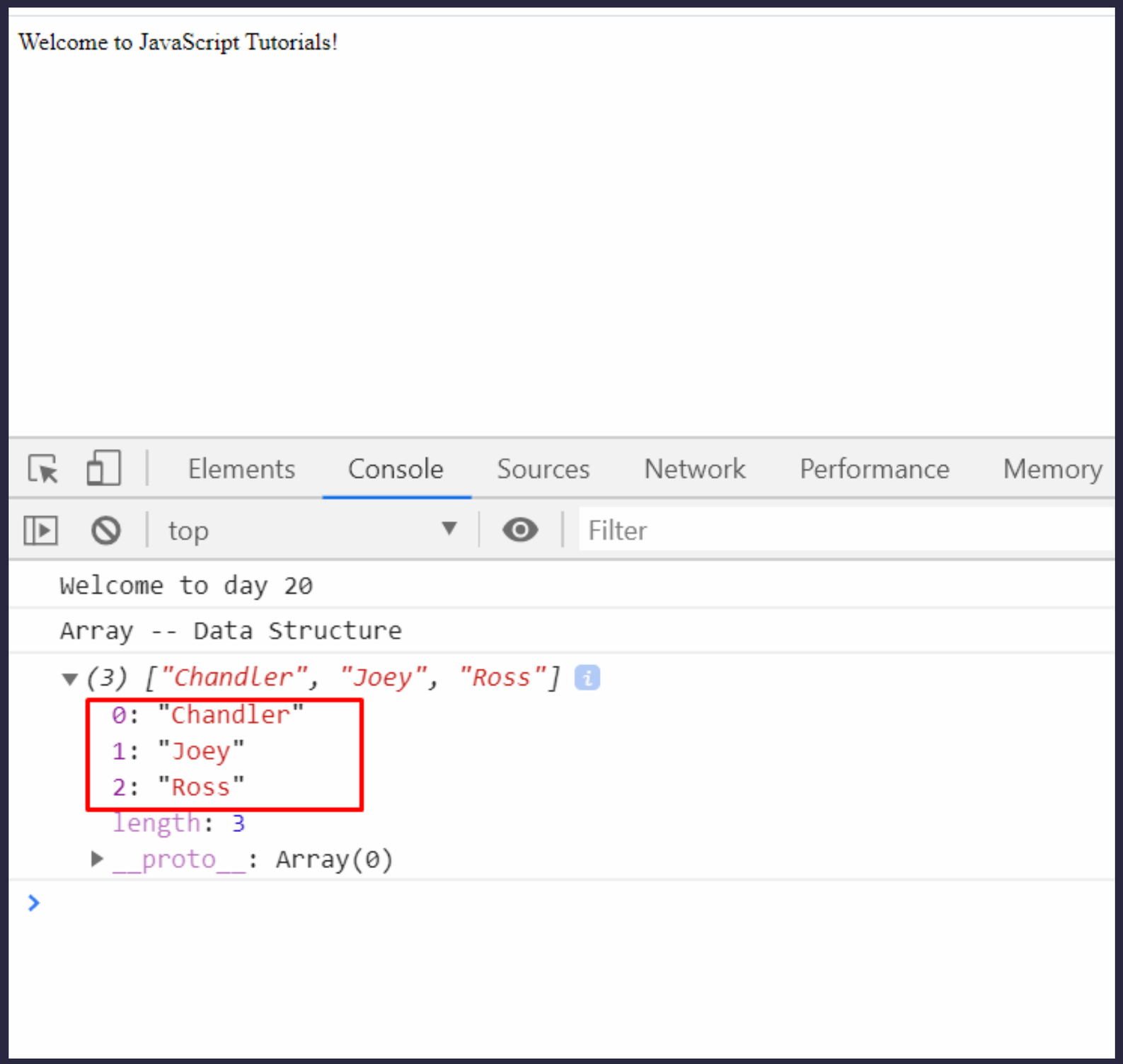
It's easy.
we need to initial a variable and
put some values in there.

See this is so easy-

```
js > JS script.js > ...
1  console.log('Welcome to day 20')
2  console.log('Array -- Data Structure')
3
4  var friendsCharacters = ['Chandler', 'Joey', 'Ross']
5  console.log(friendsCharacters)
6
```

nerdjfpb

In the browser we can see some interesting things about the array. There is some indexing here of the each elements.



nerd_jfpb

nerdjfpb

Array indexing count from one.
If we want, we can just console log
`Chandler` using array -

js > JS script.js > ...

```
1  console.log('Welcome to day 20')
2  console.log('Array -- Data Structure')
3
4  var friendsCharacters = ['Chandler', 'Joey', 'Ross']
5  console.log(friendsCharacters[0])
6
```

nerdjfpb

If you give a value which is more than index value then it will give you undefined -

JS script.js > ...

```
1  console.log('Welcome to day 20')
2  console.log('Array -- Data Structure')
3
4  var friendsCharacters = ['Chandler', 'Joey', 'Ross']
5  console.log(friendsCharacters[4])
6
```

nerd_jfpb

Do you understand the Array ?

nerdjfpb

nerdjfpb

JavaScript Series

More About Array

nerd_jfpb

Part 21

In last part we learned about array, in this part we are going to go a little deep around arary.

nerdjfpb

Suppose we've an array of favAnimeList = ['One piece', 'Dr Stone', 'Haikyuu', 'Attack on Titan']

Now we want to delete the last one.

How we can do this ?

Using the pop.

```
js > JS script.js > ...
1  console.log('Welcome to day 21')
2  console.log('More Array')
3
4  var favAnimeList = ['One piece', 'Dr Stone', 'Haikyuu', 'Attack on Titan']
5  favAnimeList.pop()
6  console.log(favAnimeList)
7
```

Now we want to add "naruto" in the array.
How we can do this ?

Just using push.
This will add the value in end.

```
JS script.js
js > JS script.js > ...
1  console.log('Welcome to day 21')
2  console.log('More Array')
3
4  var favAnimeList = ['One piece', 'Dr Stone', 'Haikyuu', 'Attack on Titan']
5  //delete the last one from array
6  favAnimeList.pop()
7  console.log(favAnimeList)
8  //add a new item in array in last position
9  favAnimeList.push('Naruto')
10 console.log(favAnimeList)
11
```

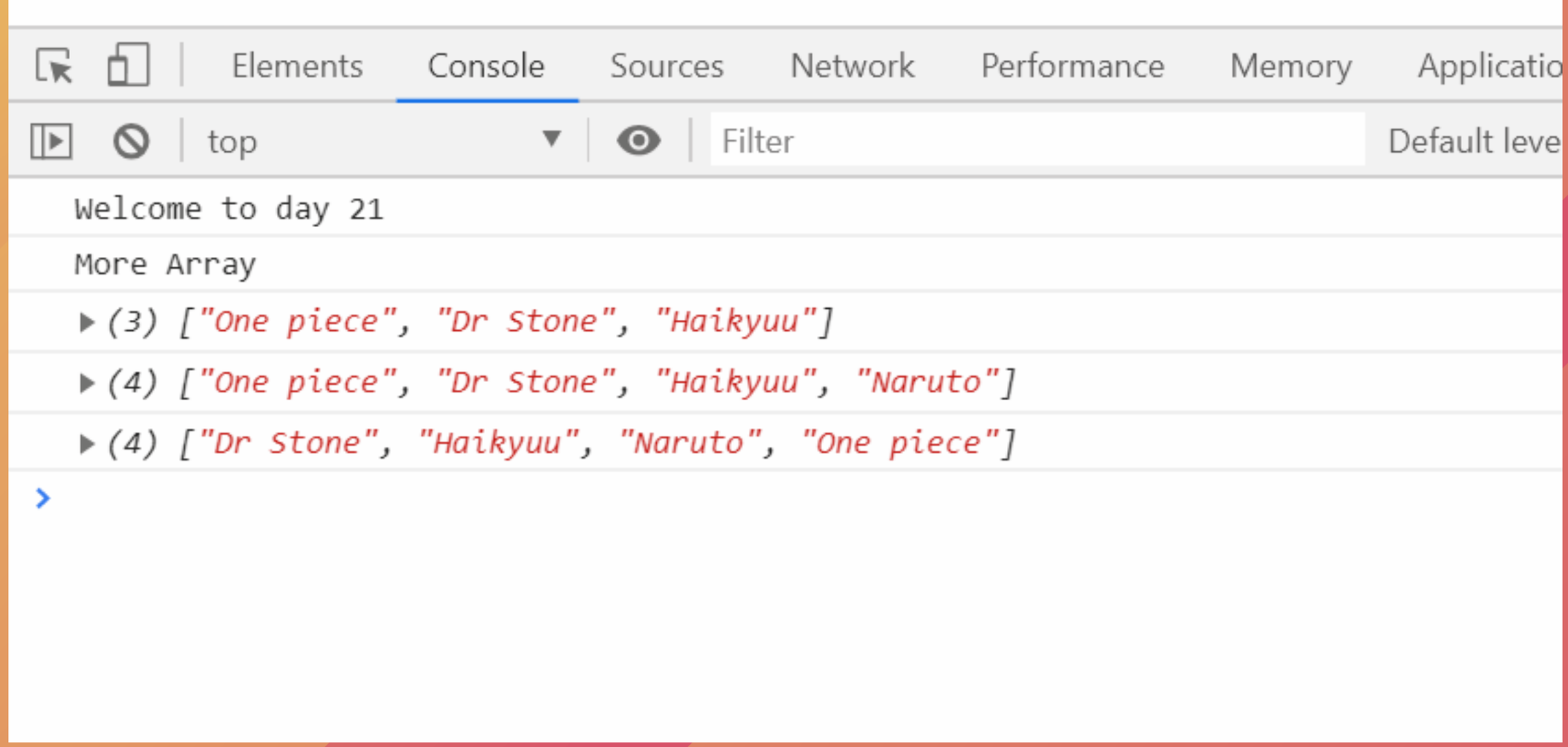
Now we can easily sort the array using sort.

script.js

```
1  console.log('Welcome to day 21')
2  console.log('More Array')
3
4  var favAnimeList = ['One piece', 'Dr Stone', 'Haikyuu', 'Attack on Titan']
5  //delete the last one from array
6  favAnimeList.pop()
7  console.log(favAnimeList)
8  //add a new item in array in last position
9  favAnimeList.push('Naruto')
10 console.log(favAnimeList)
11 //sort the array
12 favAnimeList.sort()
13 console.log(favAnimeList)
14
```

See all results on the browser

Welcome to JavaScript Tutorials!



nerdjfpb

nerdjfpb

Can you use array now ?

nerd_jfpb

nerdjfpb

nerdjfpb

JavaScript Series

Object

nerd_jfpb

Part 22

Today we are going to learn about a new javascript type. This is called `Object`

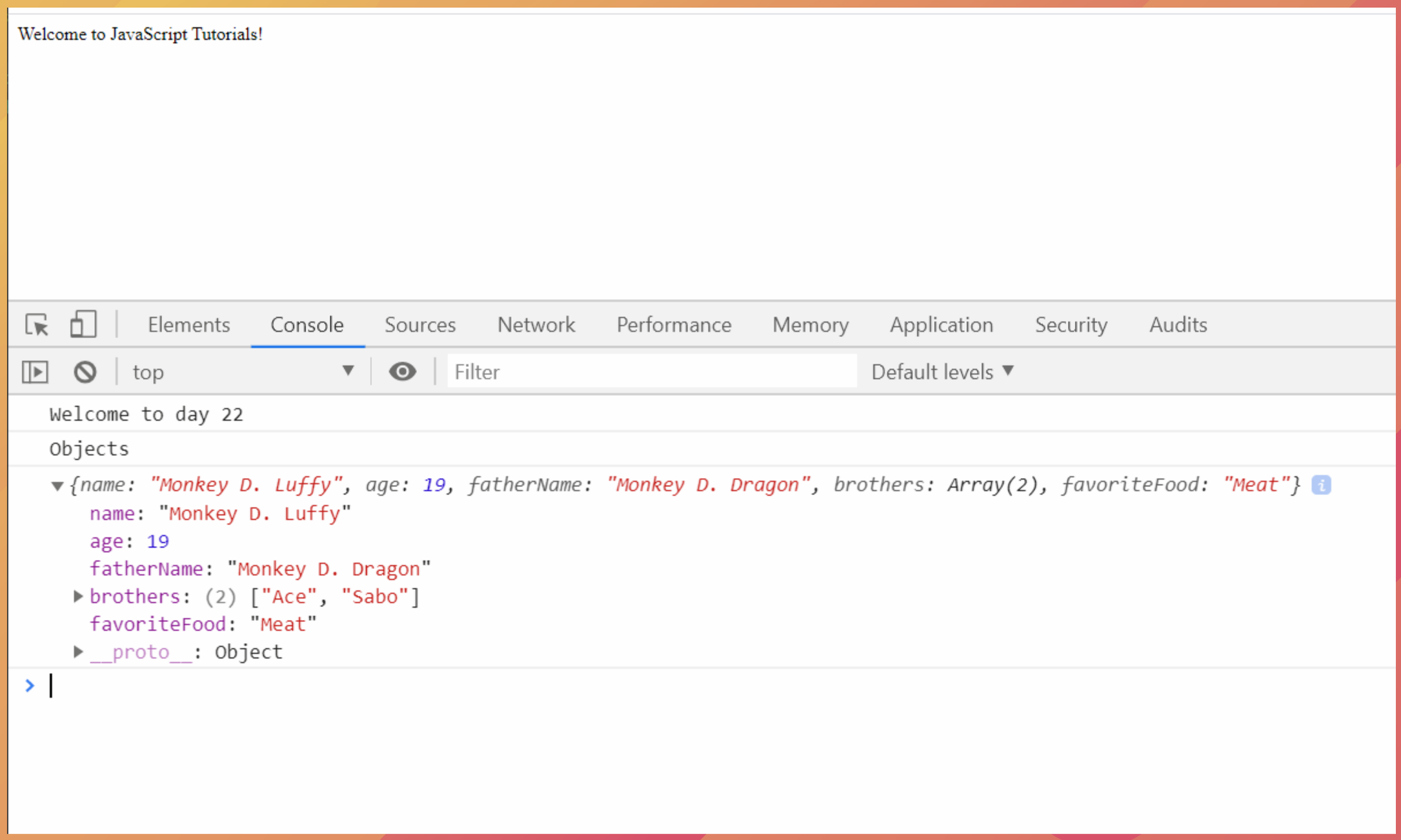
In object type we can store multiple types of data.
Like - numbers, string, arrays etc.

JS script.js

js > JS script.js > ...

```
1  console.log('Welcome to day 22')
2  console.log('Objects')
3
4  var characterDetails = {
5    name: 'Monkey D. Luffy',
6    age: 19,
7    fatherName: 'Monkey D. Dragon',
8    brothers: ['Ace', 'Sabo'],
9    favoriteFood: 'Meat'
10 }
11
```

See what we got in the `characterDetails`



We can easily change the values of the object.

```
1  console.log('Welcome to day 22')
2  console.log('Objects')
3
4  var characterDetails = {
5    name: 'Monkey D. Luffy',
6    age: 19,
7    fatherName: 'Monkey D. Dragon',
8    brothers: ['Ace', 'Sabo'],
9    favoriteFood: 'Meat'
10 }
11
12 characterDetails.age = 20
13
14 console.log(characterDetails)
15
```

Do you know we can write function inside of object ?

like -

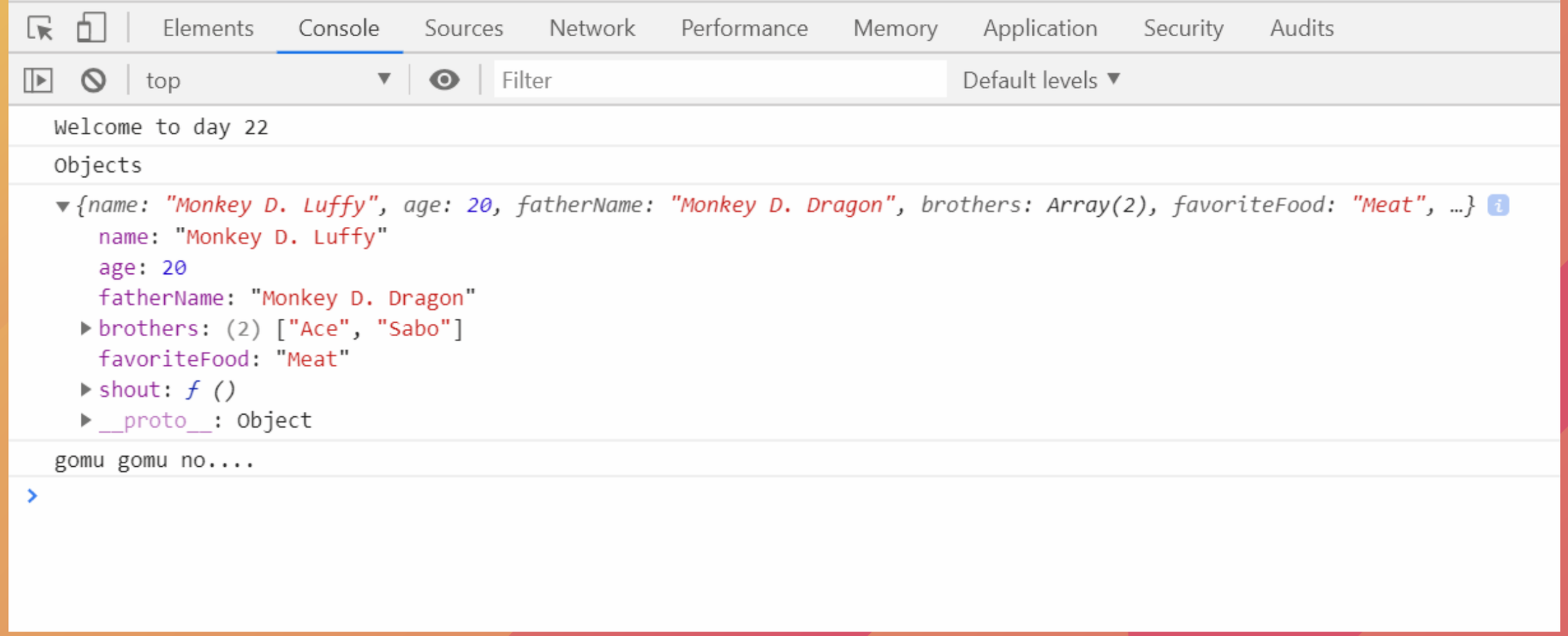
JS script.js ×

js > JS script.js > ...

```
1  console.log('Welcome to day 22')
2  console.log('Objects')
3
4  var characterDetails = {
5    name: 'Monkey D. Luffy',
6    age: 19,
7    fatherName: 'Monkey D. Dragon',
8    brothers: ['Ace', 'Sabo'],
9    favoriteFood: 'Meat',
10   shout: function() {
11     console.log('gomu gomu no....')
12   }
13 }
14
15 characterDetails.age = 20
16
17 console.log(characterDetails)
18
19 characterDetails.shout()
20
```

See the result

Welcome to JavaScript Tutorials!



nerdjfpb

Can you initial a object now ?

nerdjfpb

nerdjfpb

Are you enjoying this tutorial ?

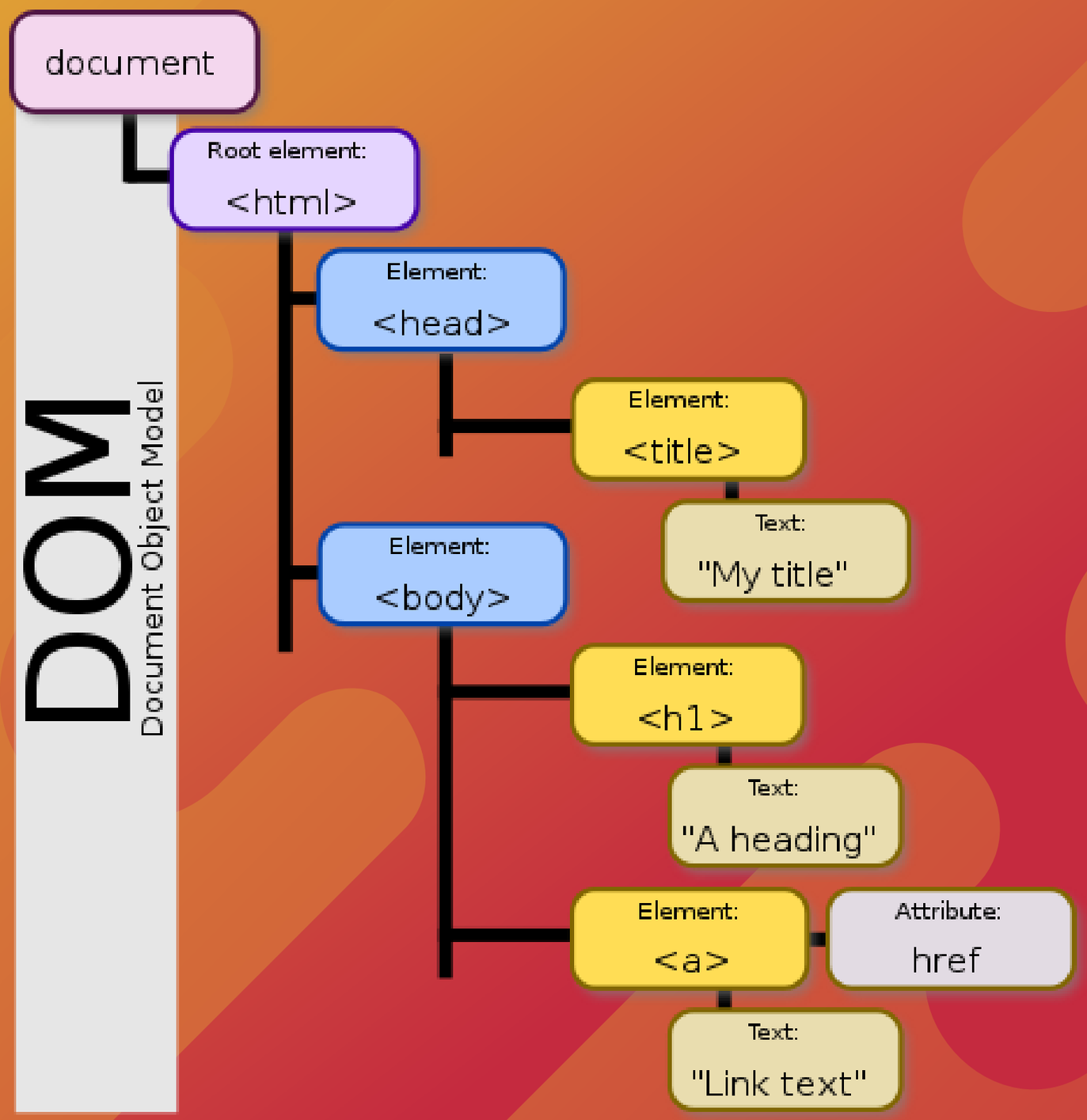
nerd_jfpb

nerdjfpb

nerdjfpb

JavaScript Series

Dom Manipulation



nerd_jfpb

Part 23

nerdjfpb

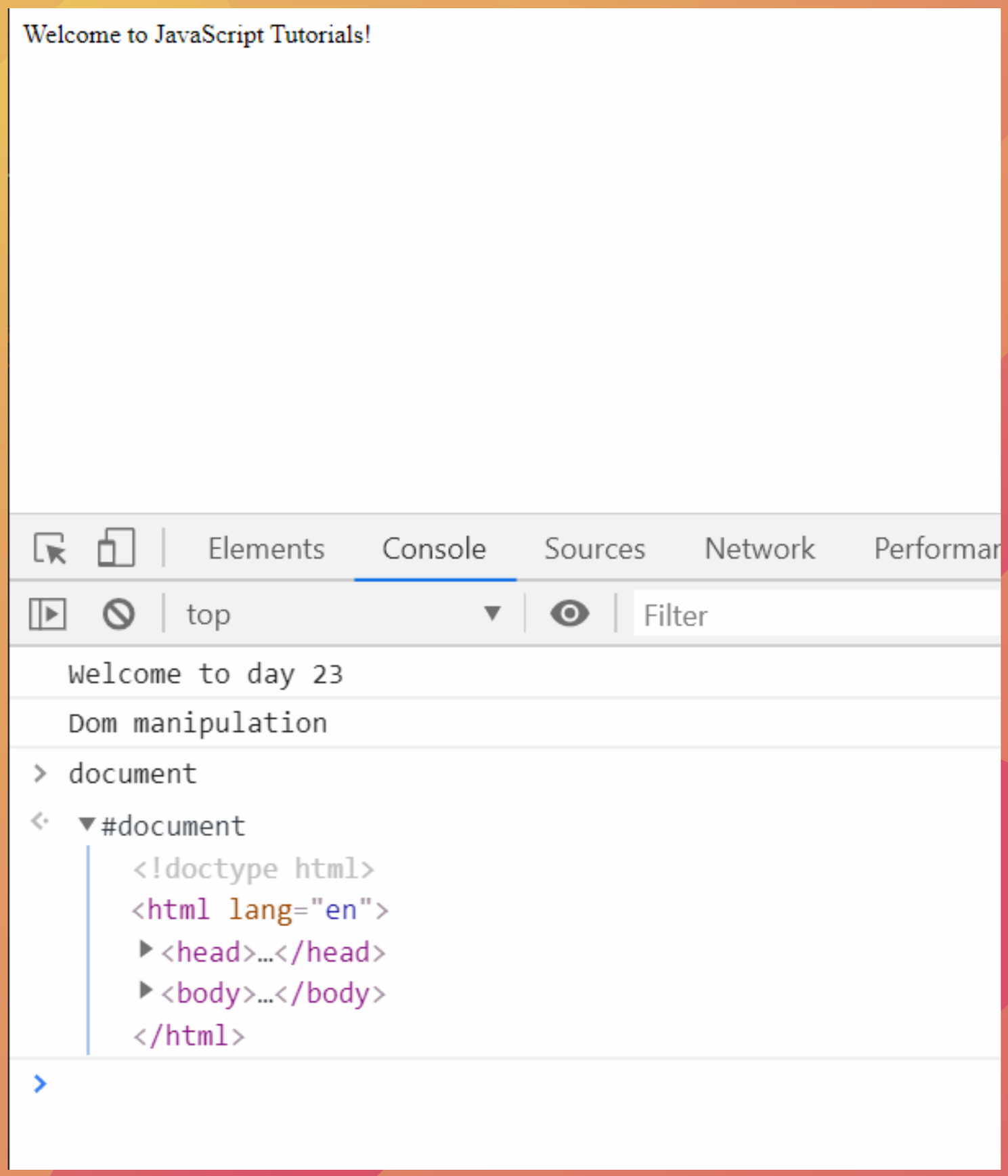
nerdjfpb

We've ended the javascript basics, what next ?

We are going to learn about the javascript dom manipulation.

nerd_jfpb

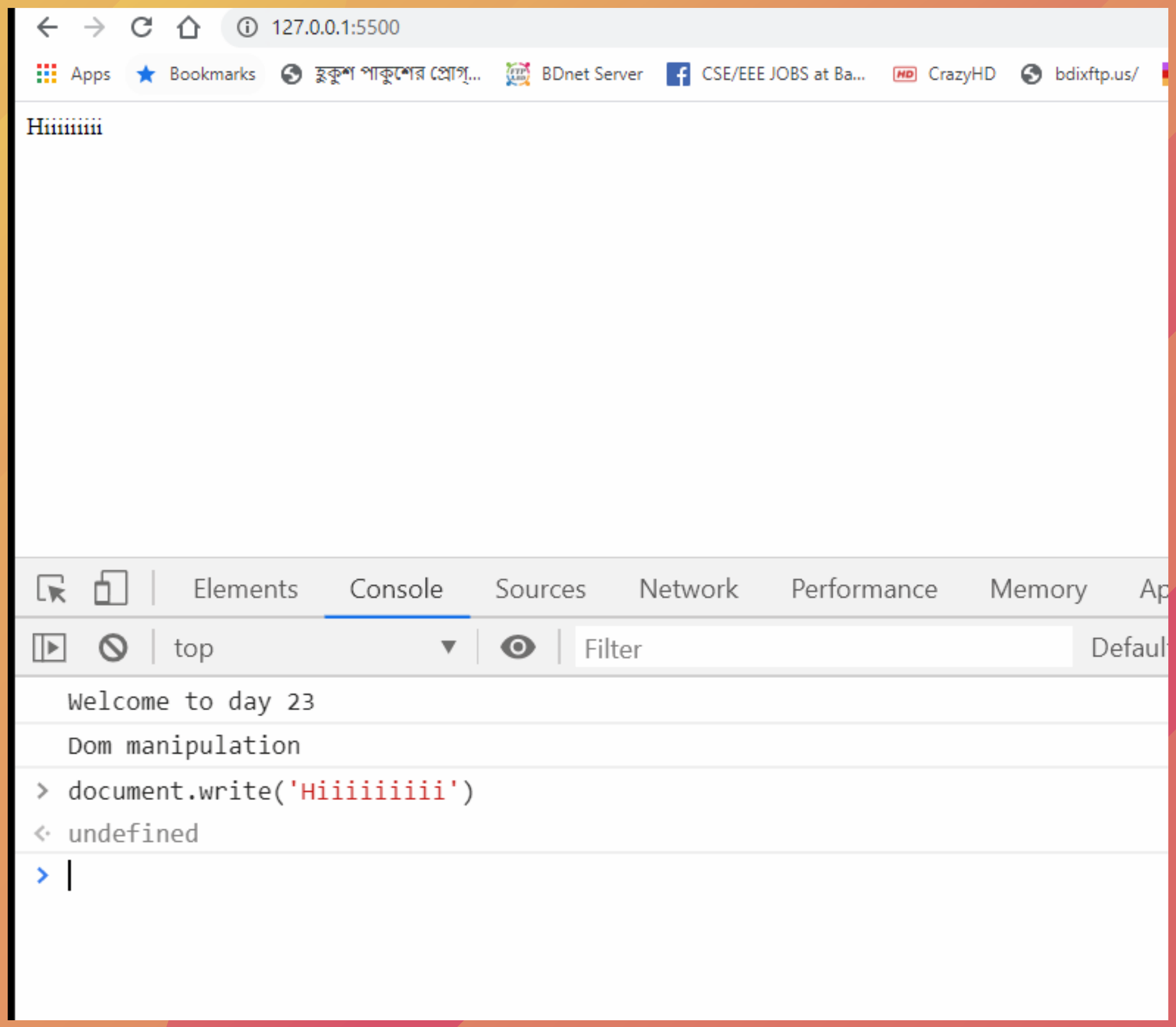
Let's go to console in write document.



We can see the whole html we wrote right ?

Now we can manipulate this with our javascript code.
Sounds fun ?

Let's try to write `document.write('Hiiiiiiiii')` in console and see the magic!



So where the document is coming from ?

Document is a global object which is available on the browser.

But it has a parent, can you guess ?

write `window` in the console and see what happens -

Hiiiiiiiii

Elements Console Sources Network Performance Memory Application Security Audits

top Filter Default levels

```
Welcome to day 23
Dom manipulation
> document.write('Hiiiiiiiii')
undefined
> window
▼Window {parent: Window, opener: null, top: Window, length: 0, frames: Window, …}
  ▶document: document
    name: ""
  ▶customElements: CustomElementRegistry {}
  ▶history: History {length: 2, scrollRestoration: "auto", state: null}
  ▶locationbar: BarProp {visible: true}
  ▶menubar: BarProp {visible: true}
  ▶personalbar: BarProp {visible: true}
  ▶scrollbars: BarProp {visible: true}
  ▶statusbar: BarProp {visible: true}
  ▶toolbar: BarProp {visible: true}
```

nerdjfpb

nerdjfpb

Interested to make yourself into a magician who can change the dom ?

nerd_jfpb

nerdjfpb

nerdjfpb

JAVASCRIPT SERIES-24

DOM SELECTOR

nerd_jfpb

nerdjfpb

In last tutorial we started the magic dom manipulation. Today we are going to learn about the dom selector so that we can select a dom part and change it as we want.

Let's start.

I made a new html for us to edit. Have a look, code available in github.

index.html

index.html > html > body > h1.main-heading

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>JavaScript Tutorial</title>
    <link rel="stylesheet" href="./css/style.css" />
  </head>
  <body>
    <h1 class="main-heading">Welcome to this otaku kingdom</h1>
    <p class="sub-heading">You can find different types of anime here.</p>
    <ul class="anime-list">
      <li>One piece</li>
      <li>Naruto</li>
      <li>One Punch Man</li>
      <li>Dr. Stone</li>
      <li>Haikyuu</li>
      <li>Boku no hero academia</li>
    </ul>
    <script src="./js/script.js"></script>
  </body>
</html>
```

Now go to console and type document. Get it will bring the autocomplete other options.

Welcome to this otaku kingdom

You can find different types of anime here.

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia

Elements Console Sources Network Performance Memory Applicati

top Filter Default lev

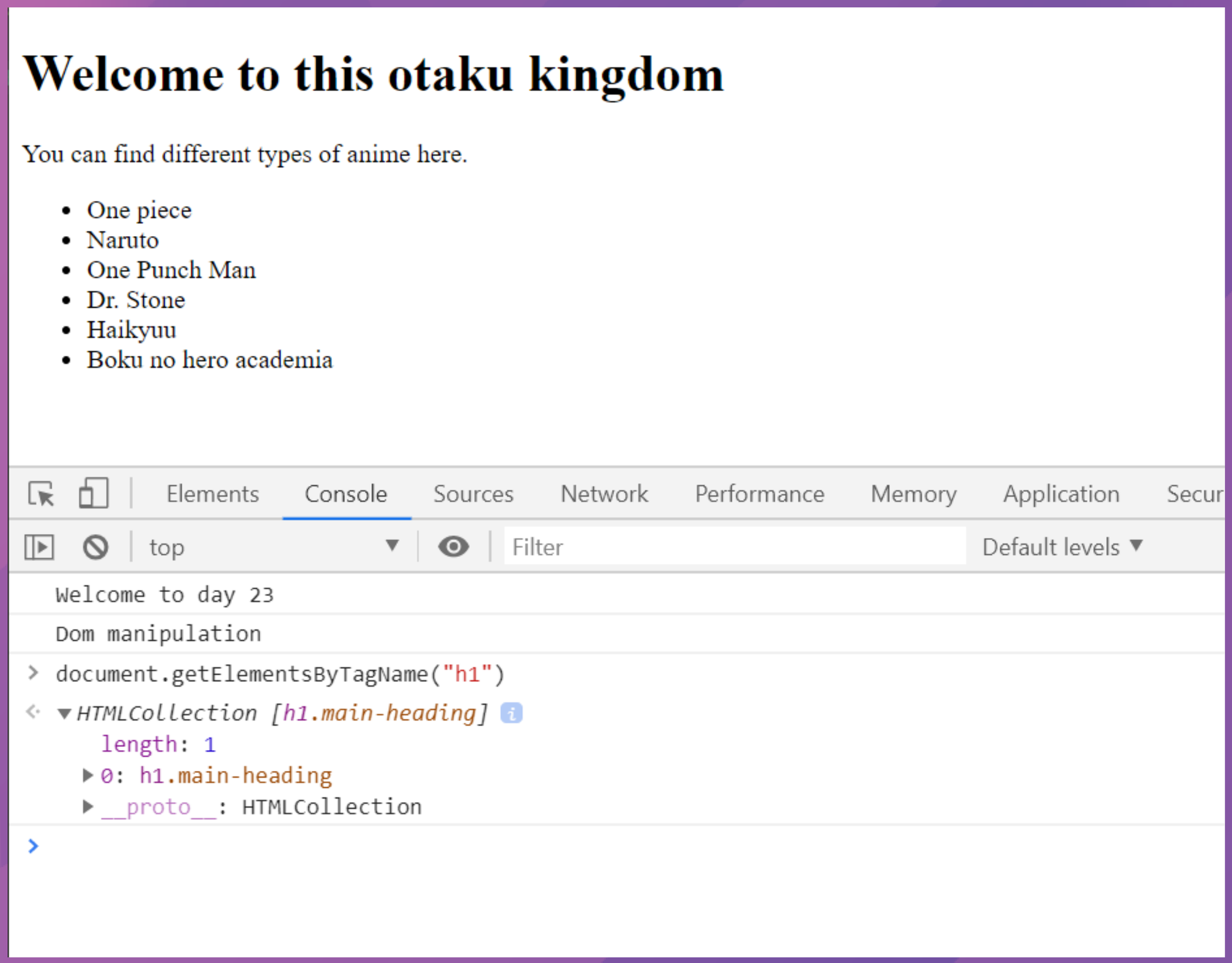
```
Welcome to day 23
Dom manipulation
> document.getElementById
ƒ
```

getElementById Document
getElementsByClassName
getElementsByName
getElementsByTagName
getElementsByTagNameNS
getSelection
getRootNode Node
__defineGetter__ Object
__lookupGetter__

nerdjfpb

So there is lot of different selector write?
We can use them to select something and edit that.
We're going to start from the tag select.

Now by just writing
`document.getElementsByTagName("h1")`
we can grab the h1. see



We got and HTMLCollection and in 0 we can see the classname also.

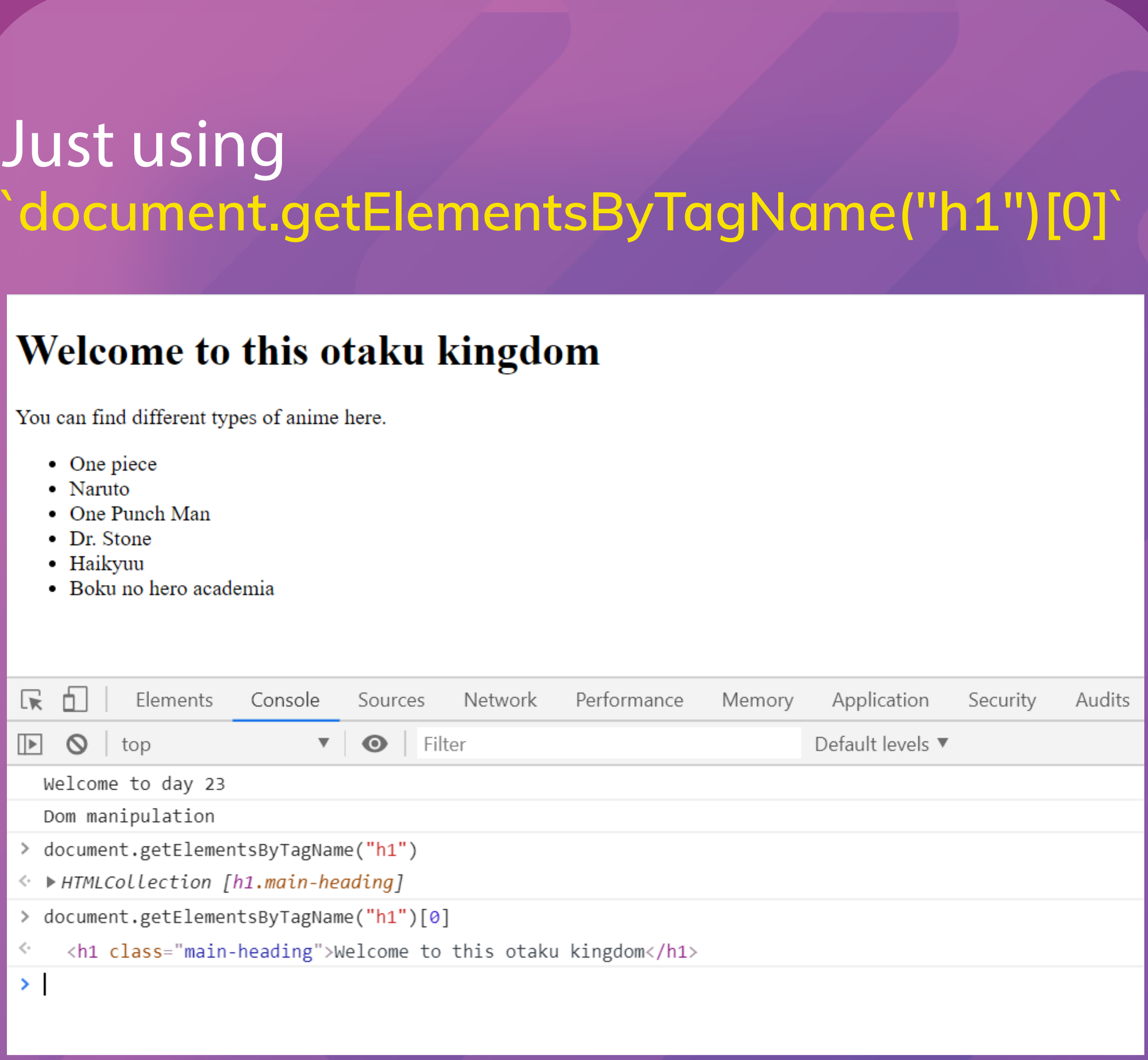
So how can get the value ?

Just using `document.getElementsByTagName("h1")[0]`

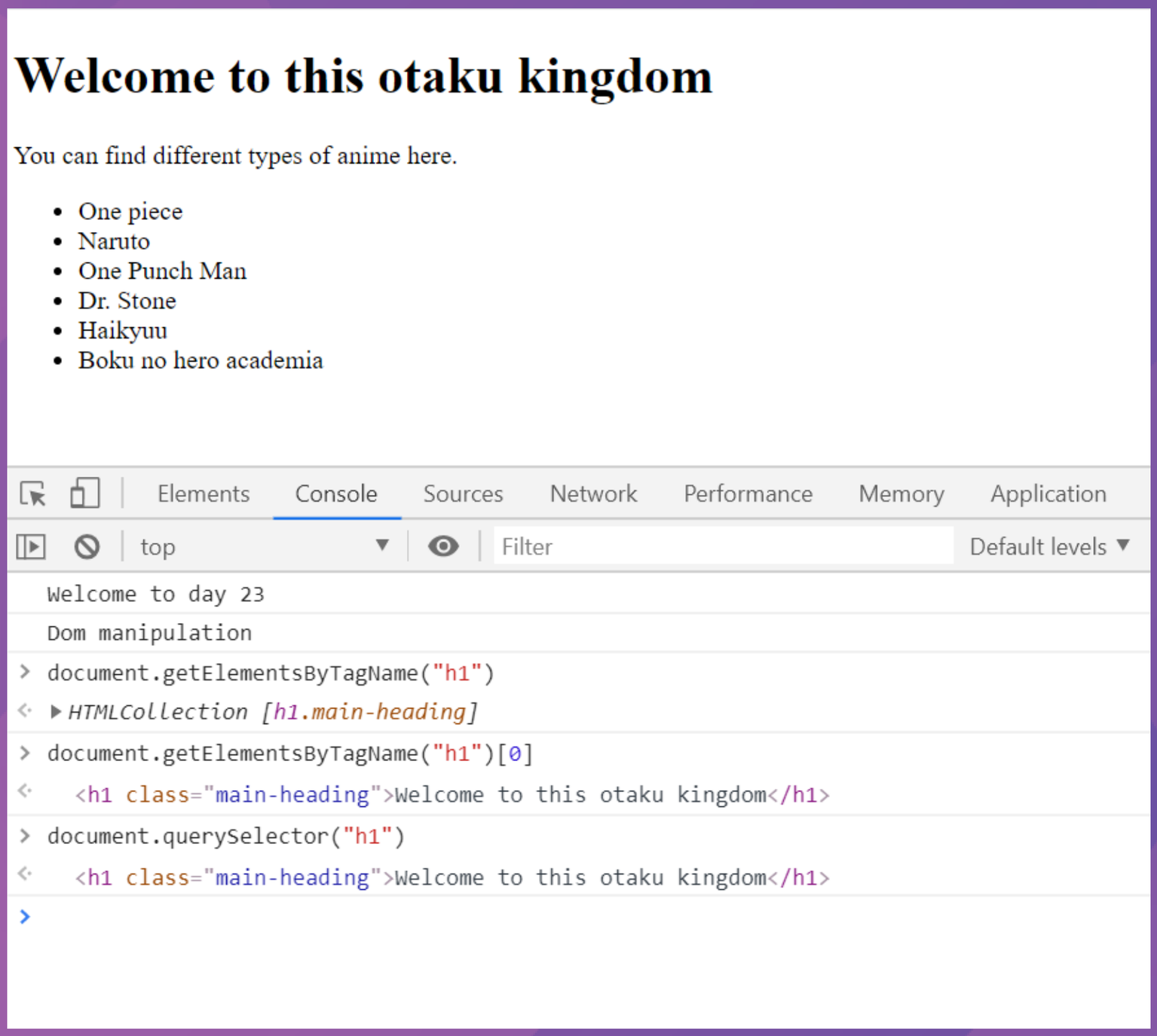
Welcome to this otaku kingdom

You can find different types of anime here.

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia



To avoid writing this big thing we can just write `document.querySelector("h1")` which is going to give the same thing for use.



So can you select the elements you have in your html ?

nerdjfpb

nerdjfpb

JAVASCRIPT SERIES-25

STYLE CHANGE IN DOM

nerd_jfpb

nerdjfpb

Today we are going to change the *style* of different elements in the dom using some *new technique* that we didn't learned yet.

We are going you use the *last day codes.*

Welcome to this otaku kingdom

You can find different types of anime here.

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia

nerdjfpb

nerdjfpb

We'll start by *selecting* the *h1* and then we'll apply some style with it. See when we use *document.querySelector("h1").style*

Welcome to this otaku kingdom

You can find different types of anime here.

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia

Elements | Console | Sources | Network | Performance | Memory | Application | Security | Audits

top | Filter | Default levels

```
Welcome to day 25
Style Change in DOM
> document.querySelector("h1").style
◂ ▾CSSStyleDeclaration {alignContent: "", alignItems: "", alignSelf: "", alignmentBaseline: "", all: "", …}
    cssText: ""
    length: 0
    parentRule: null
    cssFloat: ""
    alignContent: ""
    alignItems: ""
    alignSelf: ""
    alignmentBaseline: ""
    all: ""
    animation: ""
    animationDelay: ""
    animationDirection: ""
    animationDuration: ""
    animationFillMode: ""
```

nerd_jfpb

Now change the *background* *of h1.*

Using

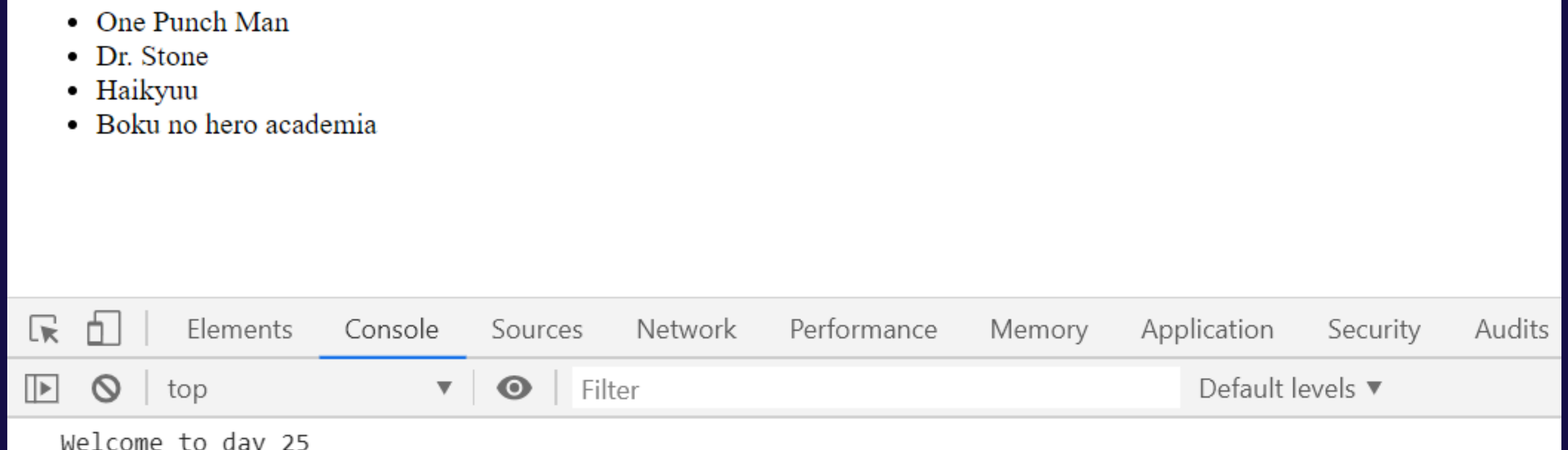
`*document.querySelector("h1").style.background = "red"*`

See result -

Welcome to this otaku kingdom

You can find different types of anime here.

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia



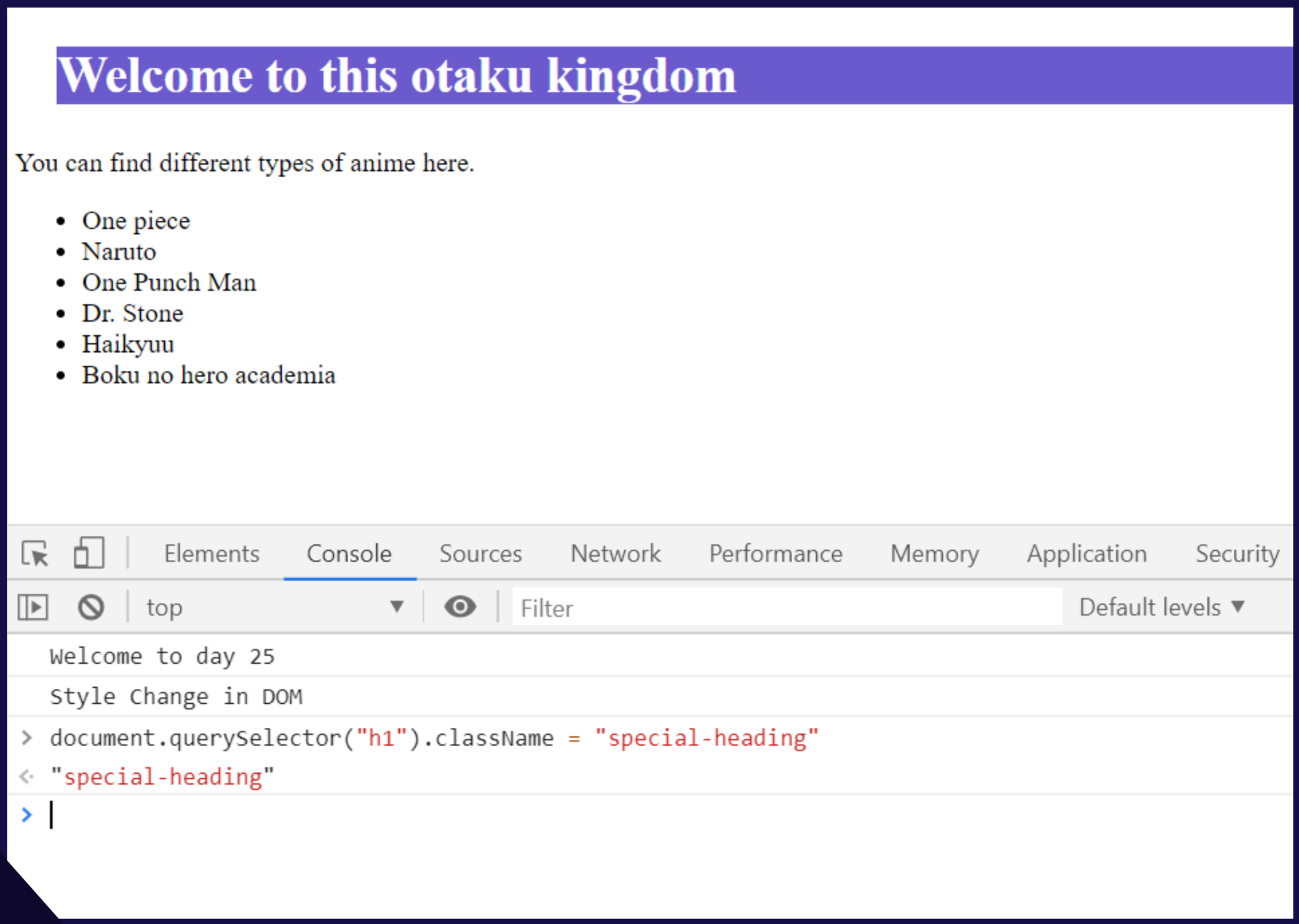
nerdjfpb

Now lets try to *add a class* in *h1 tag.*

Let's write some styles in *style.css*

nerd_jfpb

Using `document.querySelector("h1").className = "special-heading"` we can change right?



nerdjfpb

nerdjfpb

What are you going to do after this?

You have now superpower in your hand to *change anything in Dom!*

nerd_jfpb

nerdjfpb

nerdjfpb

Let's learn *something amazing* in next one!

nerd_jfpb

nerdjfpb

nerdjfpb

JAVASCRIPT SERIES-26

PLAY WITH DOM (QUERYSELECTOR)

nerd_jfpb

nerdjfpb

In last tutorial we worked with the dom for *change the styles*, now we are going to work with the dom for *changing some html.*

nerdjfpb

To change html before we use write.

Now we are going to use `*innerHTML*` which is going to *change the whole text with tag.*

Example -

Changed Text

You can find different types of anime here.

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia

Elements Console Sources Network Performance Memory

top Filter Defa

```
Welcome to day 26
Play With DOM (querySelector)
> document.querySelector("h1").innerHTML="<h3>Changed Text</h3>"
"<h3>Changed Text</h3>"
>
```

Up until now we've just used `*querySelector*`. If we use this on `*li*` then it will just give us first list of the item, so we need to call All of it by `*querySelectorAll*`

Welcome to this otaku kingdom

You can find different types of anime here.

- One piece
- Naruto
- One Punch Man
- Dr. Stone

Elements | Console | Sources | Network | Performance | Memory | Applicatic

top | Filter | Default levels

```
> document.querySelector("li")
    <li>One piece</li>
> document.querySelectorAll("li")
▼ NodeList(6) [li, li, li, li, li, li]
    length: 6
  ▶ 0: li
  ▶ 1: li
  ▶ 2: li
  ▶ 3: li
  ▶ 4: li
  ▶ 5: li
  ▶ __proto__: NodeList
>
```

nerdjfpb

Can you remember the array? Using *index* now we can select an element from `*li*` and can change it.

Welcome to this otaku kingdom

You can find different types of anime here.

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia

Elements | Console | Sources | Network | Performance | Memory | Applicati

top | Filter | Default level

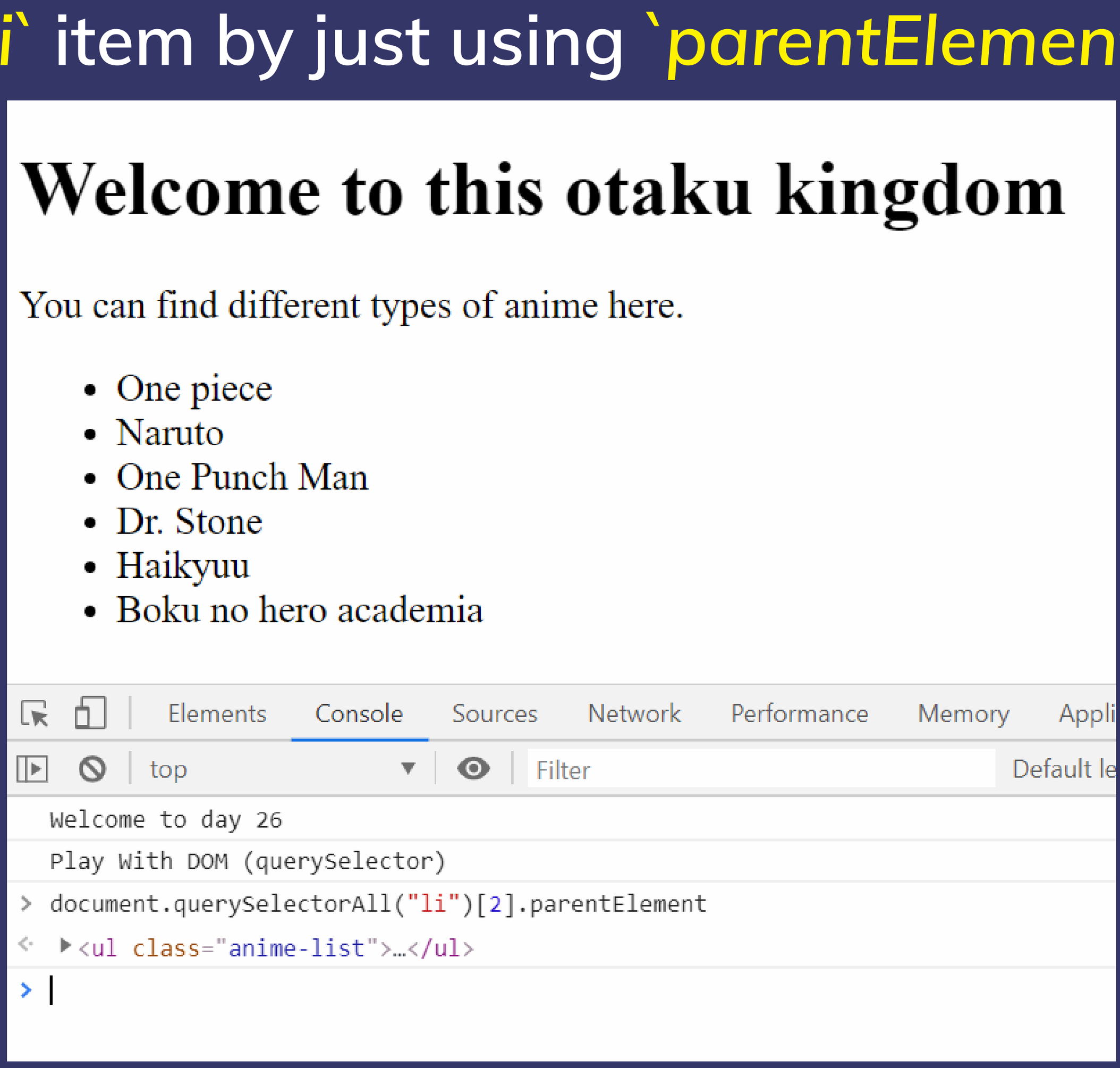
```
Welcome to day 26
Play With DOM (querySelector)
> document.querySelectorAll("li")[2].style.color="red"
"red"
>
```

Now we can get the parent from a `*li*` item by just using `*parentElement*`

Welcome to this otaku kingdom

You can find different types of anime here.

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia

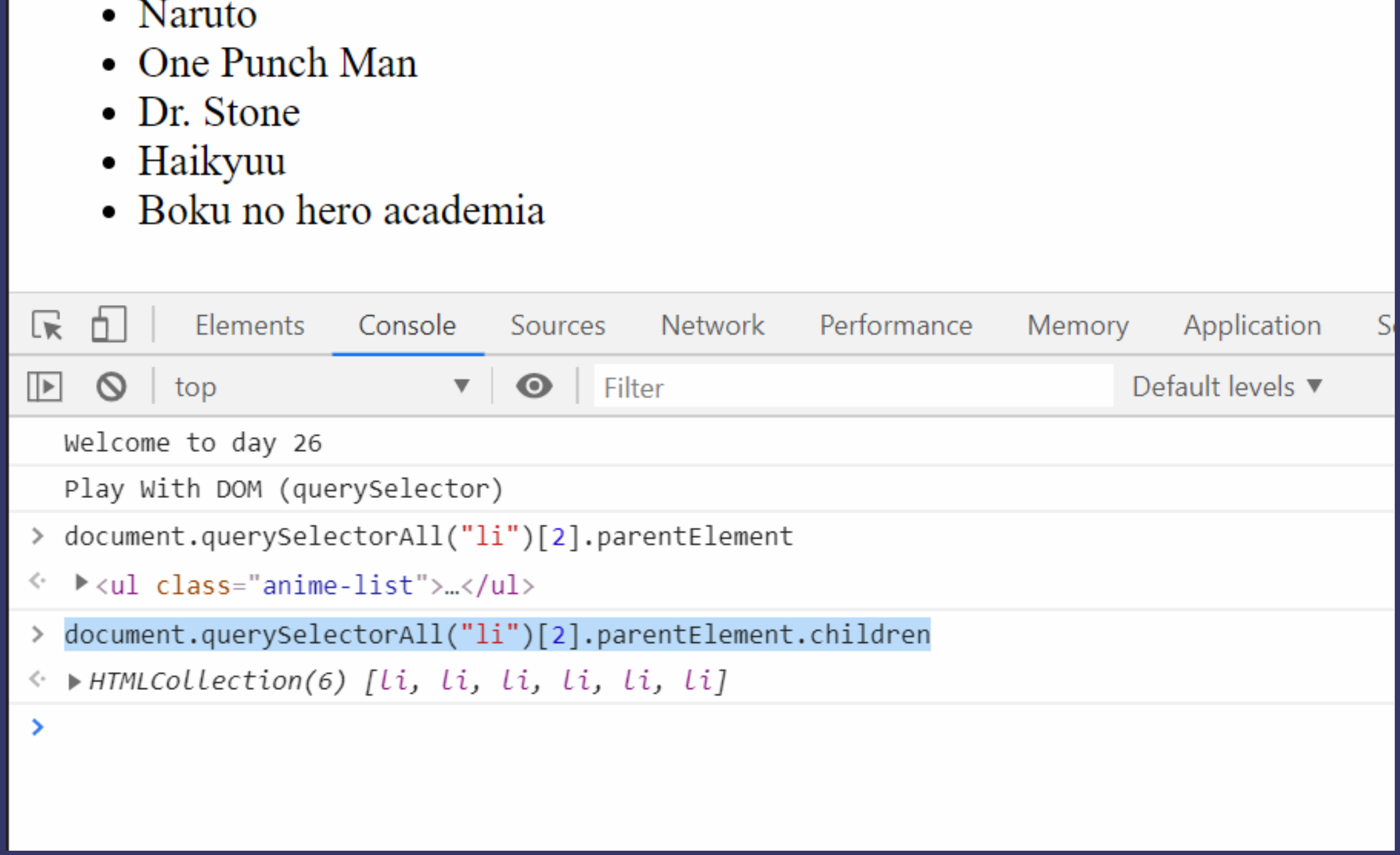


Like in this way we can use `children` too

Welcome to this otaku kingdom

You can find different types of anime here.

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia

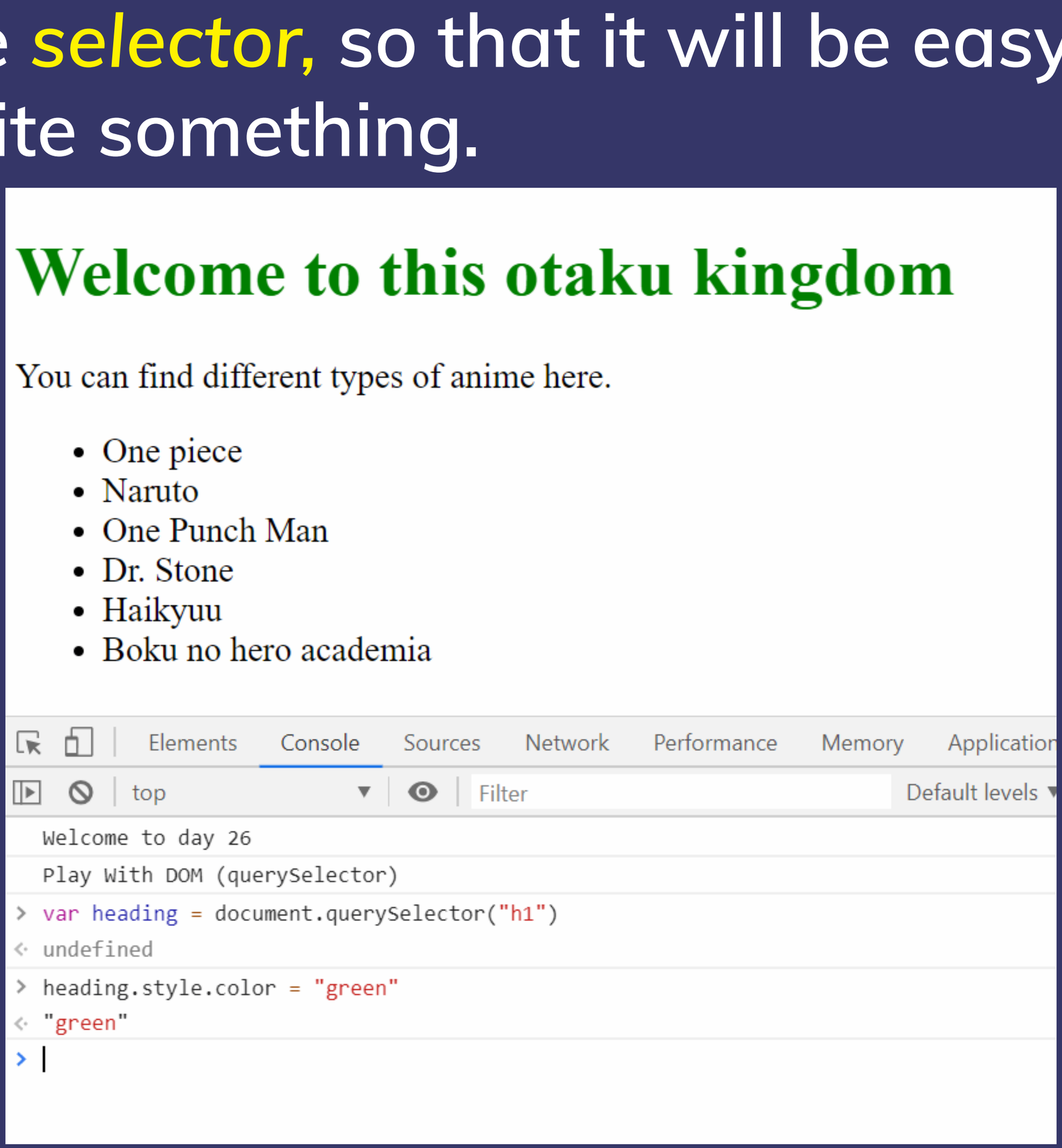


Finally we can use *variable to store* the *selector,* so that it will be easy to write something.

Welcome to this otaku kingdom

You can find different types of anime here.

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia



nerdjfpb

nerdjfpb

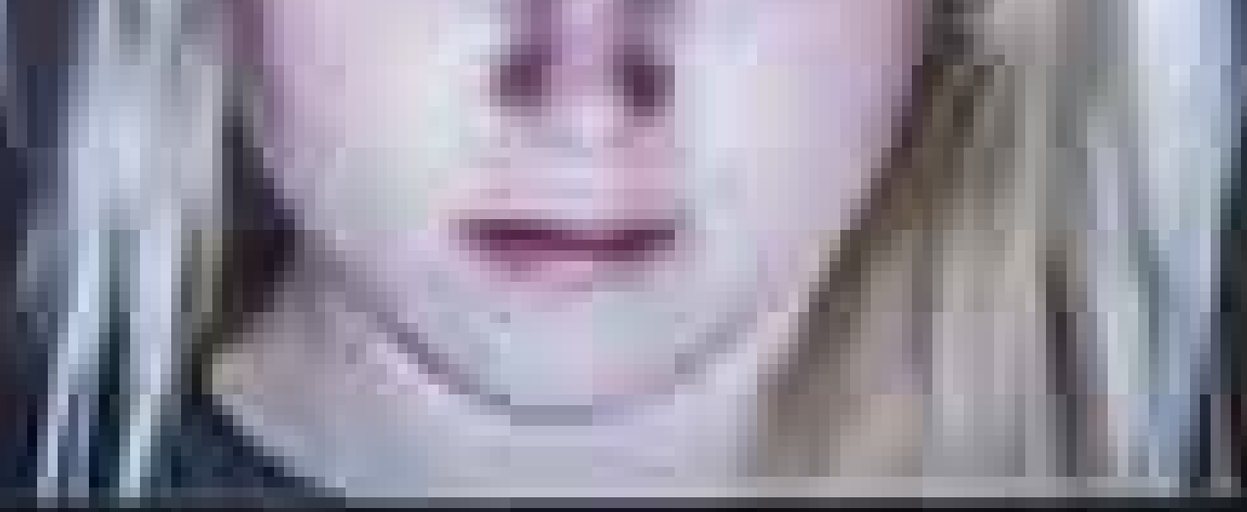
Can you play with the dom now?

If not follow & DM me part you didn't understand about it!

nerd_jfpb

nerdjfpb

nerdjfpb



JAVASCRIPT SERIES-27

DOM EVENT INTRODUCTION

nerd_jfpb

Events are things like - clicking, hovering on something or typing anything on search.

There can be many others event in a website.

We can do something based on this event using javascript.

We'll start playing just by using a button. Let's add a button to the html. You can find the code in github.

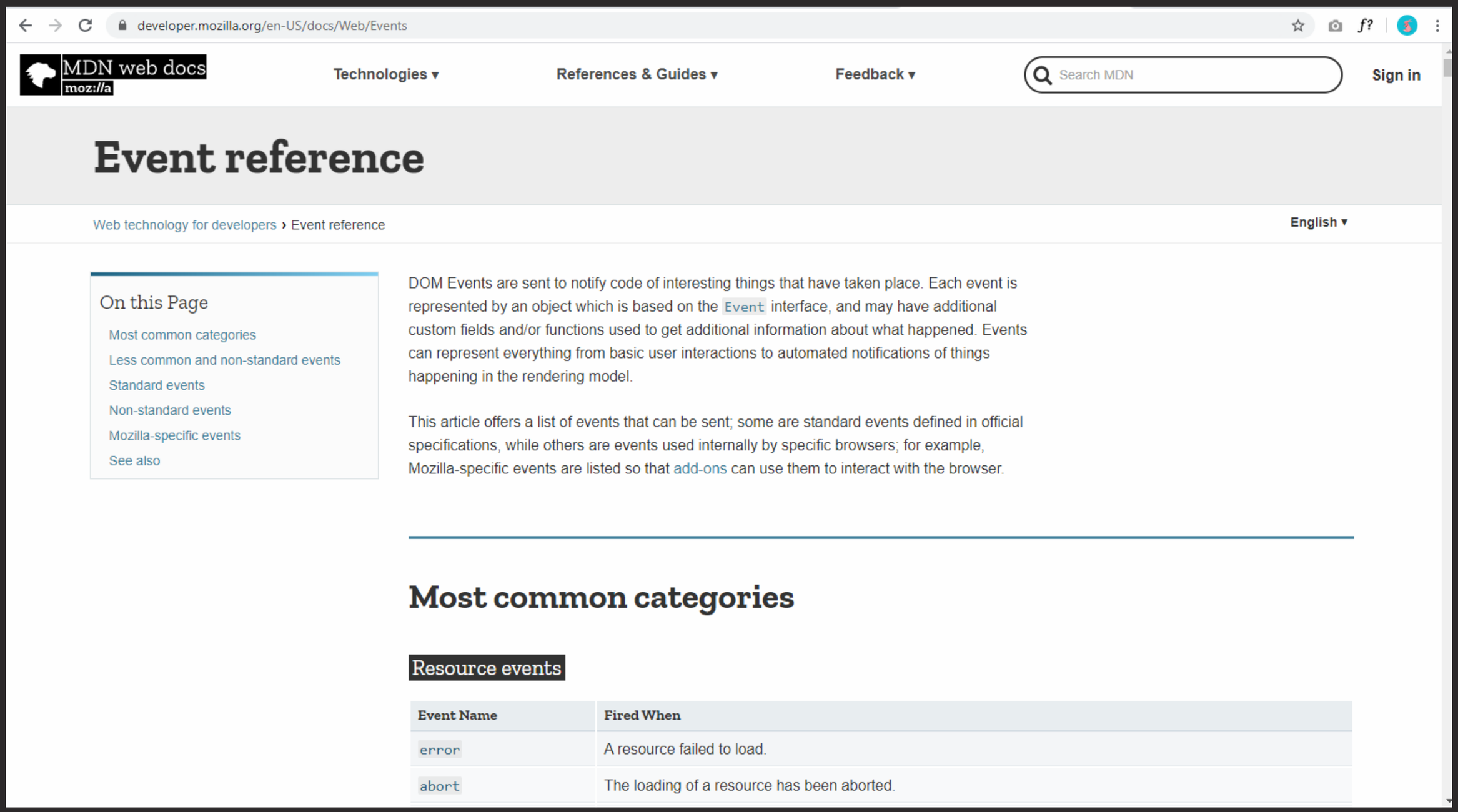
script.js | index.html

index.html > html > body > ul.anime-list > li

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>JavaScript Tutorial</title>
    <link rel="stylesheet" href="./css/style.css" />
  </head>
  <body>
    <h1 class="main-heading">Welcome to this otaku kingdom</h1>
    <p class="sub-heading">You can find different types of anime here.</p>
    <button>Anime Button</button>
    <ul class="anime-list">
      <li>One piece</li>
      <li>Naruto</li>
      <li>One Punch Man</li>
      <li>Dr. Stone</li>
      <li>Haikyuu</li>
      <li>Boku no hero academia</li>
    </ul>
    <script src="./js/script.js"></script>
  </body>
</html>
```

nerdjfpb

You can read more about events in mdn web docs.



nerd_jfpb

nerdjfpb

We are going to use a mouse event for today's example.

Mouse events

Event Name	Fired When
`auxclick`	A pointing device button (ANY non-primary button) has been pressed and released on an element.
`click`	A pointing device button (ANY button; soon to be primary button only) has been pressed and released on an element.
`contextmenu`	The right button of the mouse is clicked (before the context menu is displayed).
`dblclick`	A pointing device button is clicked twice on an element.
`mousedown`	A pointing device button is pressed on an element.
`mouseenter`	A pointing device is moved onto the element that has the listener attached.
`mouseleave`	A pointing device is moved off the element that has the listener attached.
`mousemove`	A pointing device is moved over an element. (Fired continously as the mouse moves.)
`mouseover`	A pointing device is moved onto the element that has the listener attached or onto one of its children.
`mouseout`	A pointing device is moved off the element that has the listener attached or off one of its children.
`mouseup`	A pointing device button is released over an element.
`pointerlockchange`	The pointer was locked or released.
`pointerlockerror`	It was impossible to lock the pointer for technical reasons or because the permission was denied.
`select`	Some text is being selected.
`wheel`	A wheel button of a pointing device is rotated in any direction.

nerdjfpb

First we'll get the button using `document.querySelector('button')`

nerd_jfpb

Second step will be add a EventListener.
Which is going to run the event.
addEventListener has 2 parameters.
One is the event name, another one is function where we are going to write what happens if the event occurs.

Let's write some code now

JS script.js × index.html

js > JS script.js > ...

```
1  console.log('Welcome to day 27')
2  console.log('DOM EVENT INTRODUCTION')
3
4  var button = document.querySelector('button')
5
6  button.addEventListener('click', function() {
7    console.log('Anime Button is Clicked')
8  })
9
```

See the result

Welcome to this otaku kingdom

You can find different types of anime here.

Anime Button

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia

nerdjfpb

nerdjfpb

Turn on the notification
so that you can stay with me
when I publish
the next part of this!

nerd_jfpb

nerdjfpb

nerdjfpb

JAVASCRIPT SERIES-28

ADD NEW ANIME ON THE LIST USING EVENT

nerd_jfpb

For this we need to have a html like -

Welcome to this otaku kingdom

You can find different types of anime here.

Enter anime name | Anime Button

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia

Writing html codes
You can find the codes in github.

```
    <link rel="stylesheet" href="./css/style.css" />
  </head>
  <body>
    <h1 class="main-heading">Welcome to this otaku kingdom</h1>
    <p class="sub-heading">You can find different types of anime here.</p>
    <input id="entered-anime" type="text" placeholder="Enter anime name" />
    <button id="add-anime">Anime Button</button>
    <ul id="anime-list">
```

Our first step will be selecting the ul, input & button and store it into variable.

```
var ul = document.getElementById('anime-list')
var animeName = document.getElementById('entered-anime')
var addAnime = document.getElementById('add-anime')
```

nerdjfpb

Now we are going to add a event on the button.

```
7
8  addAnime.addEventListener('click', function() {
9
10 })
11
```

We need to create an `li item` for adding the new anime.

```
addAnime.addEventListener('click', function() {
  var li = document.createElement('li')
})
```

nerdjfpb

We need to add the `animeName` value inside of that `li`

```
addAnime.addEventListener('click', function() {
  var li = document.createElement('li')
  li.appendChild(document.createTextNode(animeName.value))
  
})
```

Now we just need to add this in our list `ul`

```
addAnime.addEventListener('click', function() {
  var li = document.createElement('li')
  li.appendChild(document.createTextNode(animeName.value))
  ul.appendChild(li)
})
```

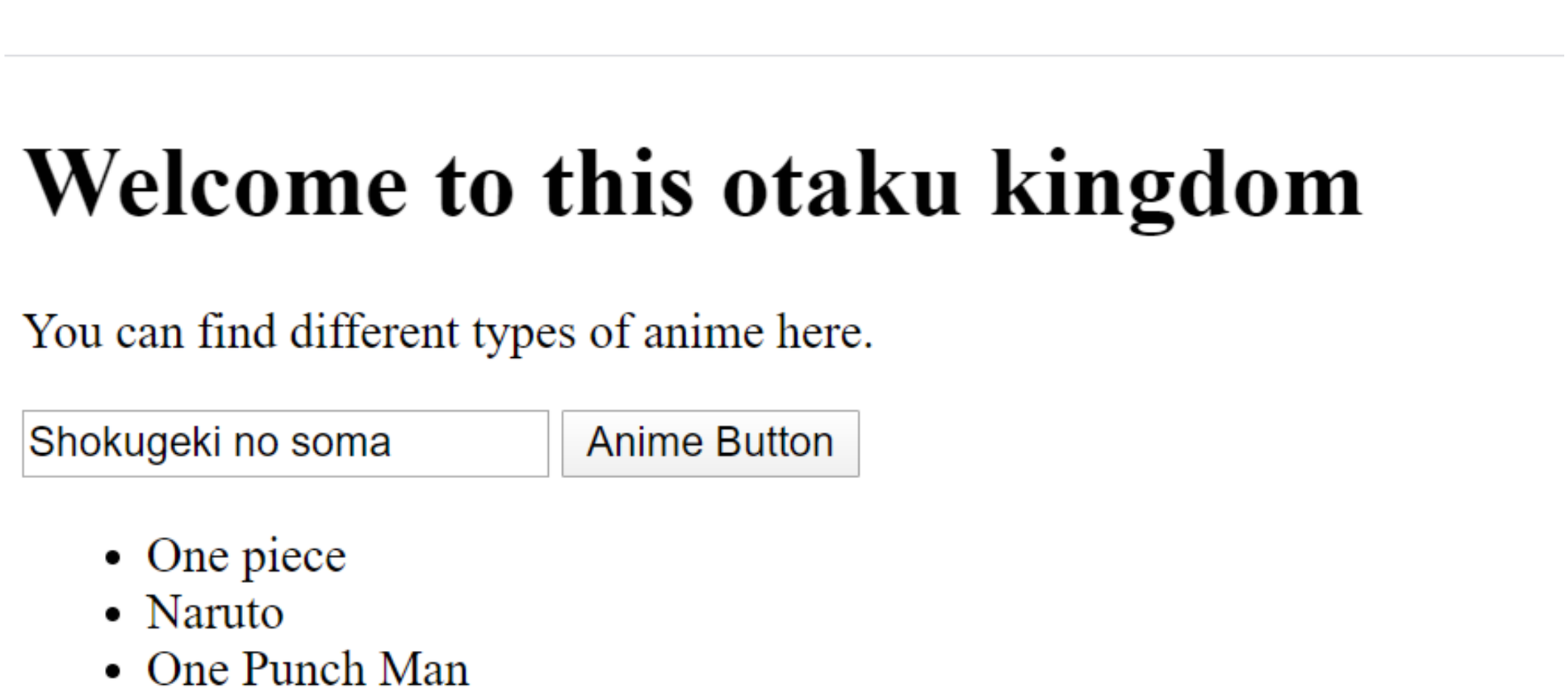
See the result in browser

Welcome to this otaku kingdom

You can find different types of anime here.

Shokugeki no soma | Anime Button

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia
- Shokugeki no soma



nerdjfpb

nerdjfpb

FOLLOW FOR SEE POSTS LIKE THIS!

nerd_jfpb

nerdjfpb

nerdjfpb

JAVASCRIPT SERIES-29

REFACTOR OUR LAST TUTORIAL CODE

nerd_jfpb

In our last code there is a easy bug, if you just add an empty anime name then it inserted in the list.

Welcome to this otaku kingdom

You can find different types of anime here.

Enter anime name | Anime Button

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia
-

Elements | Console | Sources | Network | Performance | Memory | Application | Sec

top | Filter | Default levels

```
Welcome to day 29
Refactor our last tutorial code
>
```

To solve this we just need to add a logic that value of string length should be greater than 1 or as much character you want.

```
addAnime.addEventListener('click', function() {
  if (animeName.value.length > 0) {
    var li = document.createElement('li')
    li.appendChild(document.createTextNode(animeName.value))
    ul.appendChild(li)
  }
})
```

nerdjfpb

nerdjfpb

This code is kinda messy right ?

Let's make it better together.

nerd_jfpb

We'll start take the anonymous out for the addEventListener.

```
7
8  function addAnimeInList() {
9    if (animeName.value.length > 0) {
10     var li = document.createElement('li')
11     li.appendChild(document.createTextNode(animeName.value))
12     ul.appendChild(li)
13   }
14 }
15
16 addAnime.addEventListener('click', addAnimeInList)
17
```

nerdjfpb

Now let's add a new function to get the input value.

```
function inputLengthCheck(input) {
  return input.value.length
}

function addAnimeInList() {
  if (inputLengthCheck(animeName) > 0) {
    var li = document.createElement('li')
    li.appendChild(document.createTextNode(animeName.value))
    ul.appendChild(li)
  }
}
```

We can break our code into a new function for just creating a list item.

```
11
12  function createListElement(item) {
13    var li = document.createElement('li')
14    li.appendChild(document.createTextNode(item.value))
15    ul.appendChild(li)
16  }
17
18  function addAnimeInList() {
19    if (inputLengthCheck(animeName) > 0) {
20      createListElement(animeName)
21    }
22  }
```

Now we have another bug, did you noticed yet? When we add a new anime list, the input field doesn't clear after that.

Welcome to this otaku kingdom

You can find different types of anime here.

Shokugeki No Soma | Anime Button

- One piece
- Naruto
- One Punch Man
- Dr. Stone
- Haikyuu
- Boku no hero academia
- Shokugeki No Soma

Elements | Console | Sources | Network | Performance | Memory | Application | Security

top | Filter | Default levels

```
Welcome to day 29
Refactor our last tutorial code
>
```

To solve this we just need to add a single line to clear the value.

```
11
12  function createListElement(item) {
13    var li = document.createElement('li')
14    li.appendChild(document.createTextNode(item.value))
15    ul.appendChild(li)
16    item.value = ''
17  }
18
```

nerdjfpb

nerdjfpb

DO YOU LIKED THE TUTORIAL ?
TURN ON THE POST NOTIFICATION TO FIND WHAT IS COMING NEXT!

nerd_jfpb

nerdjfpb

nerdjfpb

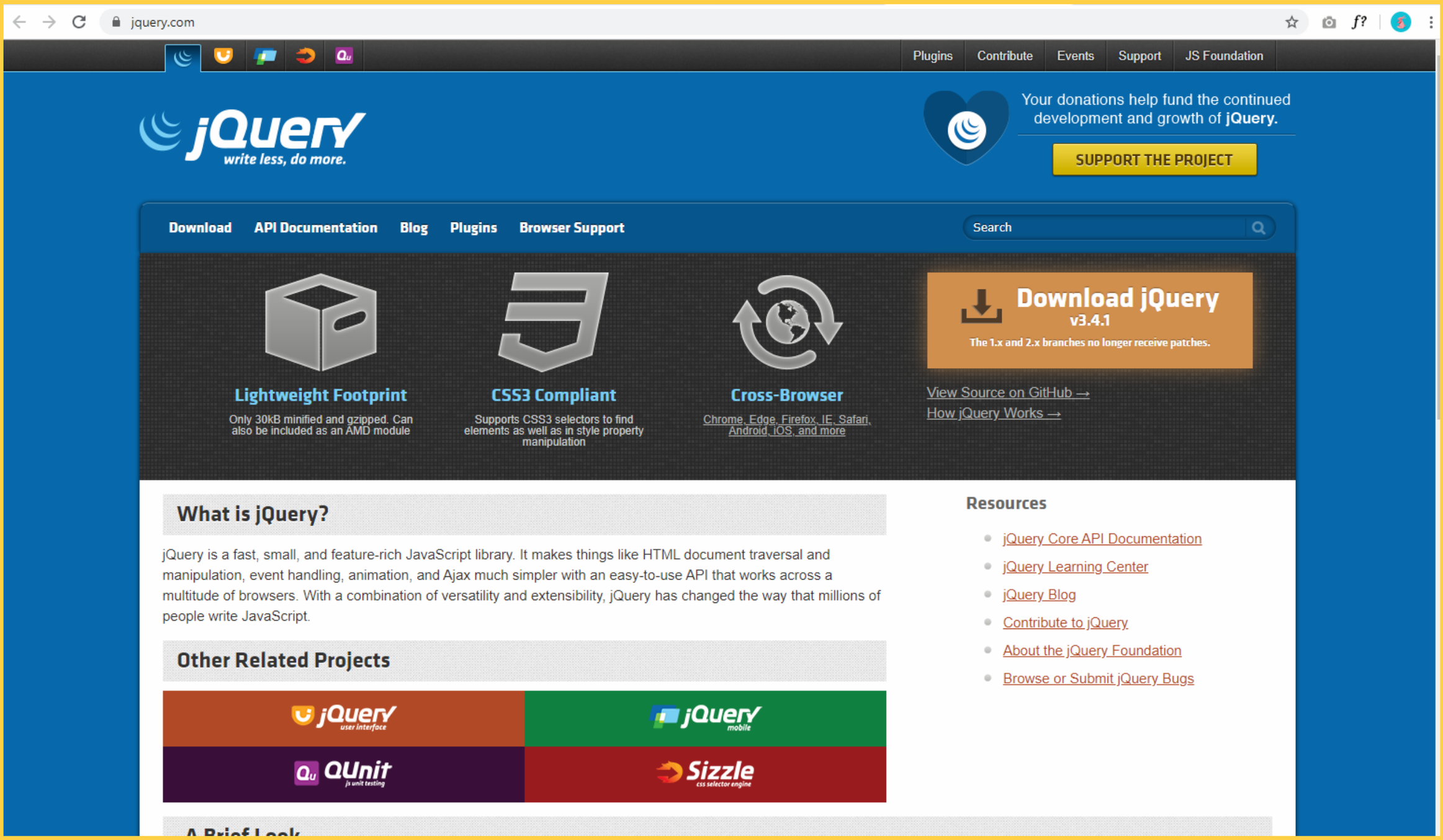
JAVASCRIPT SERIES-30

JQUERY ?

nerd_jfpb

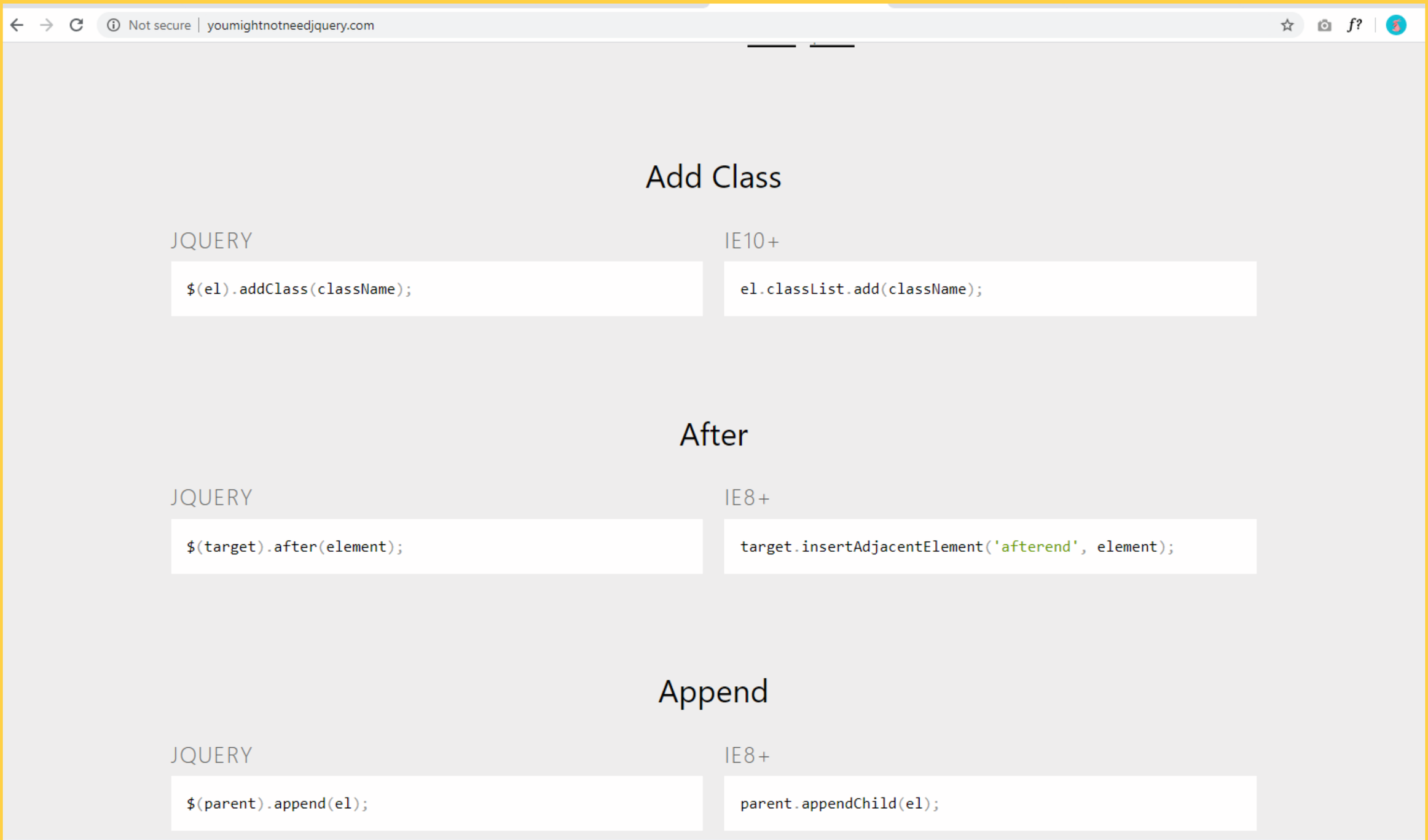
nerdjfpb

jQuery is a fast, small, and feature-rich JavaScript library. This helps to write js codes easily. This saves a ton of time.

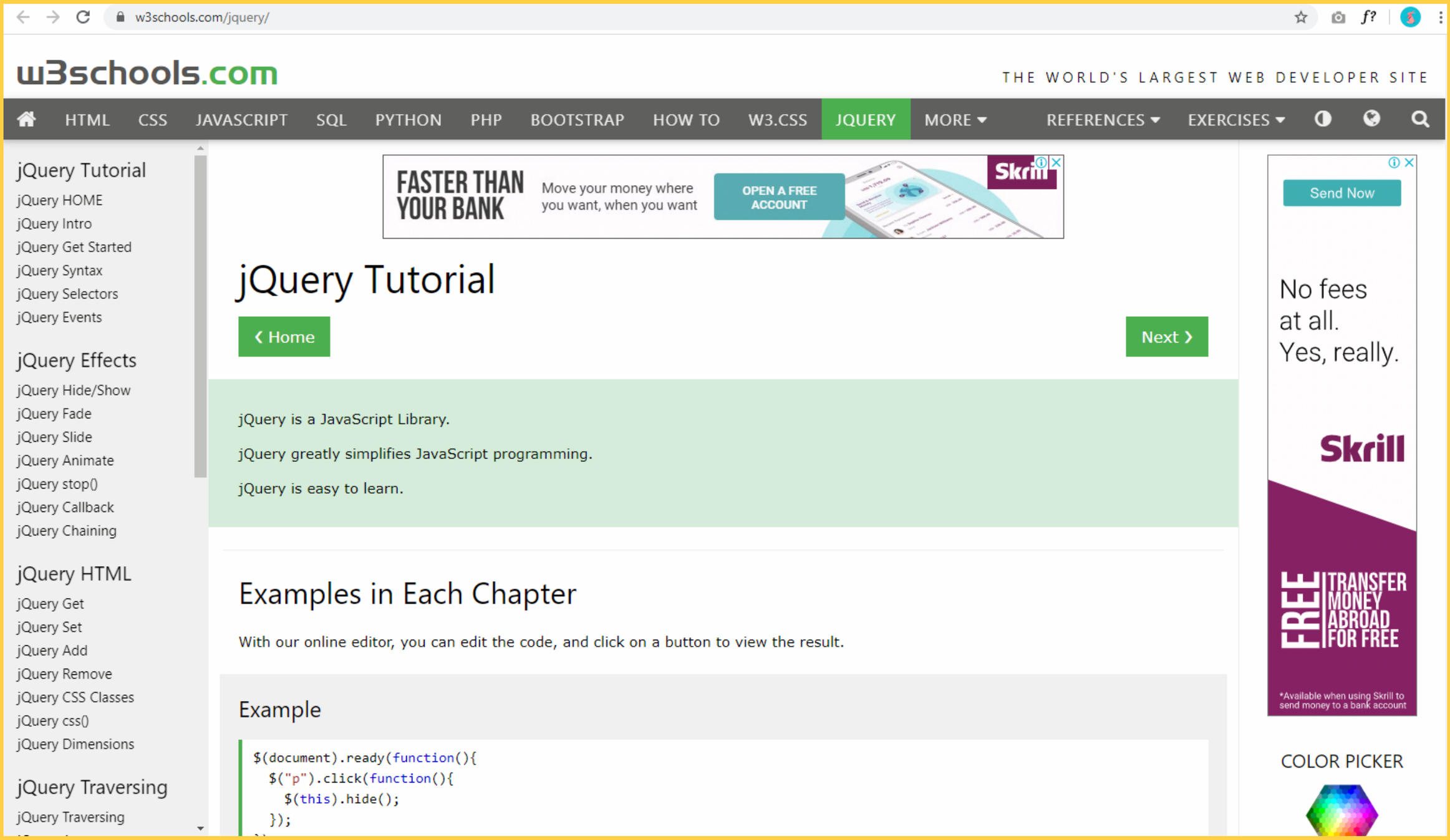


nerdjfpb

But people don't use it on new projects because there are more better library & you can do the same thing using vanila js. Check this out - http://youmightnotneedjquery.com/



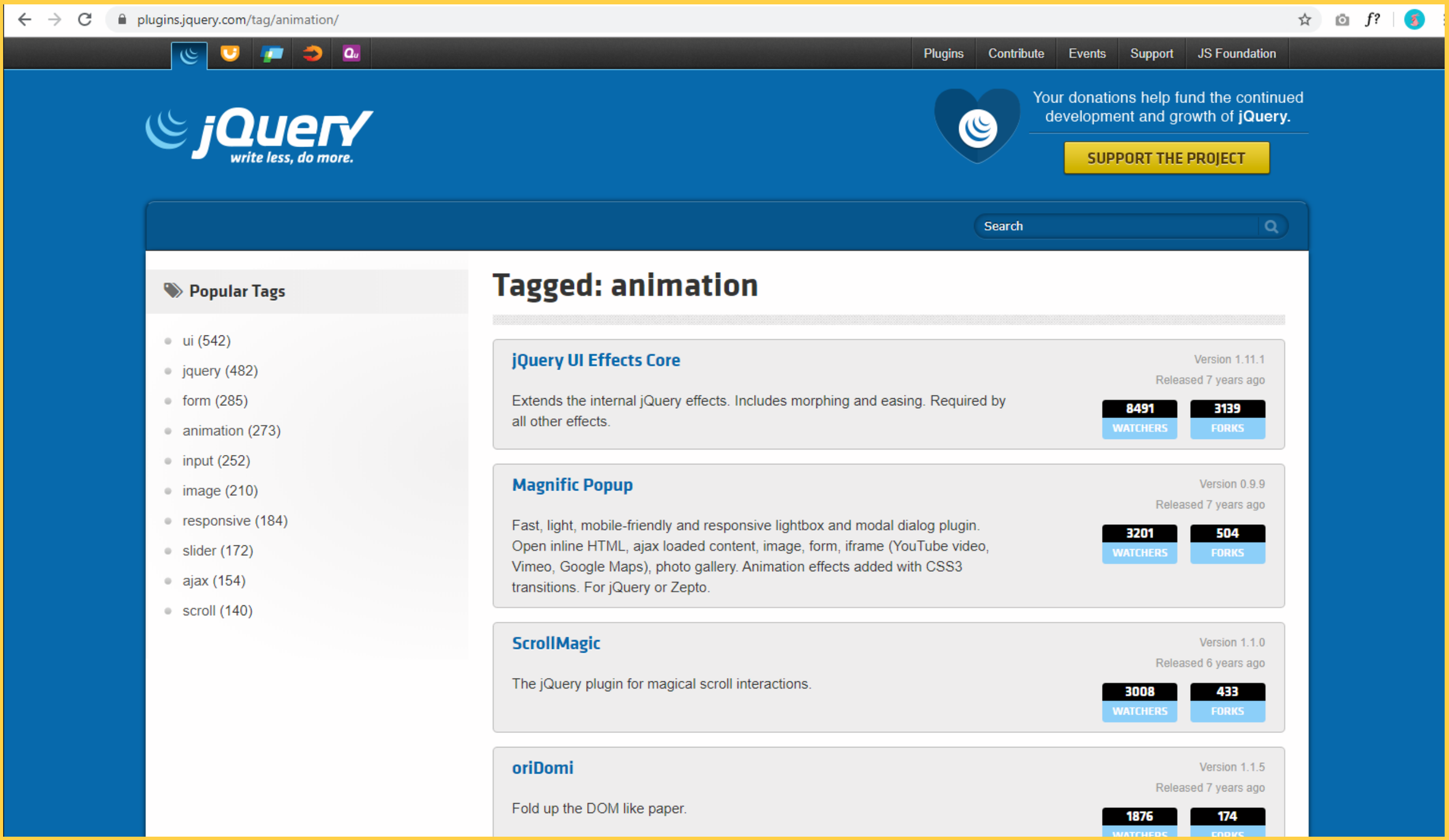
But jQuery saves a ton of time before and helped to write better javascript. So there are many projects which use jQuery. To understand those project you should learn it.



nerdjfpb

nerdjfpb

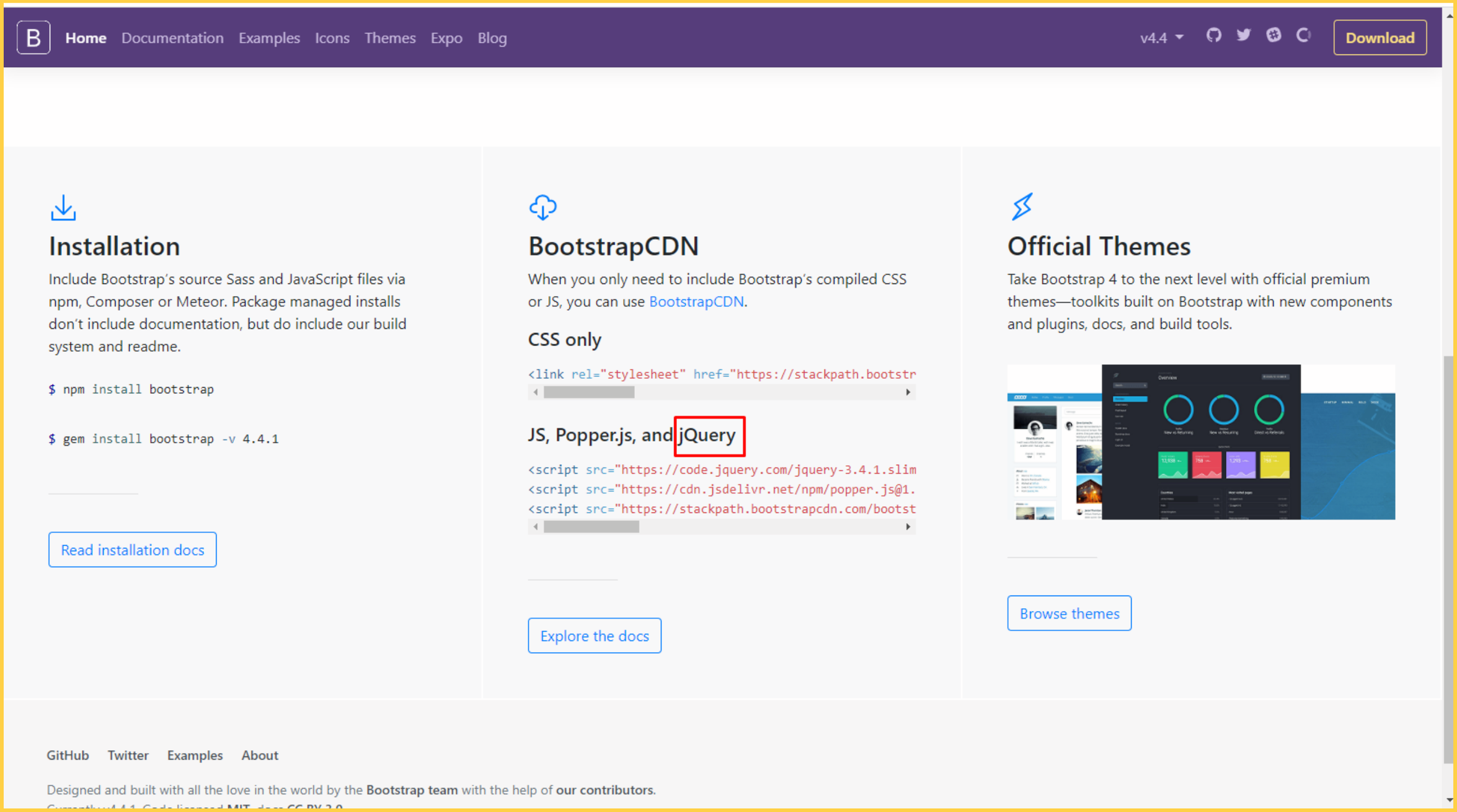
jQuery has lots of plugins which can help you to do things easier.



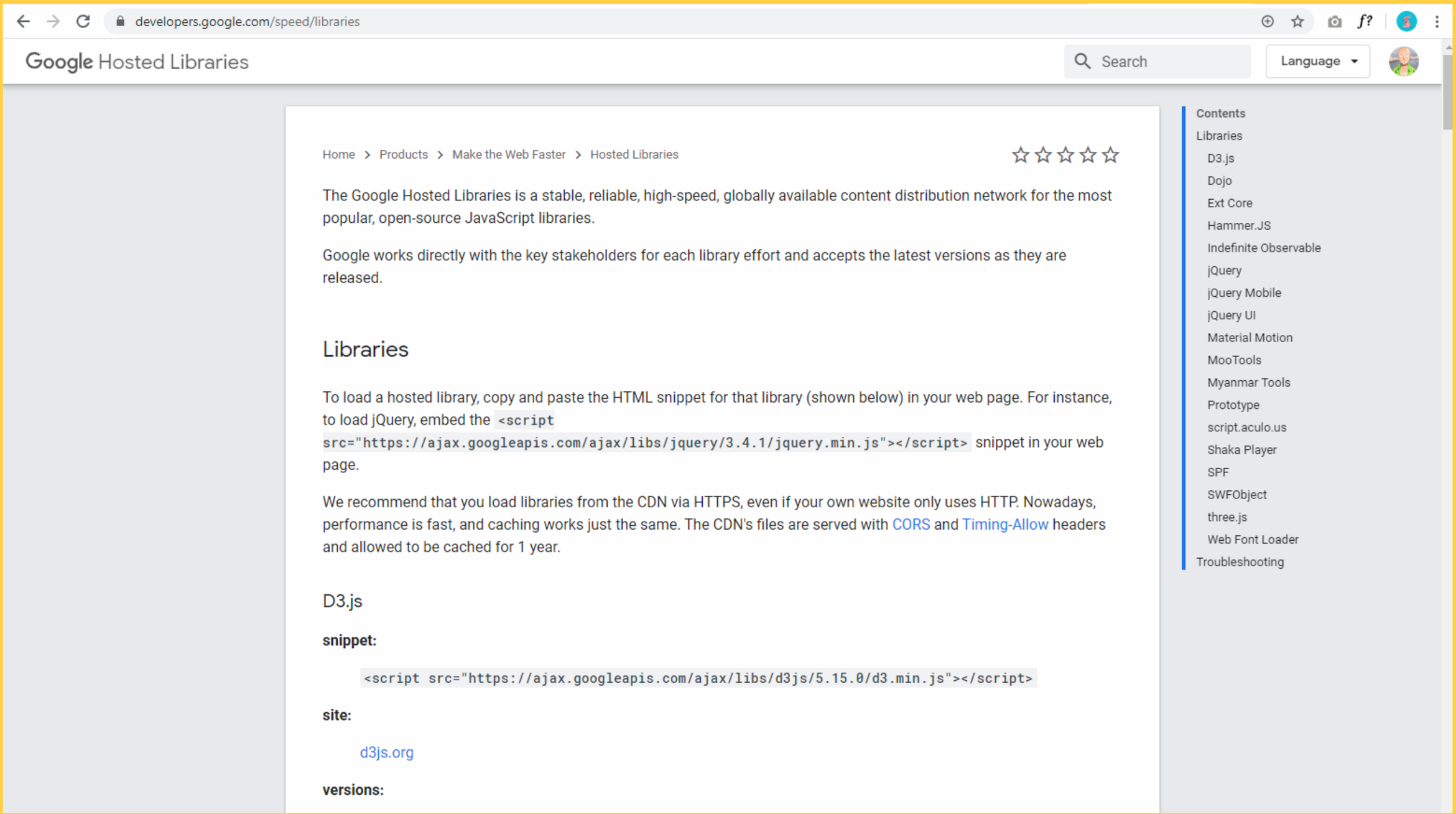
nerd_jfpb

nerdjfpb

Bootstrap still using the jQuery.



jQuery is really lightweight and it's also hosted in google so you can use cdn in your project very easily.



nerdjfpb

nerdjfpb

Easy JavaScript for Designers.

nerd_jfpb

nerdjfpb

It will be better if you start with other js library/frameworks instead of jQuery (use react,angular & vue logo image here)

nerdjfpb

nerdjfpb

DO YOU WANT TO USE JQUERY IN YOUR NEW PROJECT ?

nerd_jfpb