JavaScript

Lesson №5

General information

JavaScript

What is JavaScript

- Programming language, often written as JS
- Created for dynamic web pages and web applications
- Traditionally JS programs are called scripts
  (in the past there were JS interpreters)
- JS is not related to Java
- JS could run on a server or in user's browser

History of JS

History

- 1995 – In Netscape Navigator 2.0 (JS was initially called Mocha, then LiveScript)
- 1996 – JS standartized as ECMAScript
- 1997 – First version
- 1998 – Second version
- 1999 – Third version
- 2009 – Fifth version (there is no fourth version)
- 2011 – Version 5.1

JS capabilities

Capabilities

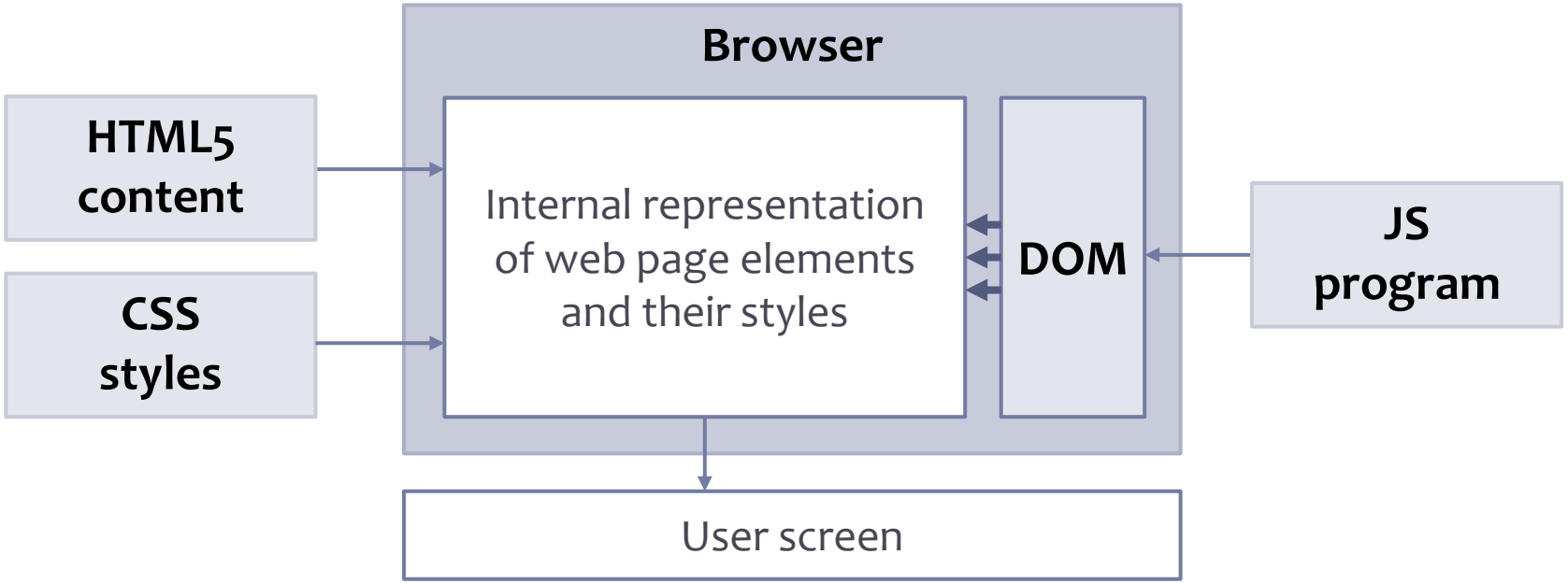
- General purpose language
- Procedural, functional and object-oriented programming
- Access to HTML elements and their styles
- Processing mouse motion, button clicks, …

In SUICA

- Programs are in JS
- Programs use Suica, which is written in JS

Relation with HTML and CSS

- Access to properties and styles of HTML elements via DOM (Document Object Model)



Appearance

The appearance of a JS code

- Looks like C (somewhat)
- No pointers
- No types of variables

Usage

Location

- Server-side JS (not used in this course)
- Client-side JS

Usage

- External file used by all pages in a site
- Script element in a web page
- Attribute to a single HTML element

Working environent

JS console

- A developer's module in browsers supporting JS

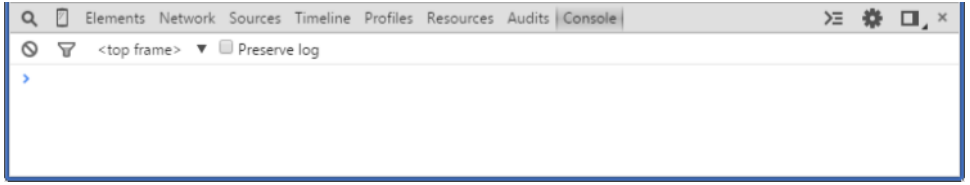
Used for

- Error messages and warnings
- Printing text results
- Real-time execution of commands

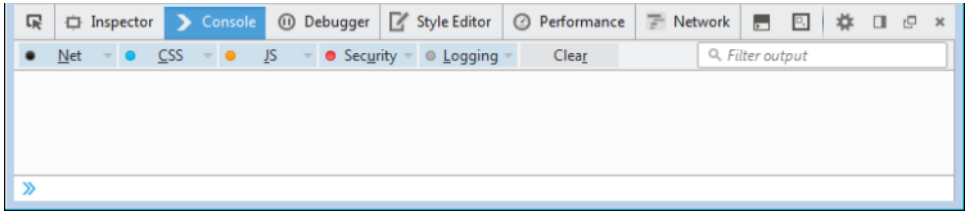
```
console.log(…);
```

Opening the console

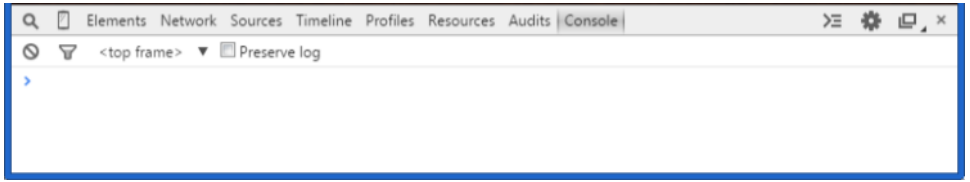
- Chrome & Ctrl-Shift-I . . . .



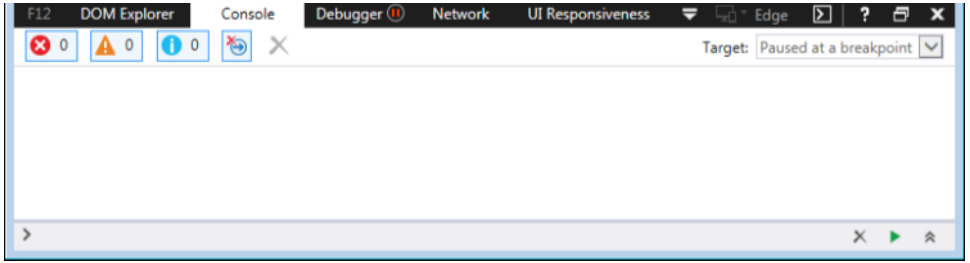
- FireFox & Ctrl-Shift-I . . . .



- Opera & Ctrl-Shift-I . . . . .



- Internet Explorer & F12 . .

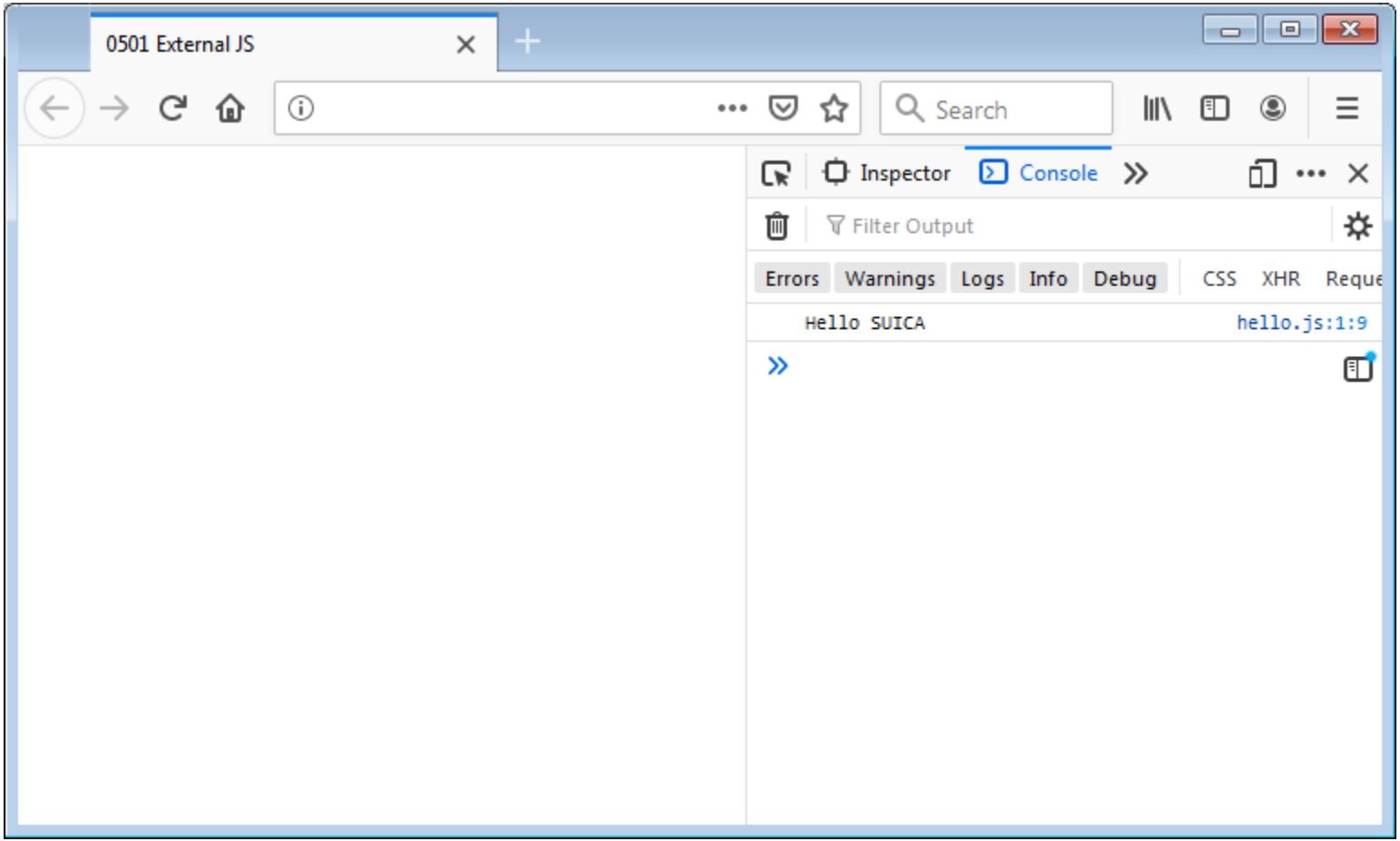


External JS

JS in external files

- Sharing JS code among several pages
- Used in <script> inside <head> or <body>, attribute src points the path and name of the JS file
- Element <script> must be closed by </script>
- Traditional extension .js

```
<head>
  <script src="hello.js"></script>
</head>
```


0501 External JS
Search
Inspector
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Reque
Hello SUICA
hello.js:1:9

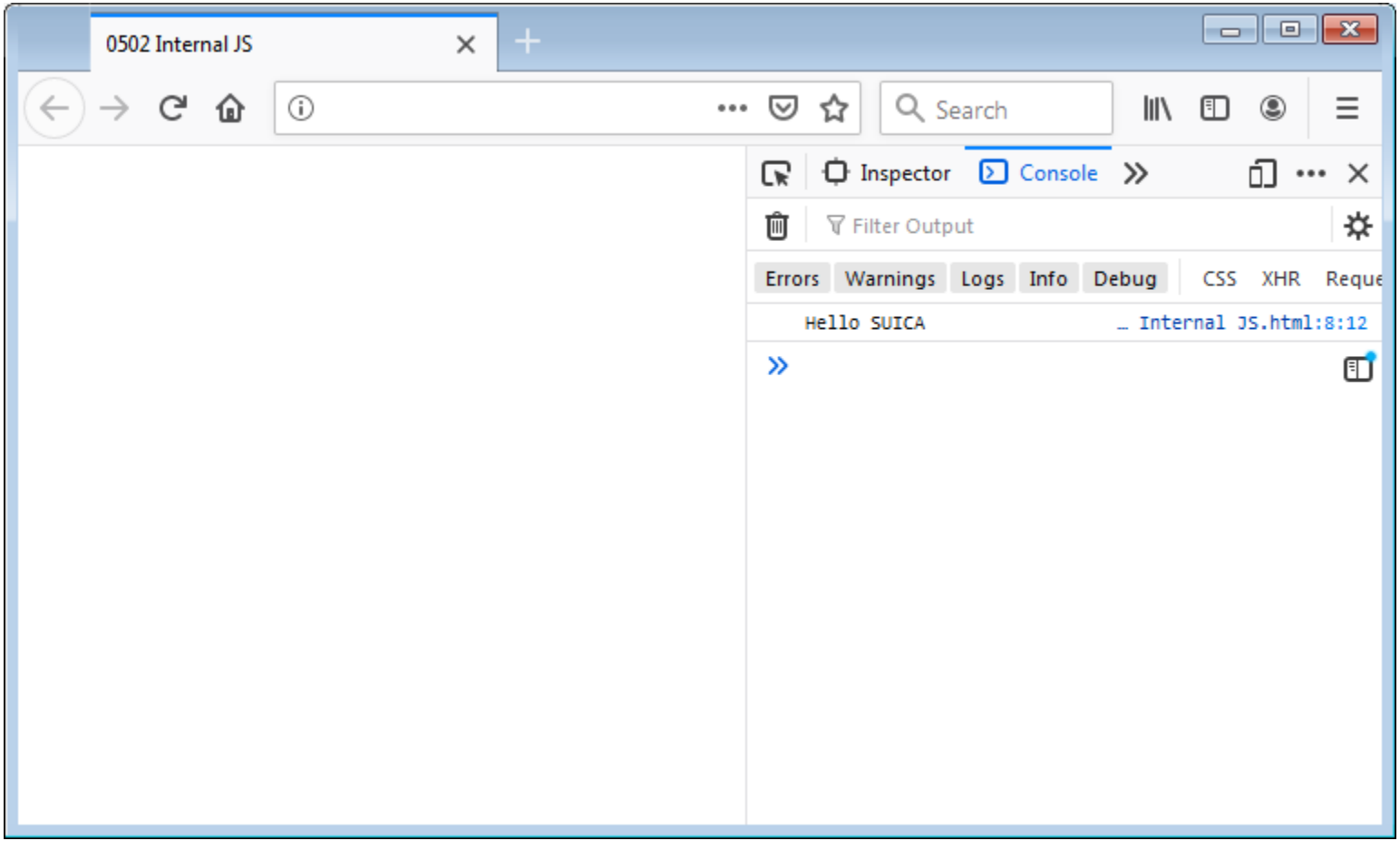
TRY IT

Internal JS

JS in <script>

- Located in <head> or in <body>

```
<head>
   <title>0502 Internal JS</title>
   <script>
      console.log('Hello SUICA');
   </script>
</head>
```


0502 Internal JS
Search
Inspector
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Hello SUICA
Internal JS.html:8:12

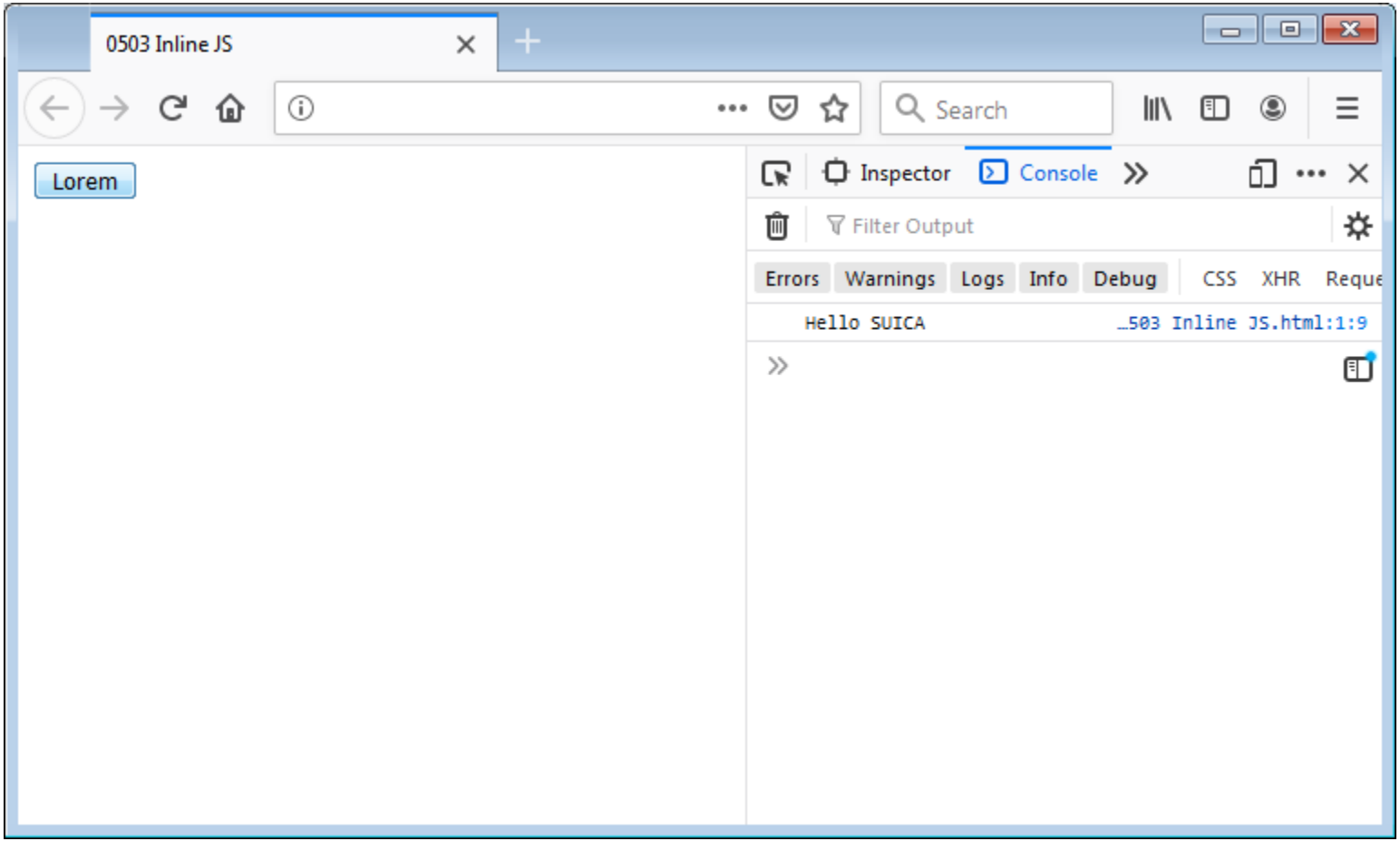
TRY IT

Inline JS

JS in attribute of HTML element

- Used in attributes reacting on user's interaction (e.g. clicking, hovering, etc.)

```
<body>
   <button onclick="console.log('Hello SUICA');">
      Lorem
   </button>
</body>
```


0503 Inline JS
Search
Lorem
Inspector
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Reque
Hello SUICA
…503 Inline JS.html:1:9

TRY IT

Data types

Data types

Simple data types

- Numbers
- Strings
- Booleans

Complex data types

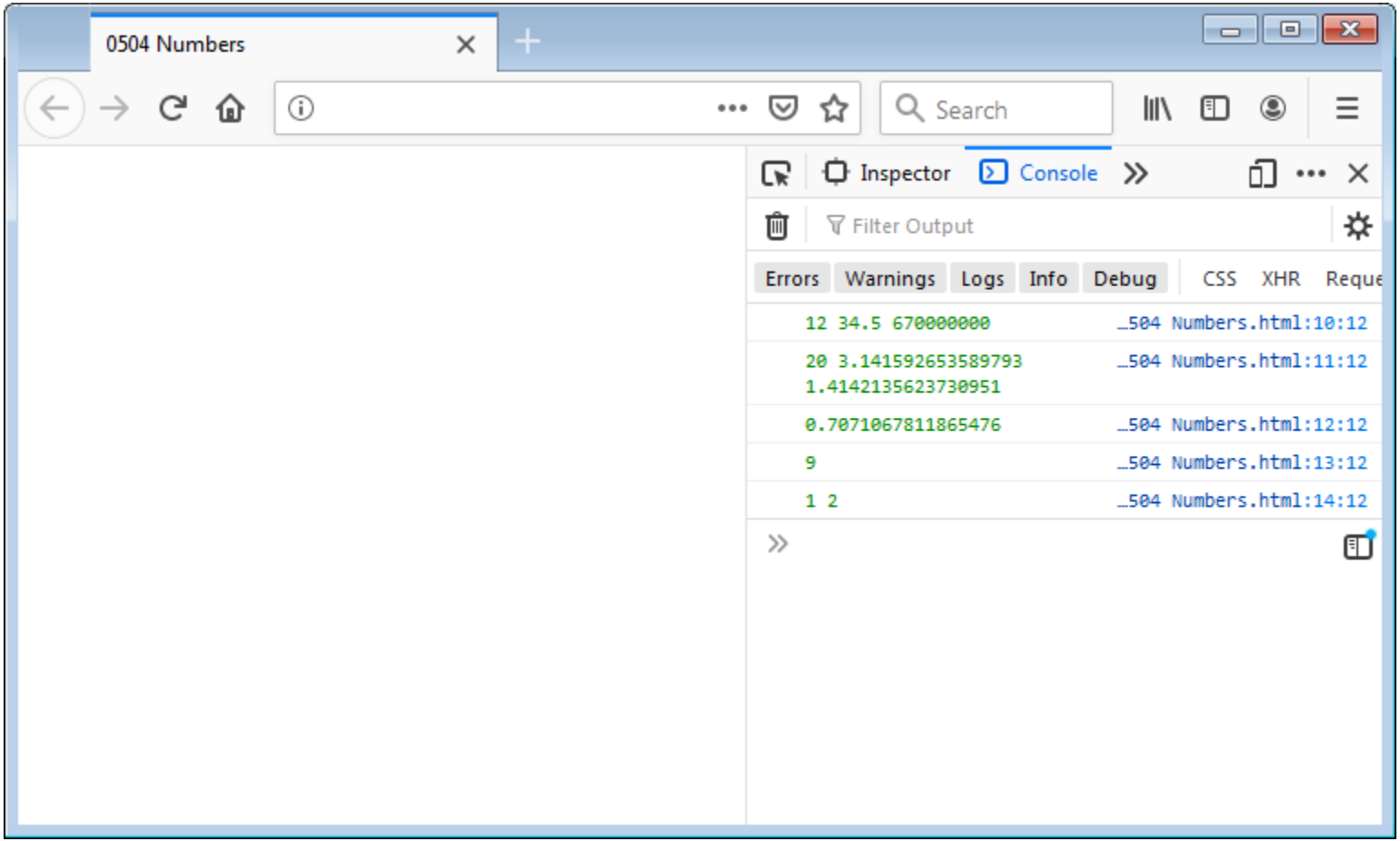
- Arrays
- Objects
- Functions

Simple data types

Numbers

- No distinction between integer and floating point numbers
- Supported are all traditional operations
- Functions and constants are defined in Math

```
console.log( 12, 34.5, 6.7e+8 );
console.log( (2+3)*4, Math.PI, Math.SQRT2 );
console.log( Math.sin(Math.PI/4) );
console.log( Math.max(3,1,4,1,5,9,2,6) );
console.log( Math.floor(1.8), Math.round(1.8) );
```

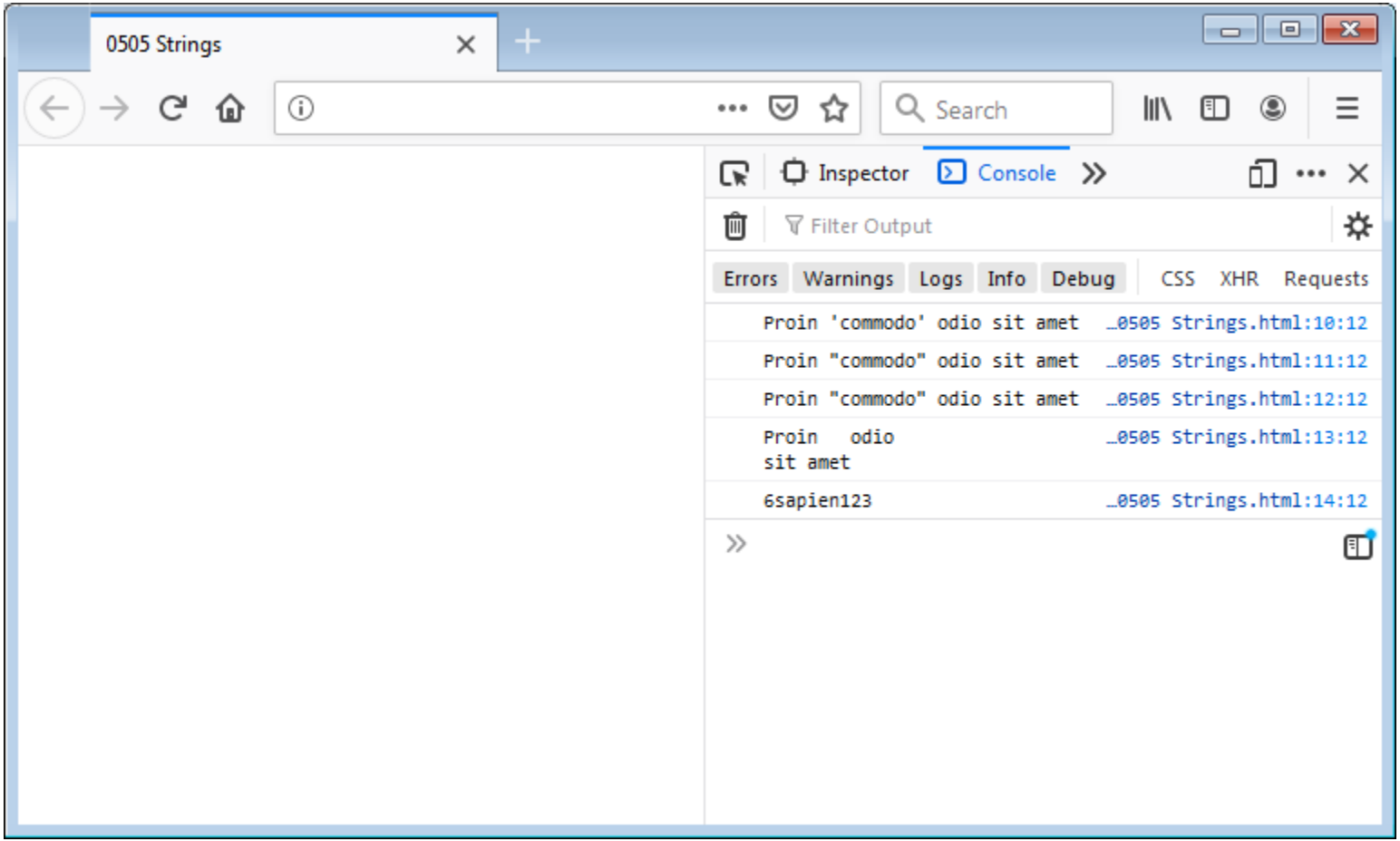

0504 Numbers
Search
Inspector
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Reque
12 34.5 670000000
…504 Numbers.html:10:12
20 3.141592653589793
1.4142135623730951
…504 Numbers.html:11:12
0.7071067811865476
…504 Numbers.html:12:12
9
…504 Numbers.html:13:12
1 2
…504 Numbers.html:14:12

TRY IT

Strings

- Constants framed in quotes "…" or '…'
- Special symbols preceded by \
- New line \n and tab \t
- Automatic conversion of numbers to strings

```
console.log("Proin 'commodo' odio sit amet");
console.log('Proin "commodo" odio sit amet');
console.log("Proin \"commodo\" odio sit amet");
console.log('Proin\todio\nsit amet');
console.log(1+2+3+'sapien'+1+2+3);
```

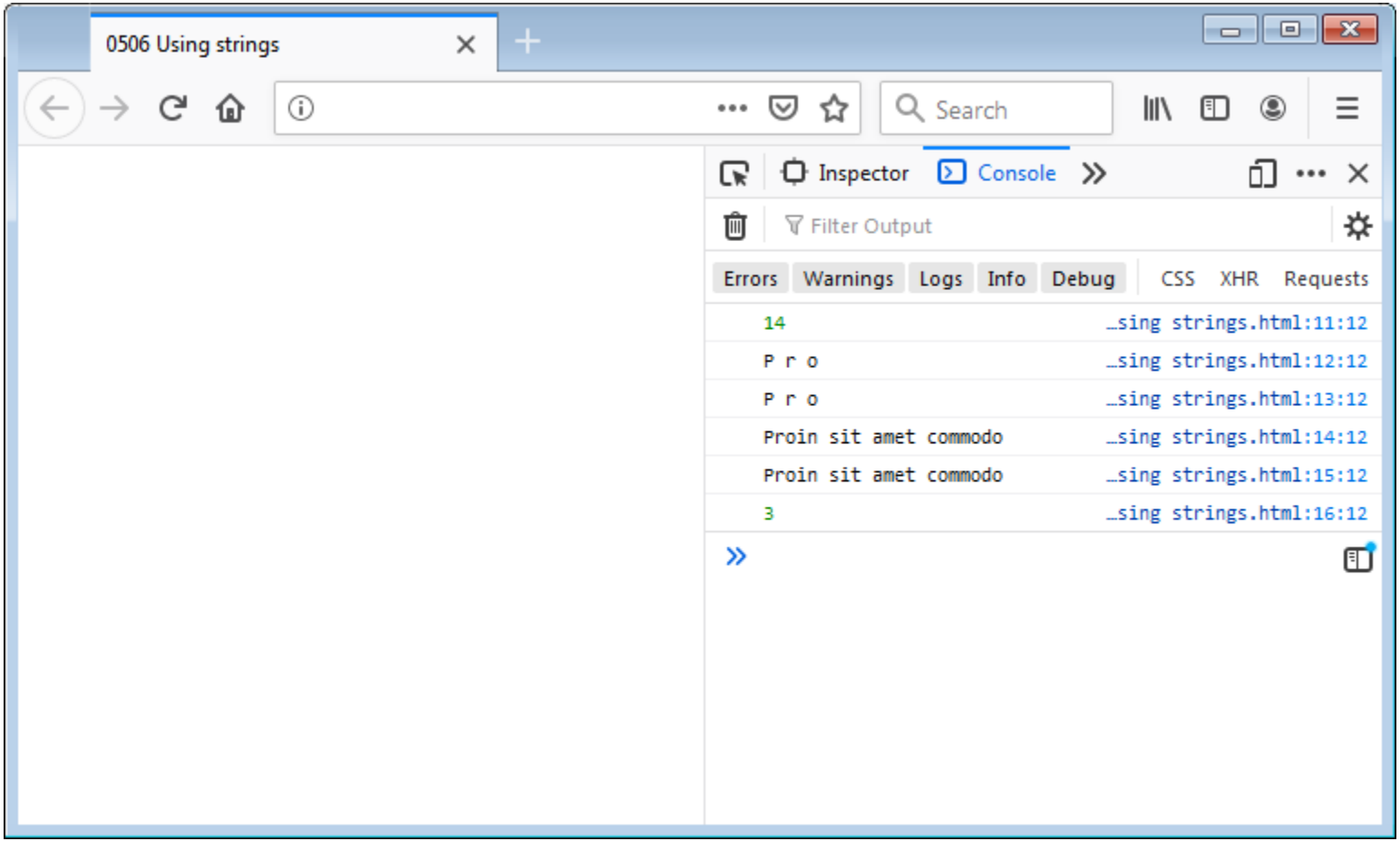

0505 Strings
Search
Inspector
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
Proin 'commodo' odio sit amet
…0505 Strings.html:10:12
Proin "commodo" odio sit amet
…0505 Strings.html:11:12
Proin "commodo" odio sit amet
…0505 Strings.html:12:12
Proin odio
sit amet
…0505 Strings.html:13:12
6sapien123
…0505 Strings.html:14:12

TRY IT

Using strings

- Property length for string length
- Method charAt and […] to access a character
- Method concat and + to concatenate
- Methods indexOf and search to search for a substring

```
var a = "Proin sit amet";
console.log(a.length);
console.log(a.charAt(0),a.charAt(1),a.charAt(2));
console.log(a[0],a[1],a[2]);
console.log(a.concat(" commodo"));
console.log(a+" commodo");
console.log(a.indexOf('in'));
```

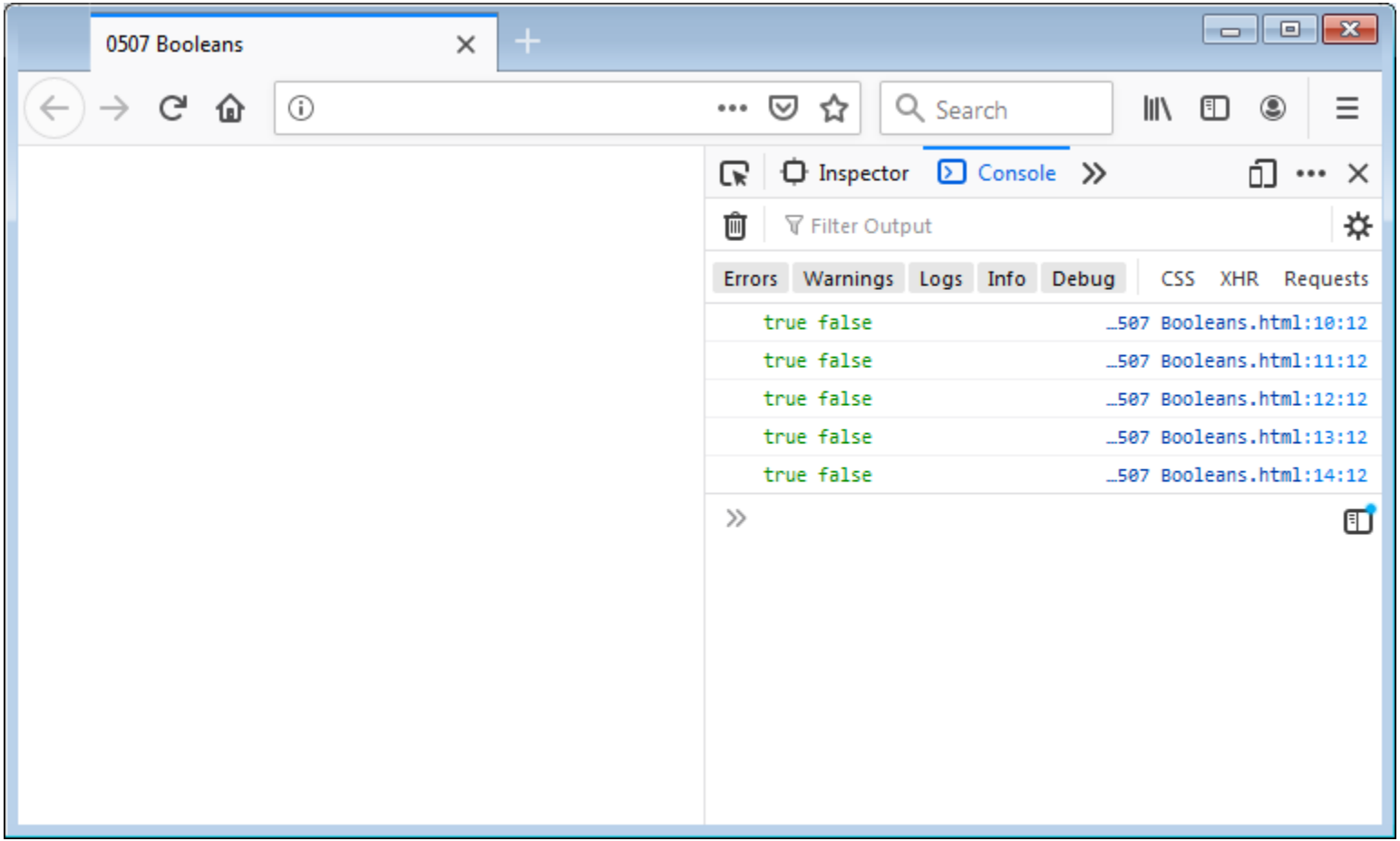

0506 Using strings
Search
Inspector
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
14 …sing strings.html:11:12
P r o …sing strings.html:12:12
P r o …sing strings.html:13:12
Proin sit amet commodo …sing strings.html:14:12
Proin sit amet commodo …sing strings.html:15:12
3 …sing strings.html:16:12

TRY IT

Booleans

- Constants true and false
- Standard boolean operators &&, ||, !
- Coparison results with <, = and > are boolean
- Function Boolean defines whether a value will be considered as true or false

```
console.log(true,false);
console.log(5>3,2<=1);
console.log(true || false, true && false);
console.log(Boolean("zero"),Boolean(0));
console.log(Boolean(5),Boolean(""));
```


0507 Booleans
Search
Inspector
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
true false …507 Booleans.html:10:12
true false …507 Booleans.html:11:12
true false …507 Booleans.html:12:12
true false …507 Booleans.html:13:12
true false …507 Booleans.html:14:12

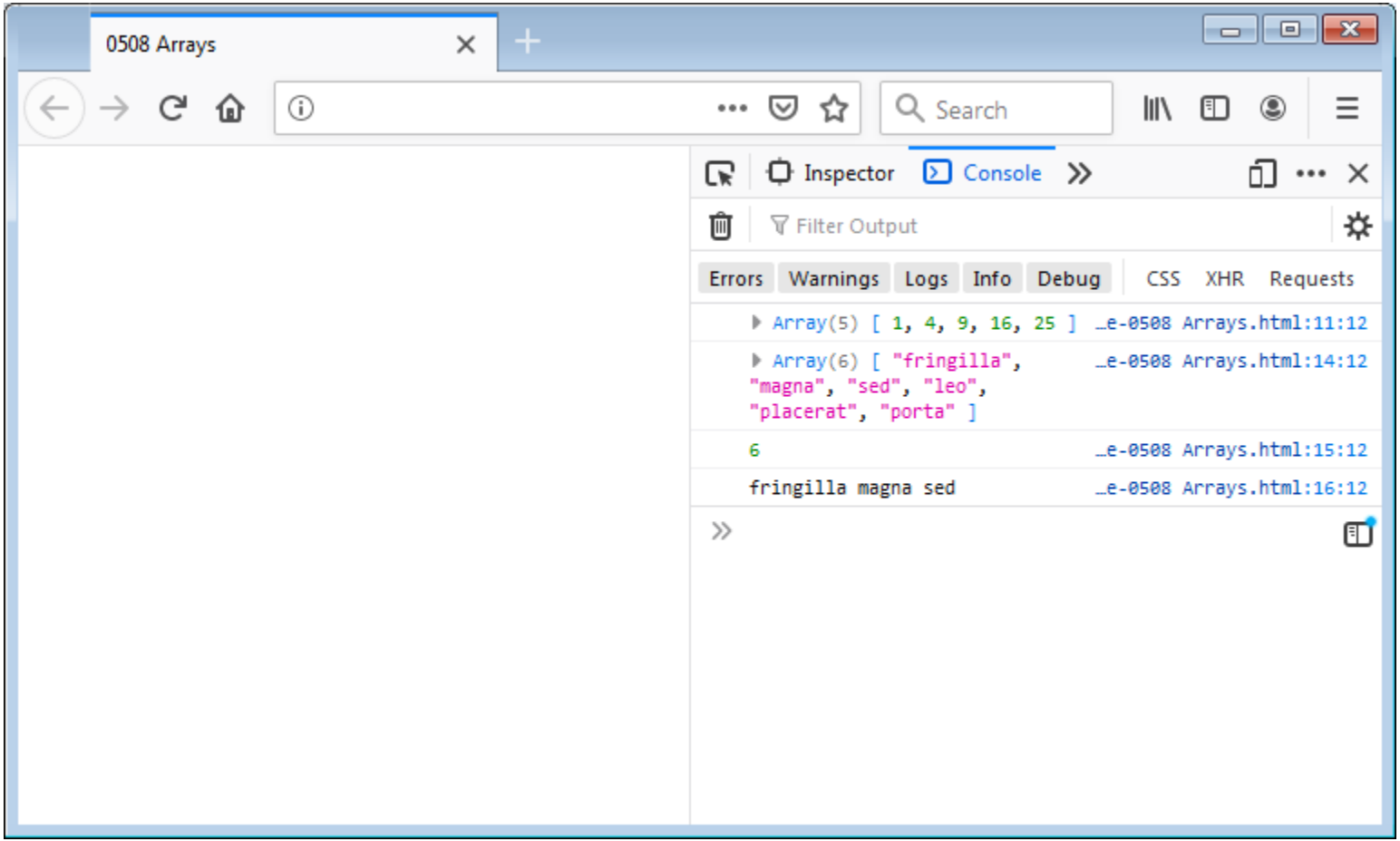
TRY IT

Complex data types

Arrays

- Access to elements with [...]
- Indices start from 0, number of elements is length
- Array elements could be of different data types

```
a = [1,4,9,16,25];
console.log(a);
b = ['fringilla','magna','sed','leo'];
console.log(b);
console.log(b.length);
console.log(b[0],b[1],b[2]);
```


0508 Arrays
Search
Inspector
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
Array(5) [1, 4, 9, 16, 25] …e-0508 Arrays.html:11:12
Array(6) ["fringilla", "magna", "sed", "leo", "placerat", "porta"] …e-0508 Arrays.html:14:12
6 …e-0508 Arrays.html:15:12
fringilla magna sed …e-0508 Arrays.html:16:12

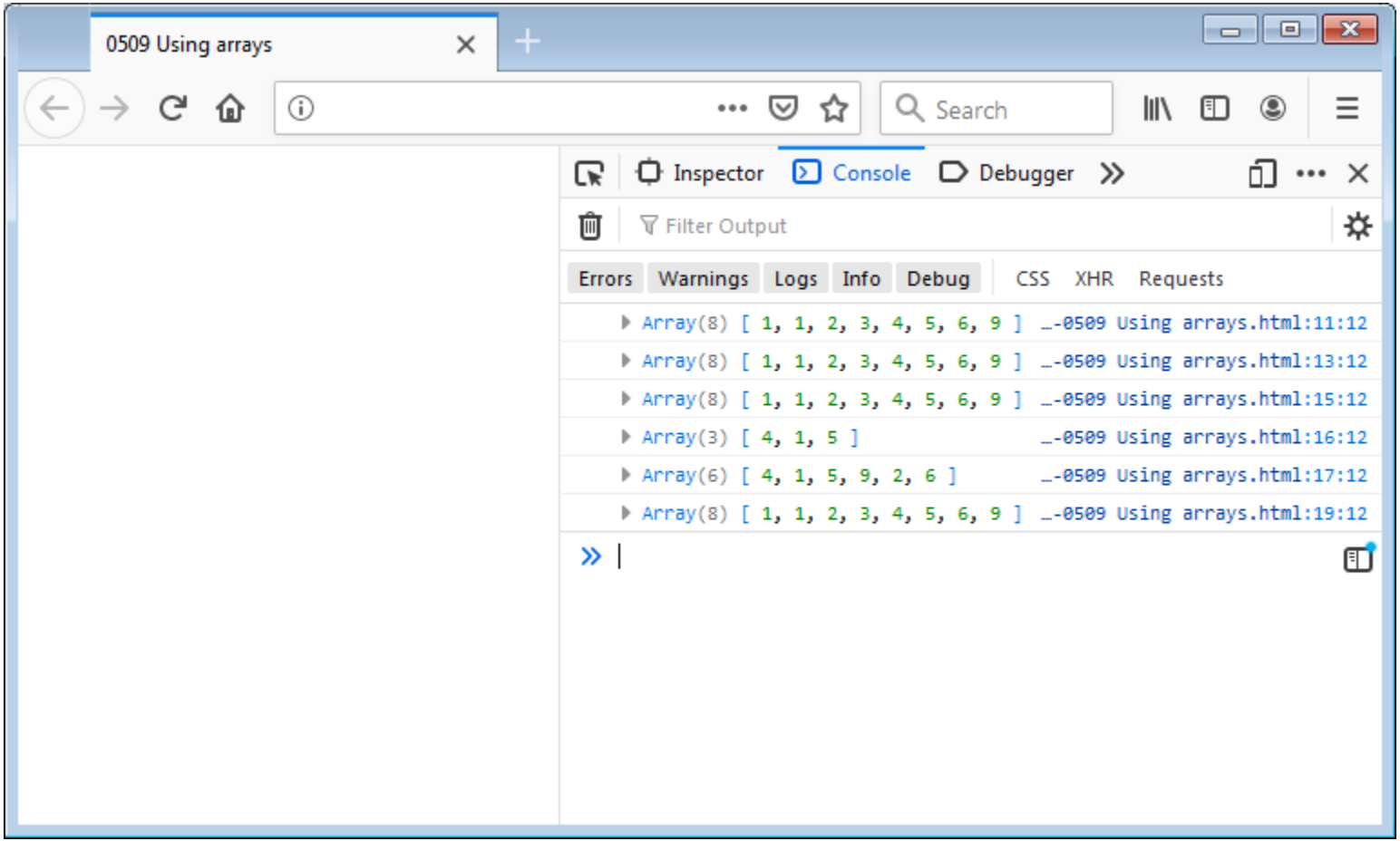
TRY IT

Using arrays

- Adding elements to the end with push
- Sorting with sort
- Extracting subarray with slice
  - from index to another index (exclusive)
  - from index to the end

From index 2 to index 4 (inclusive)

```
a = [3,1,4,1,5];
a.push(9,2);
a.push(6);
console.log(a.slice(2,5));
console.log(a.slice(2));
a.sort();
```


0509 Using arrays
Search
Inspector
Console
Debugger
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
Array(8) [1, 1, 2, 3, 4, 5, 6, 9] …-0509 Using arrays.html:11:12
Array(8) [1, 1, 2, 3, 4, 5, 6, 9] …-0509 Using arrays.html:13:12
Array(8) [1, 1, 2, 3, 4, 5, 6, 9] …-0509 Using arrays.html:15:12
Array(3) [4, 1, 5] …-0509 Using arrays.html:16:12
Array(6) [4, 1, 5, 9, 2, 6] …-0509 Using arrays.html:17:12
Array(8) [1, 1, 2, 3, 4, 5, 6, 9] …-0509 Using arrays.html:19:12

TRY IT

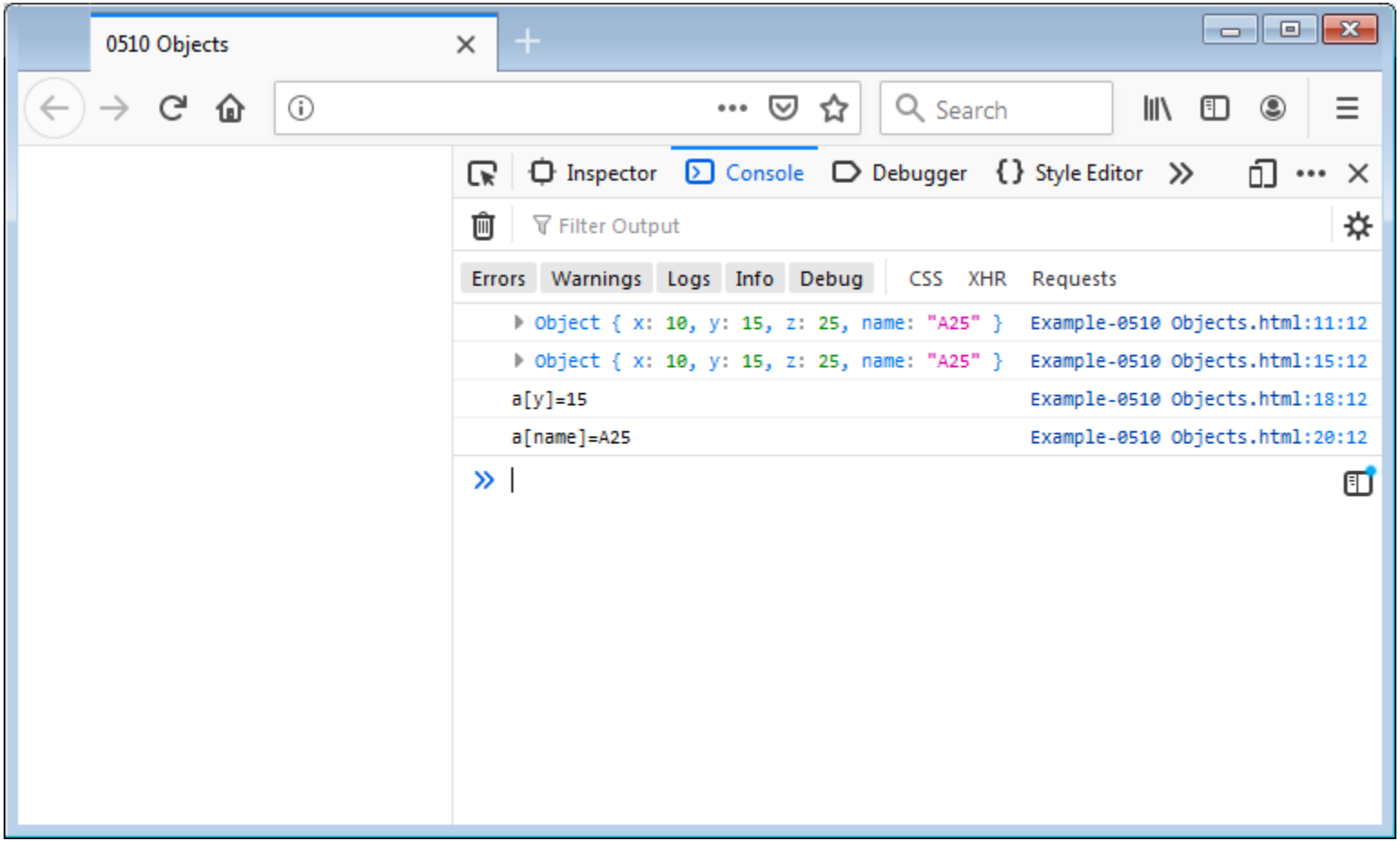
Objects

- Constants {name:value, name:value, … }
- Access to elements with obect.name or object[name]

```
a = {x:10, y:15};
console.log(a);

a.z = a.x+a.y;
a.name = "A"+a.z;
console.log(a);

i = 'name';
console.log('a['+i+']='+a[i]);
```


0510 Objects
Search
Inspector
Console
Debugger
Style Editor
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
Object { x: 10, y: 15, z: 25, name: "A25" }
Example-0510 Objects.html:11:12
Object { x: 10, y: 15, z: 25, name: "A25" }
Example-0510 Objects.html:15:12
a[y]=15
Example-0510 Objects.html:18:12
a[name]=A25
Example-0510 Objects.html:20:12

TRY IT

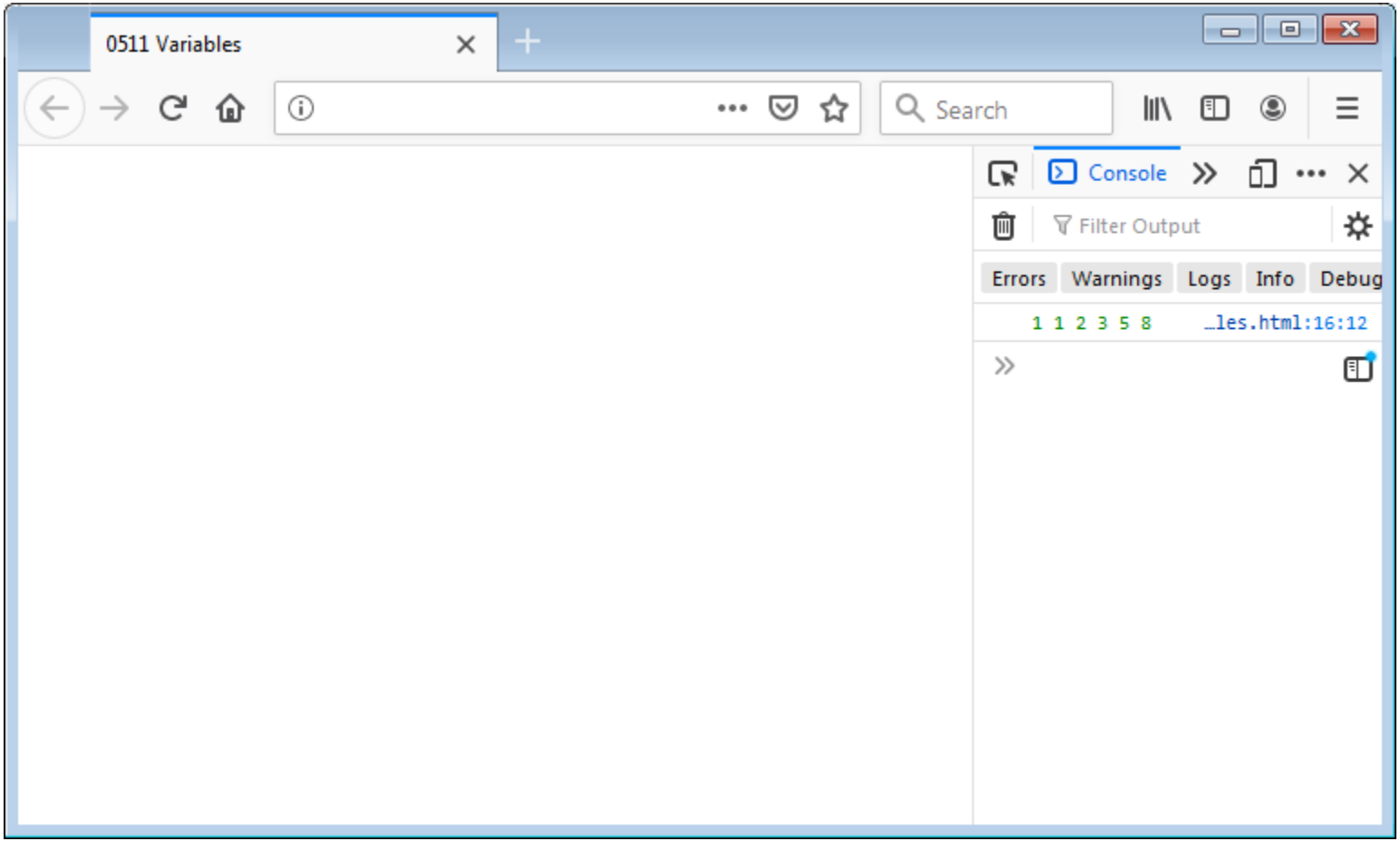
Syntax

Variables

Variables in JS

- Could change the type of their values
- There are local and global variables
- Local variable is created with var name;
- Value is assigned with name = value;
- If a variable does not exist, it is created as global

```
x1 = 1;
x2 = 1;
x3 = x1+x2;
```


0511 Variables
Search
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
1 1 2 3 5 8
…les.html:16:12

TRY IT

If

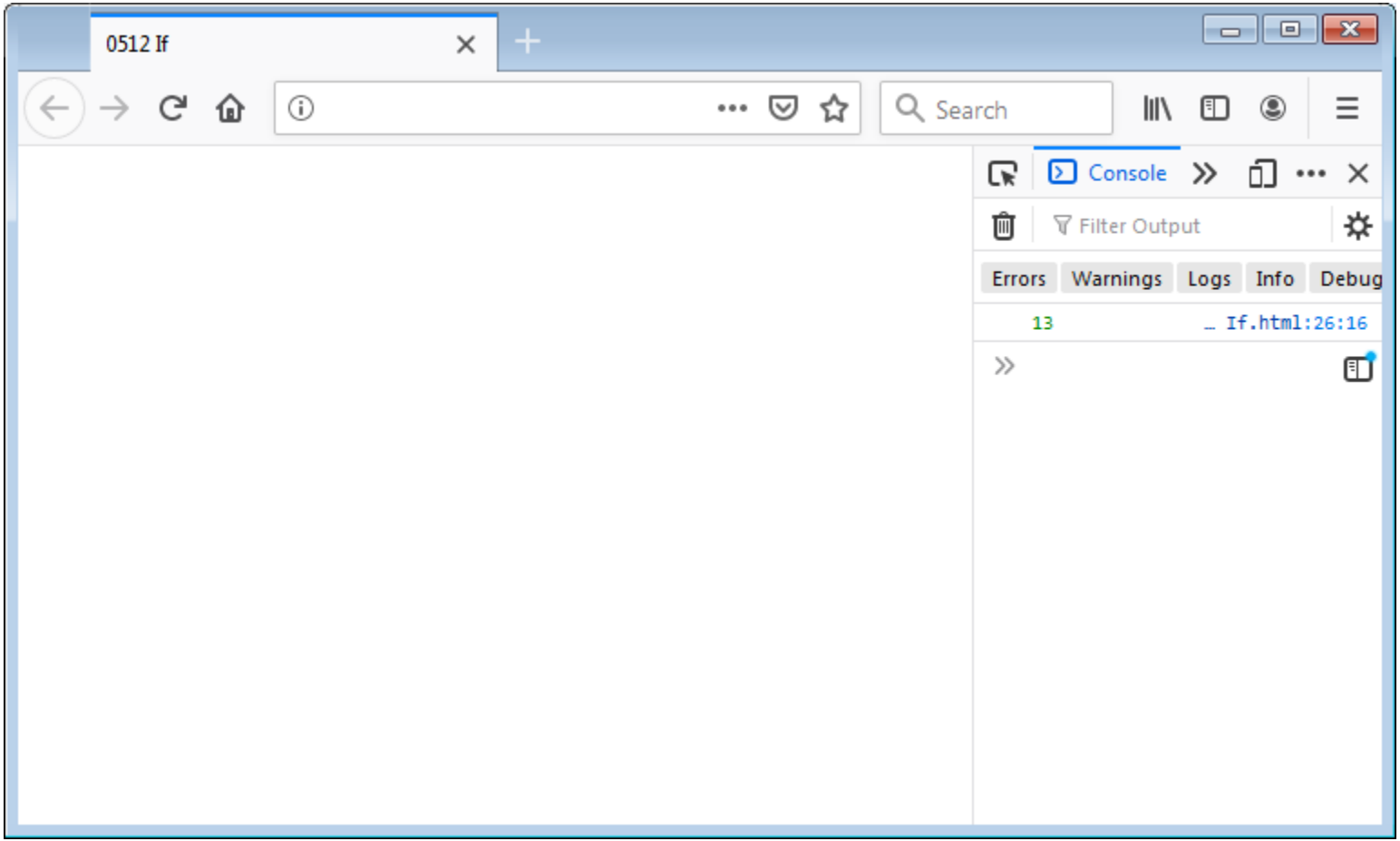
- As a command

  if (condition) {commands}

  if (condition) {commands} else {commands}

  if (condition) {commands} else if (condition) {commands} …

```
if (x>y)
{
   if (x>z)
      { console.log(x); }
   else
      { console.log(z); }
}
```


0512 If
Search
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
13
If.html:26:16

TRY IT

Attention!

- All data types can be used as boolean values
- Examples for data, which is considered false:
  - The actual value false
  - Numerical zero 0
  - Empty string ""
  - Undefined variable undefined
  - Empty value null
  - Not-a-number values NaN
- Convertion to boolean values is done automatically in command if, function Boolean и operator !

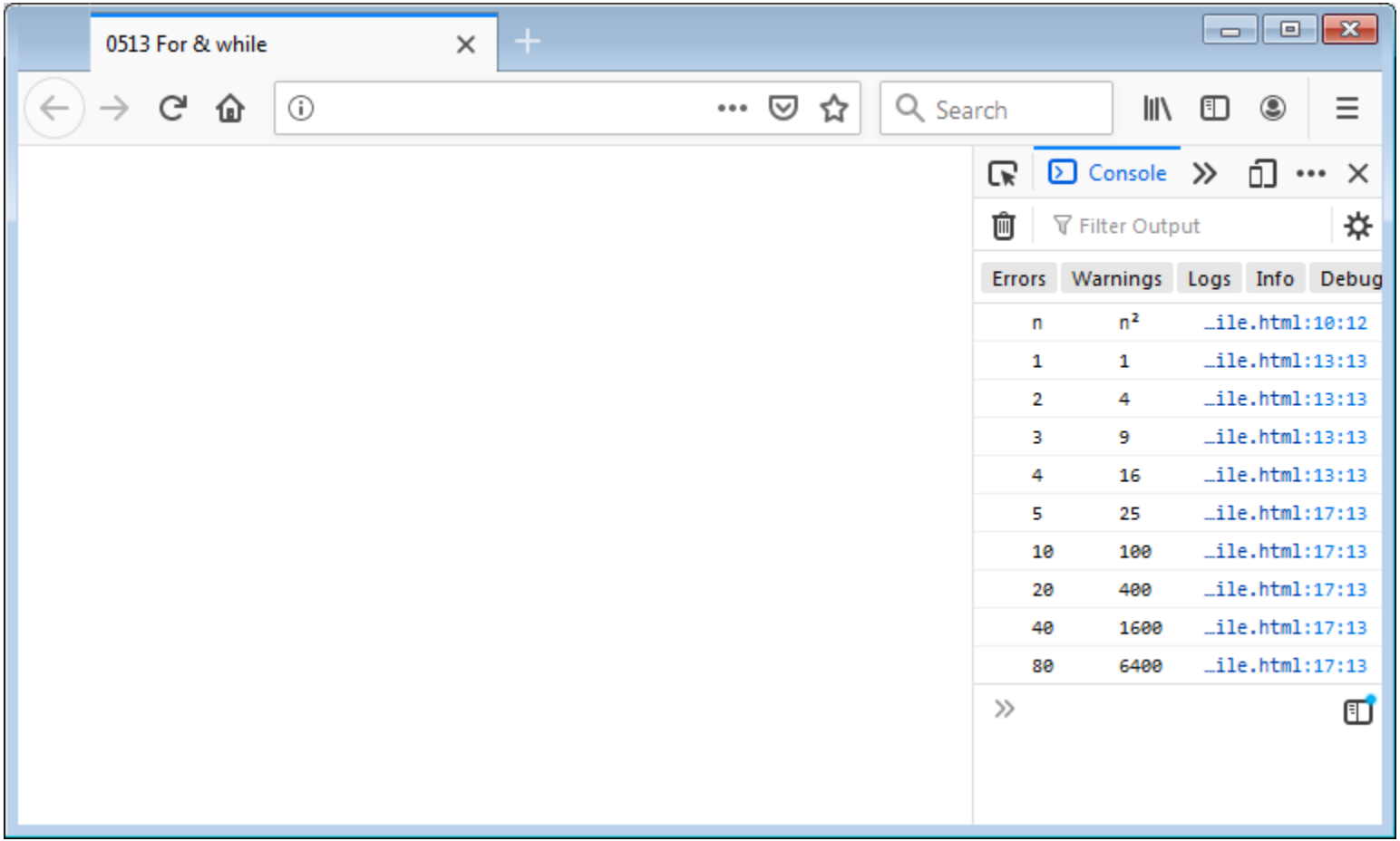
Cycle

- Standard cycle with initial values, condition and increment
  for (initial; condition; step) {commands}

```
for (var i=1; i<5; i++)
{
   console.log(i+'\t'+i*i);
}
```

- Condition checked on every step while (condition) {commands}

```
while (i<100)
{
   console.log(i+'\t'+i*i);
   i = 2*i;
}
```

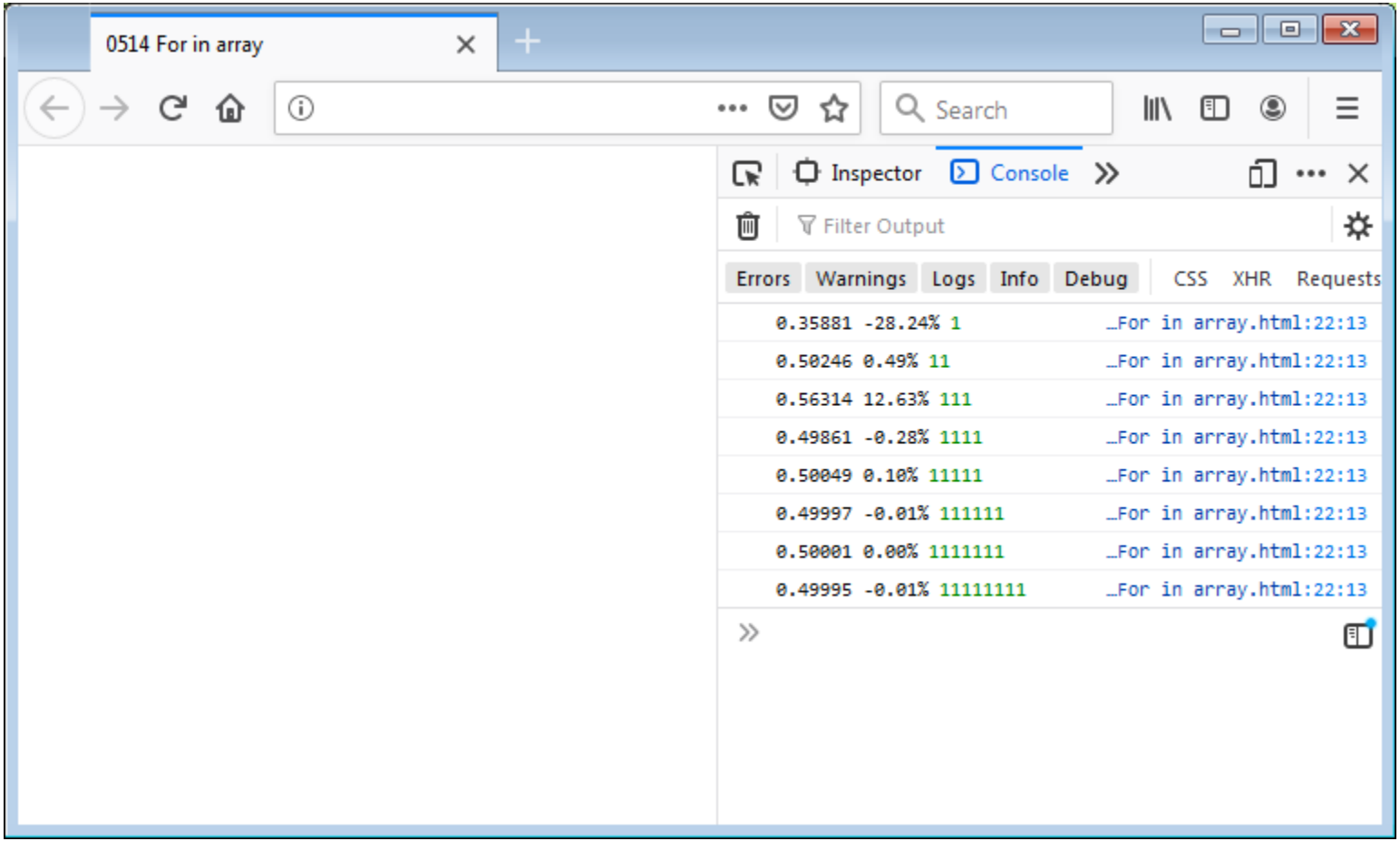

0513 For & while
Search
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
n n² …ile.html:10:12
1 1 …ile.html:13:13
2 4 …ile.html:13:13
3 9 …ile.html:13:13
4 16 …ile.html:13:13
5 25 …ile.html:17:13
10 100 …ile.html:17:13
20 400 …ile.html:17:13
40 1600 …ile.html:17:13
80 6400 …ile.html:17:13

TRY IT

- Traversal with for (variable in array) {…}
- Variables gets indices as its value
- Example for calculating the average of n number in an array

```
a = [];
for (n=1; n<100000000; n*=10)
{
   for (var i=0; i<n; i++) a.push(Math.random());
   avg = 0;
   for (var i in a) avg+=a[i]/a.length;
   console.log(avg);
}
```

- In for (variable in object) {…} the variable gets object's elements names as values


0514 For in array
Search
Inspector
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
0.35881 -28.24% 1 …For in array.html:22:13
0.50246 0.49% 11 …For in array.html:22:13
0.56314 12.63% 111 …For in array.html:22:13
0.49861 -0.28% 1111 …For in array.html:22:13
0.50049 0.10% 11111 …For in array.html:22:13
0.49997 -0.01% 111111 …For in array.html:22:13
0.50001 0.00% 1111111 …For in array.html:22:13
0.49995 -0.01% 11111111 …For in array.html:22:13

TRY IT

Functions

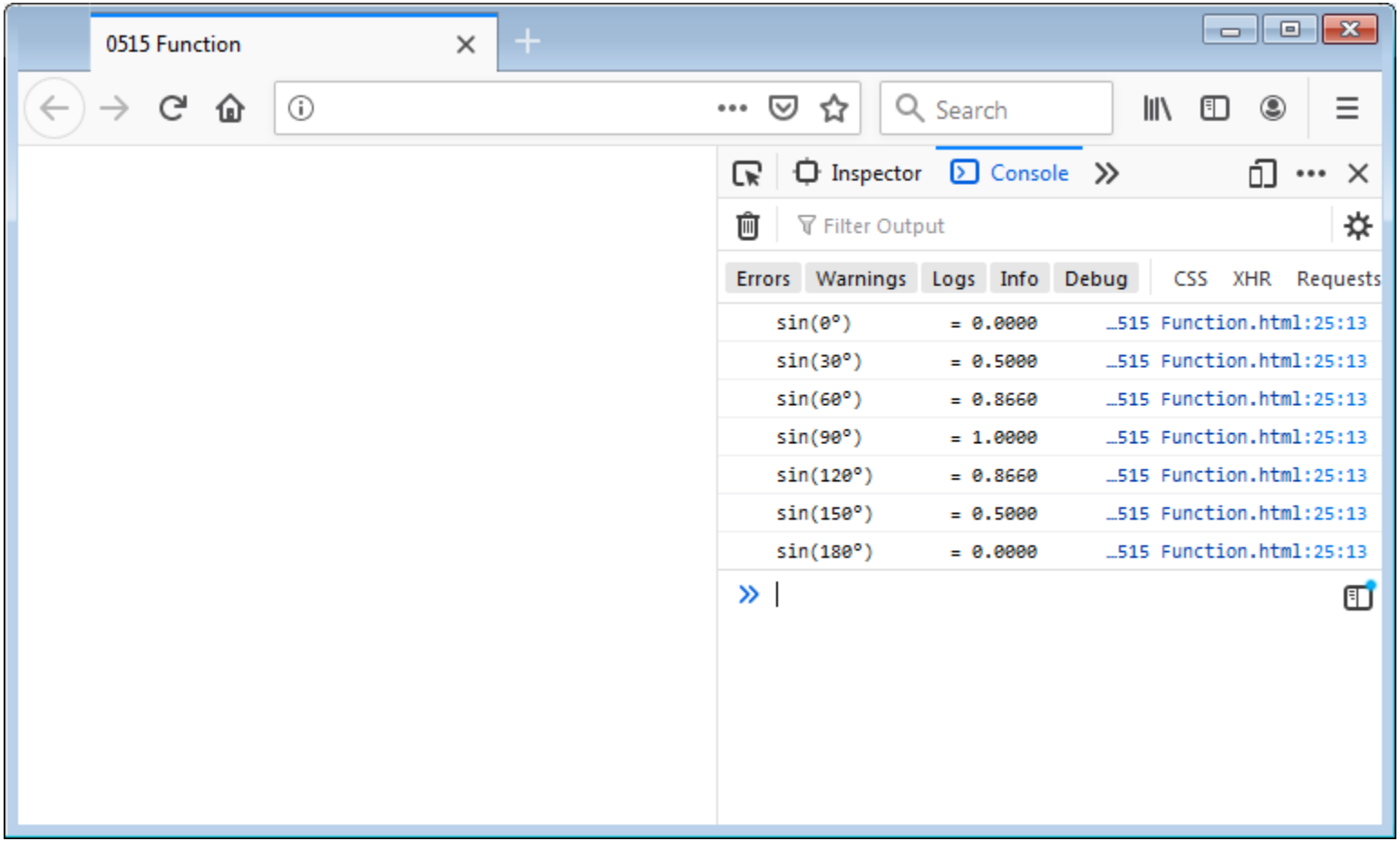
Functions in JS

Function definition

- Reserved word function
- No function type and parameters types
- Result is returned with return

```
function rad(x) { return x*Math.PI/180; }

function sin(x)
{  var r = rad(x);
   return Math.sin(r);
}
```

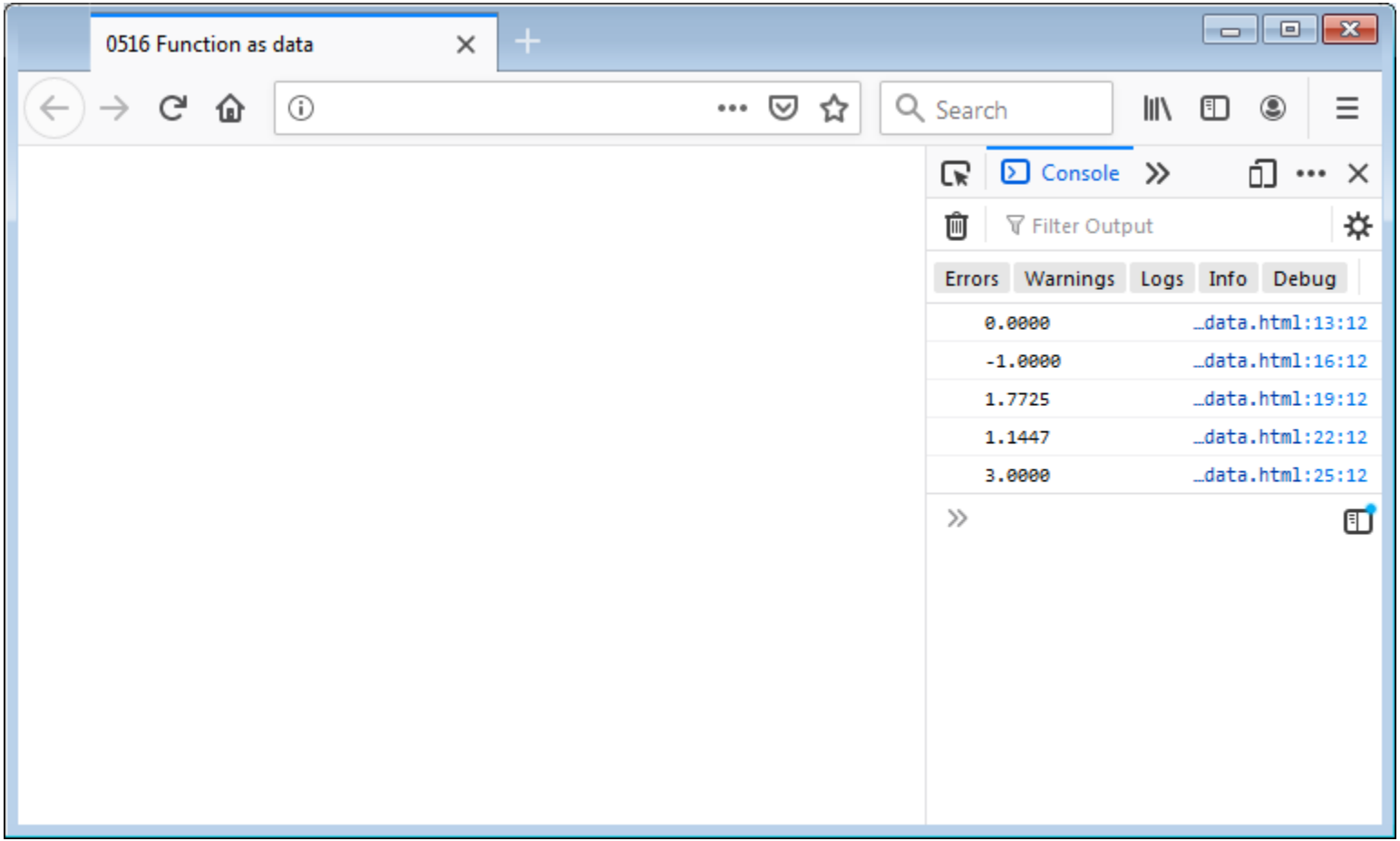

0515 Function
Search
Inspector
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
sin(0°) = 0.0000 …515 Function.html:25:13
sin(30°) = 0.5000 …515 Function.html:25:13
sin(60°) = 0.8660 …515 Function.html:25:13
sin(90°) = 1.0000 …515 Function.html:25:13
sin(120°) = 0.8660 …515 Function.html:25:13
sin(150°) = 0.5000 …515 Function.html:25:13
sin(180°) = 0.0000 …515 Function.html:25:13

TRY IT

Functions as data

- No pointers in JS
- Function without (…) is data
- Could be assigned to variables or passed as parameters

```
x = Math.PI;
fun = Math.sin;
console.log(fun(x));
```


0516 Function as data
Search
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
0.0000 _data.html:13:12
-1.0000 _data.html:16:12
1.7725 _data.html:19:12
1.1447 _data.html:22:12
3.0000 _data.html:25:12

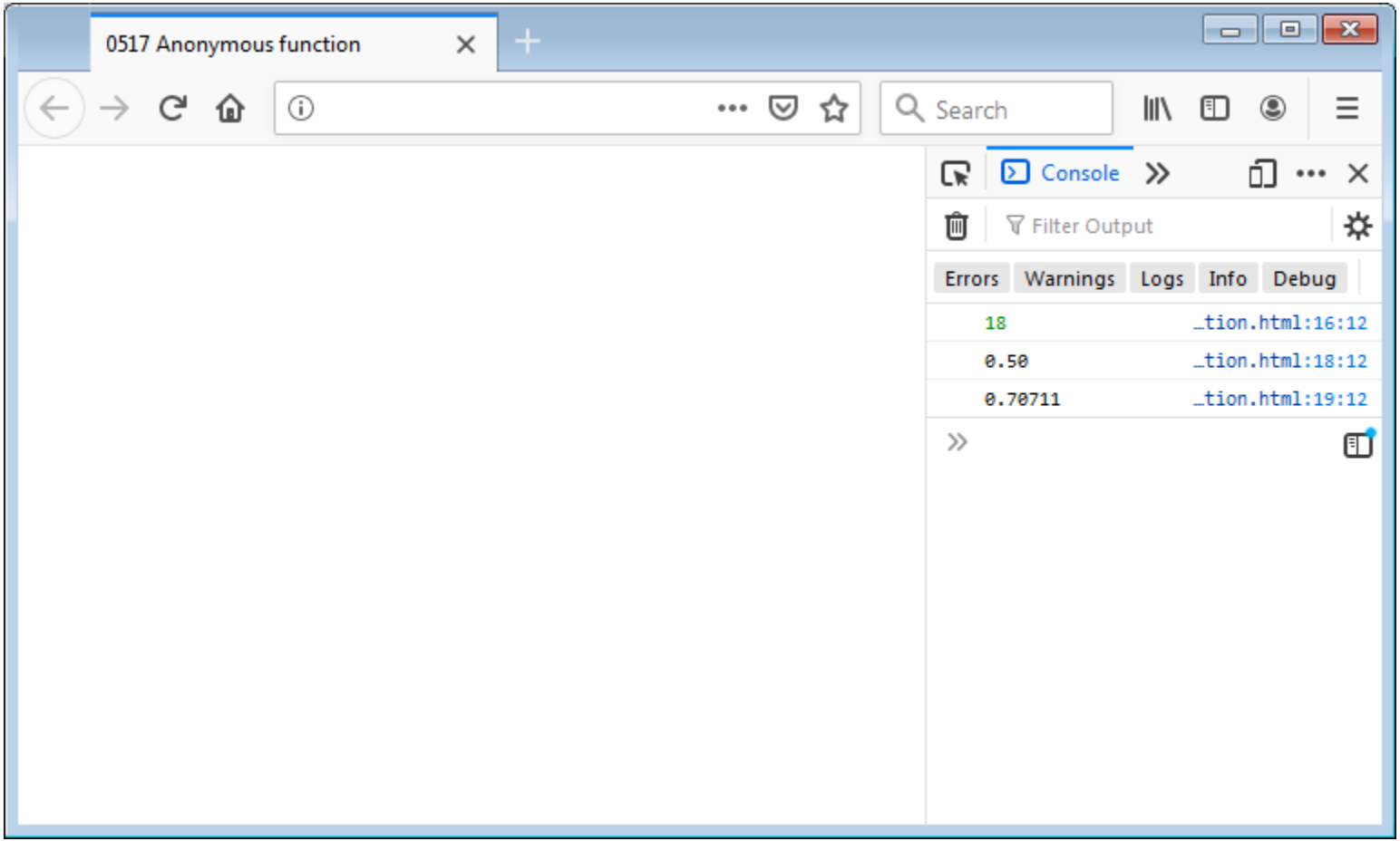
TRY IT

Anonymous functions

- Function with parameters and body, but without name
- Often used when a function is assigned to data

```
fun = function(a,b) { return (a-1)*(b+1); };
console.log(fun(4,5));

function calc(fun,arg,n)
{
   return fun(arg).toFixed(n);
}
console.log(
   calc( function(x){return 1/Math.sqrt(x);},4,2 )
);
```


0517 Anonymous function
Search
Console
Filter Output
Errors
Warnings
Logs
Info
Debug
18
_tion.html:16:12
0.50
_tion.html:18:12
0.70711
_tion.html:19:12

TRY IT

Summary

JavaScript

Language JavaScript

- Similar to C
- Used in a browser

Data types

- Simple: numbers, strings, booleans
- Complex: arrays, objects, functions

Variables and functions

- No type, only data have types
- Anonymous functions

More

Additional information

- Here: http://www.w3schools.com/js/
- There: http://www.w3schools.com/jsref

The end

Comments, questions