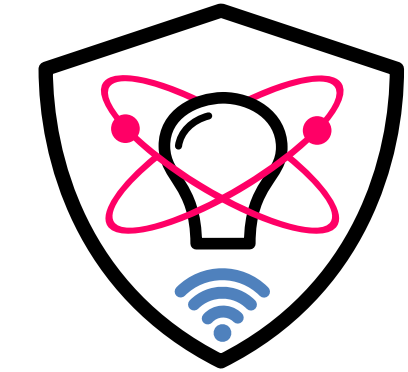


JavaScript

Тема №5

Обща информация

JavaScript

Какво е JavaScript

- Език за програмиране, често се изписва само като JS
- Създаден за динамични уеб страници и уеб приложения
- По традиция програмите на JS се наричат скриптове (в миналото е имало само интерпретатори за JS)
- JS няма общо с Java
- Може да се изпълнява на сървър, може и в браузър

История на JS

История

- 1995 – В Netscape Navigator 2.0 (първоначално JS се е казвал Mocha, после LiveScript и накрая JavaScript)
- 1996 – Стандартизиране на JS като ECMAScript
- 1997 – Първа версия
- 1998 – Втора версия
- 1999 – Трета версия
- 2009 – Пета версия (няма четвърта)
- 2011 – Текуща версия 5.1 на JS

Какво може JS

Възможности

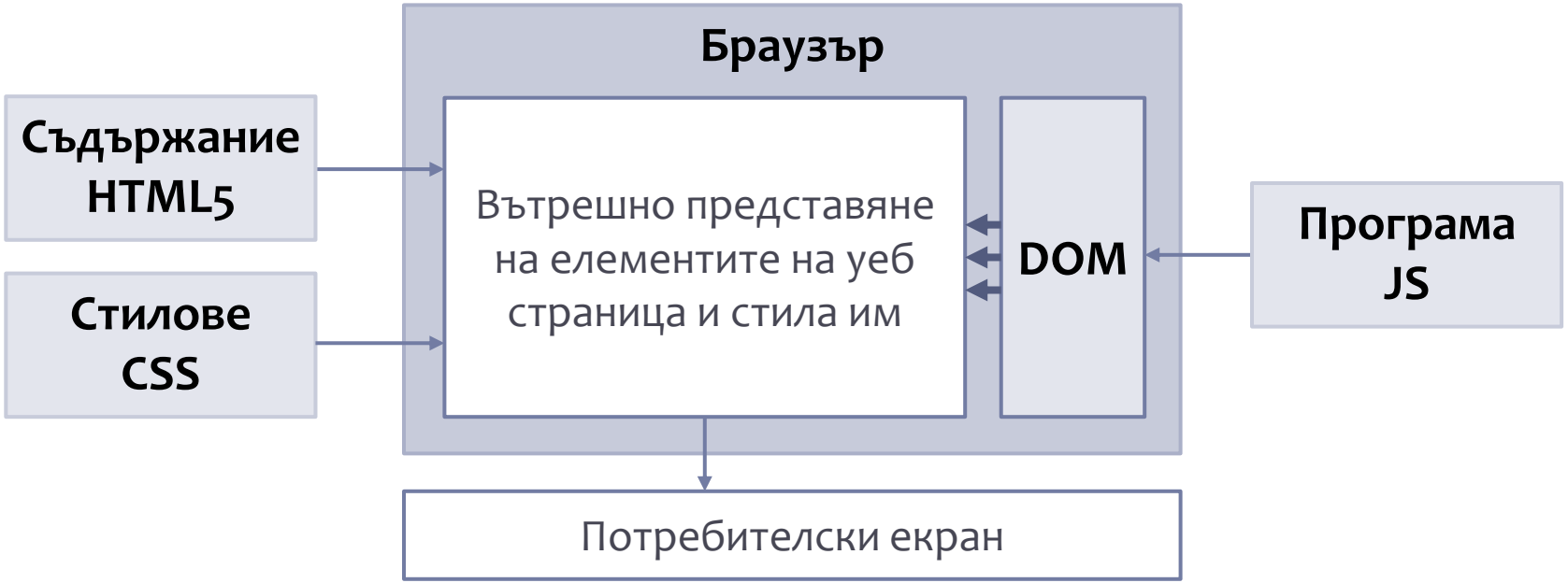
- Език с общо предназначение
- Процедурно, функционално и обектно програмиране
- Промяна на елементите и стиловете на уеб страница
- Обработване на движение на мишката, кликване на бутон, …

В курса СУИКА

- Програмите ни ще са на JS
- Ще ползват библиотека Suica, написана на JS

Връзка с HTML и CSS

- Достъп до свойствата и стиловете на елементите на уеб страница през DOM (Document Object Model)



Как изглежда

Външен вид на JS код

- Синтактично подобен на C
- Без указатели
- Без типове на променливите

Използване

Разположение

- На сървъра (server-side JS) – няма да ползваме
- При клиента (client-side JS)

Използване

- Във външен файл, общ за целия сайт
- В скриптов елемент, общ за цялата страница
- В атрибут към конкретен HTML елемент
- Може да се разполага и на трите места едновременно

Работна среда

JS Конзола

- Модул към браузърите, помагащ на работата с JS

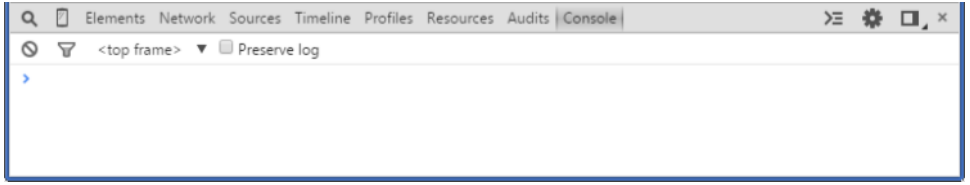
Ще я използваме за

- Съобщения за грешки
- Изход на временни резултати
- Изпълнение на нови команди (много рядко)

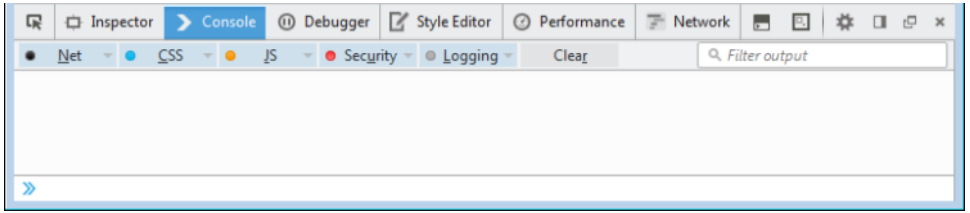
```
console.log(…);
```

Показване на конзола

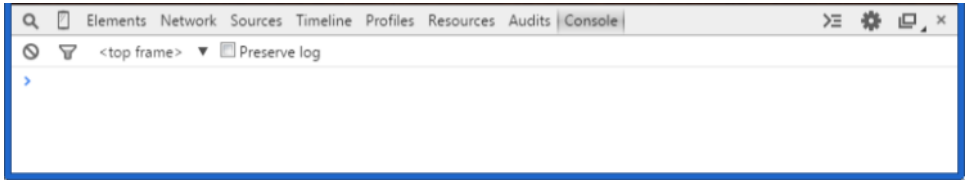
- Chrome и Ctrl-Shift-I . . . . .



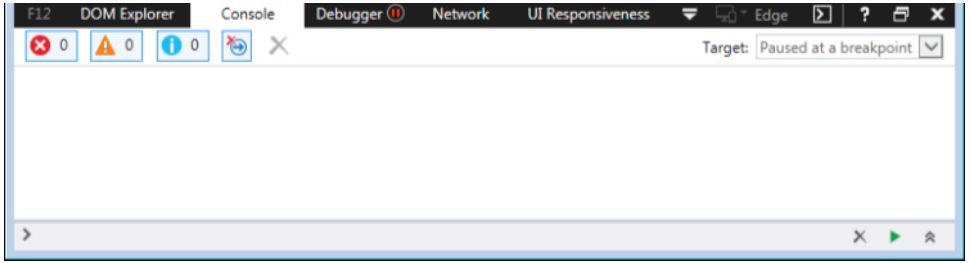
- FireFox и Ctrl-Shift-I . . . . .



- Opera и Ctrl-Shift-I . . . . . .



- Internet Explorer и F12 . . .

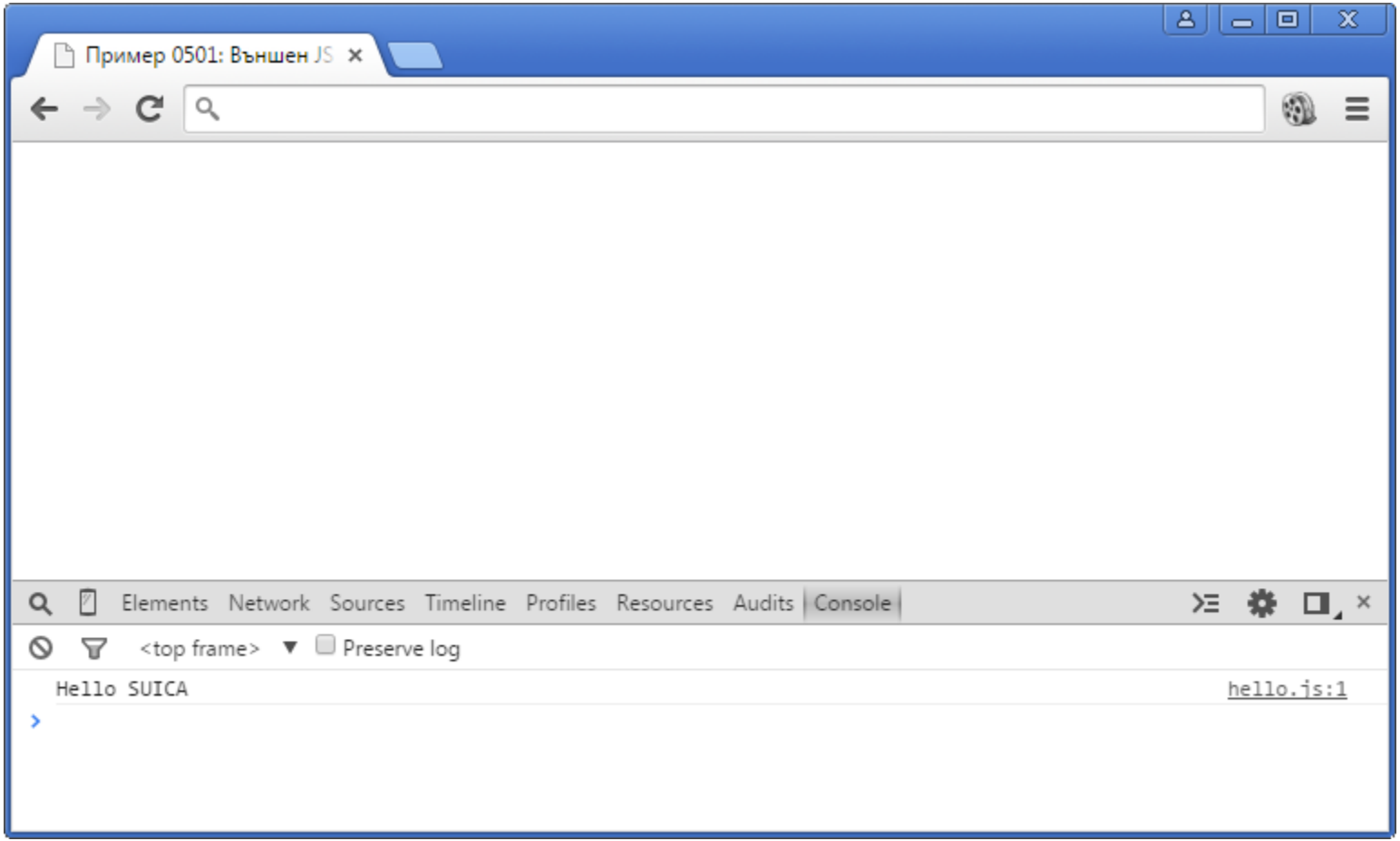


Външен JS

JS във външни файлове

- Споделяне на JS код между различни страници
- Използва се елемент <script> поставен в <head> или <body>, като атрибута src указва пътя и името до файла с JS код
- Елементът <script> трябва да се затвори със </script>
- Традиционно разширение .js

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


Пример 0501: Външен JS
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
Hello SUICA
hello.js:1

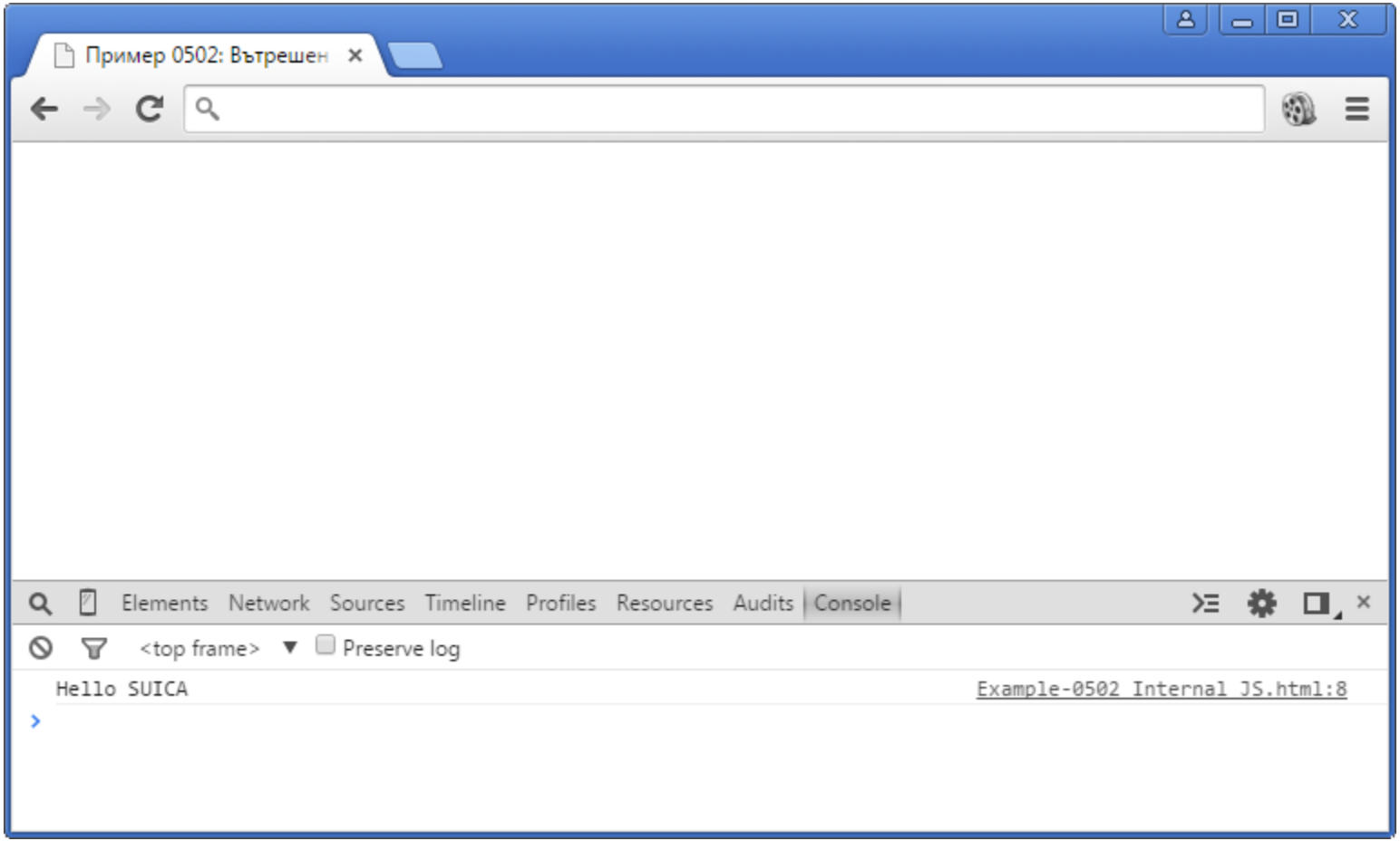
ПРОБА

Вътрешен JS

JS в елемента <script>

- Може да е в <head> или <body>

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


Пример 0502: Вътрешен
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
Hello SUICA
Example-0502 Internal JS.html:8

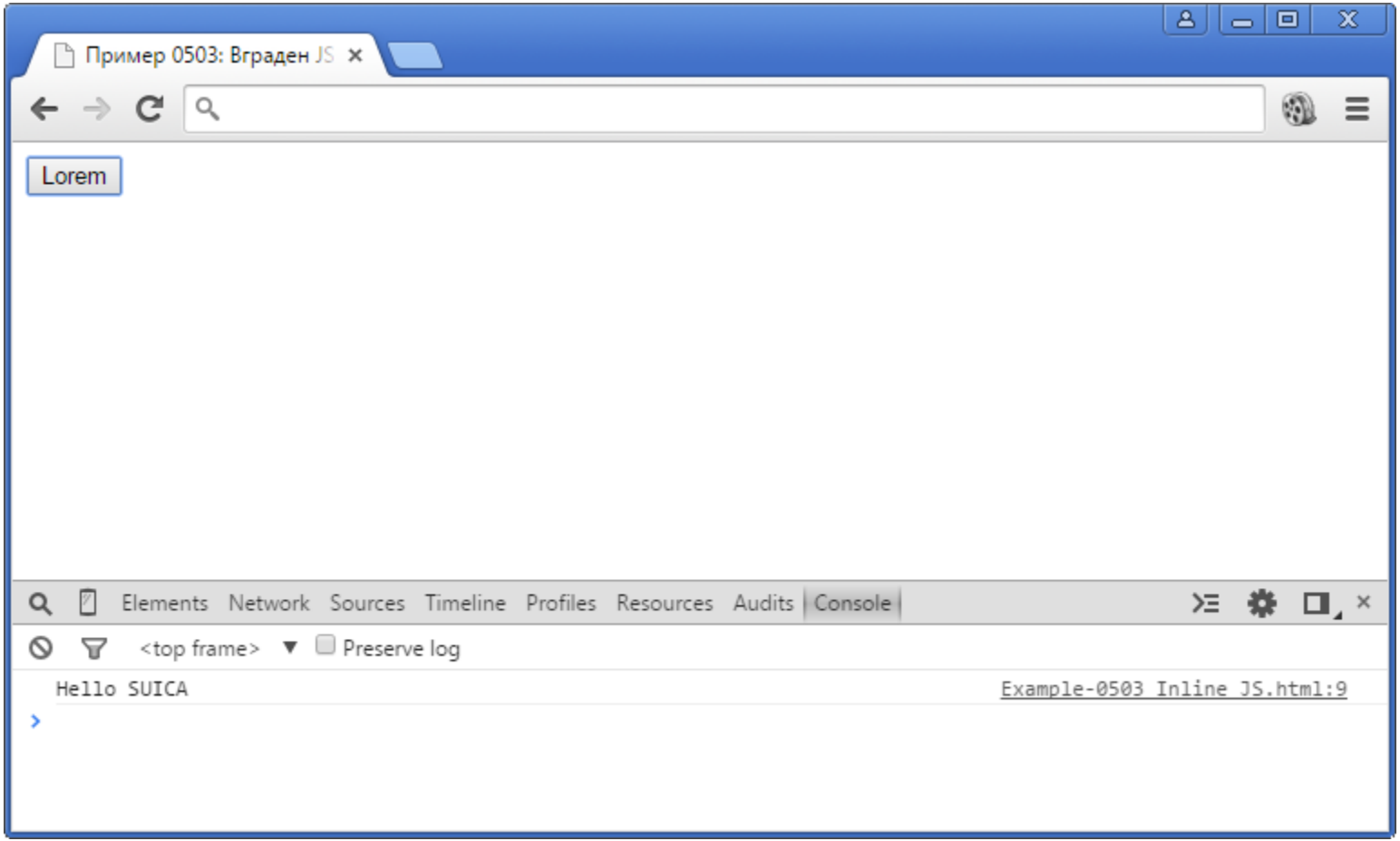
ПРОБА

Вграден JS

JS в атрибута на HTML елемент

- Удобни за атрибути, реагиращи на действия на потребителя, като кликване на бутон

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


Пример 0503: Вграден JS
Lorem
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
Hello SUICA
Example-0503 Inline JS.html:9

ПРОБА

Типове данни

Типове данни

Прости типове

- Числови
- Текстови
- Булеви

Съставни типове

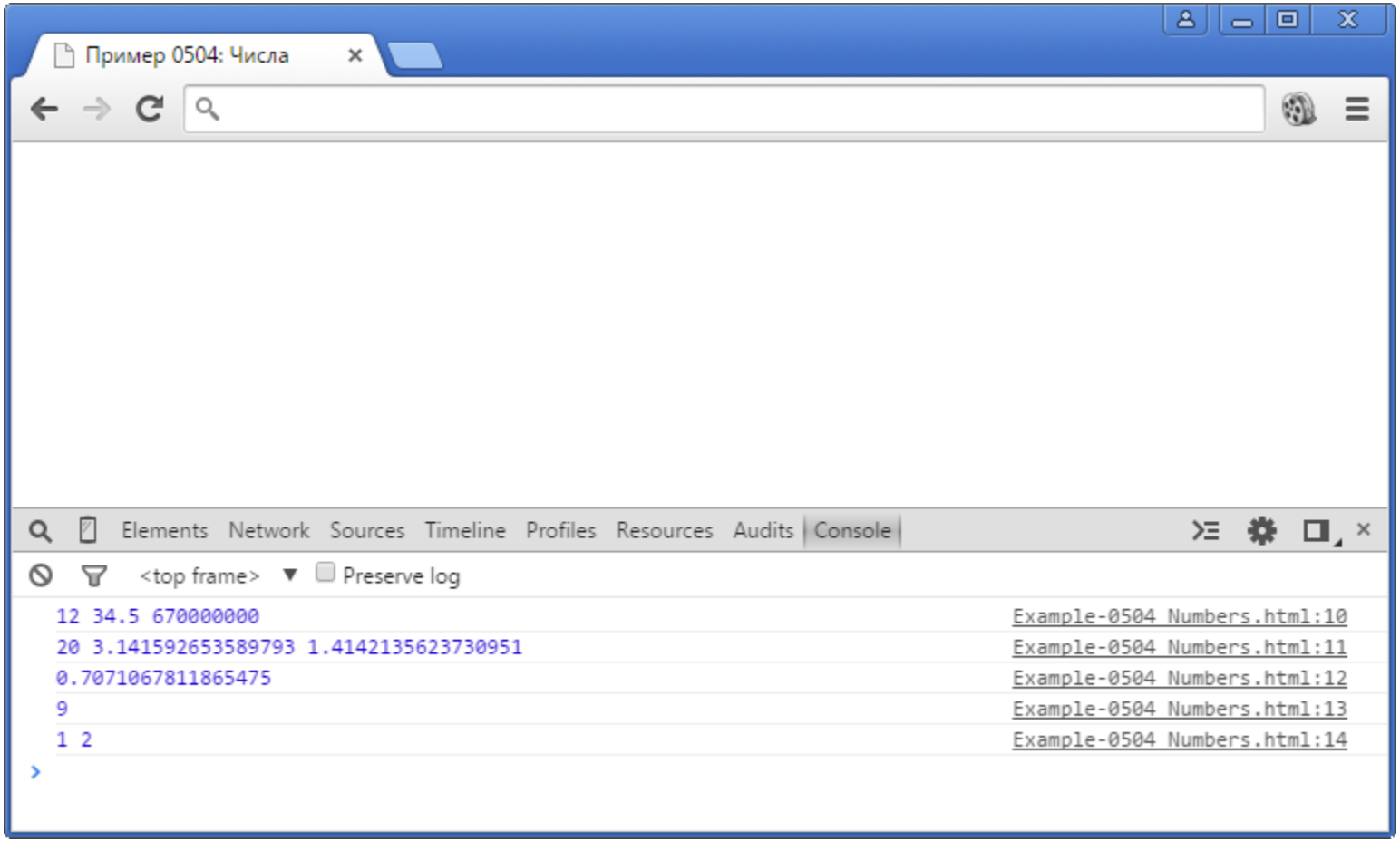
- Масиви
- Обекти
- Функции

Прости типове данни

Числови данни

- Няма разделение на цели и дробни
- Поддържат се стандартните операции
- Числовите функции и константи са в 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) );
```

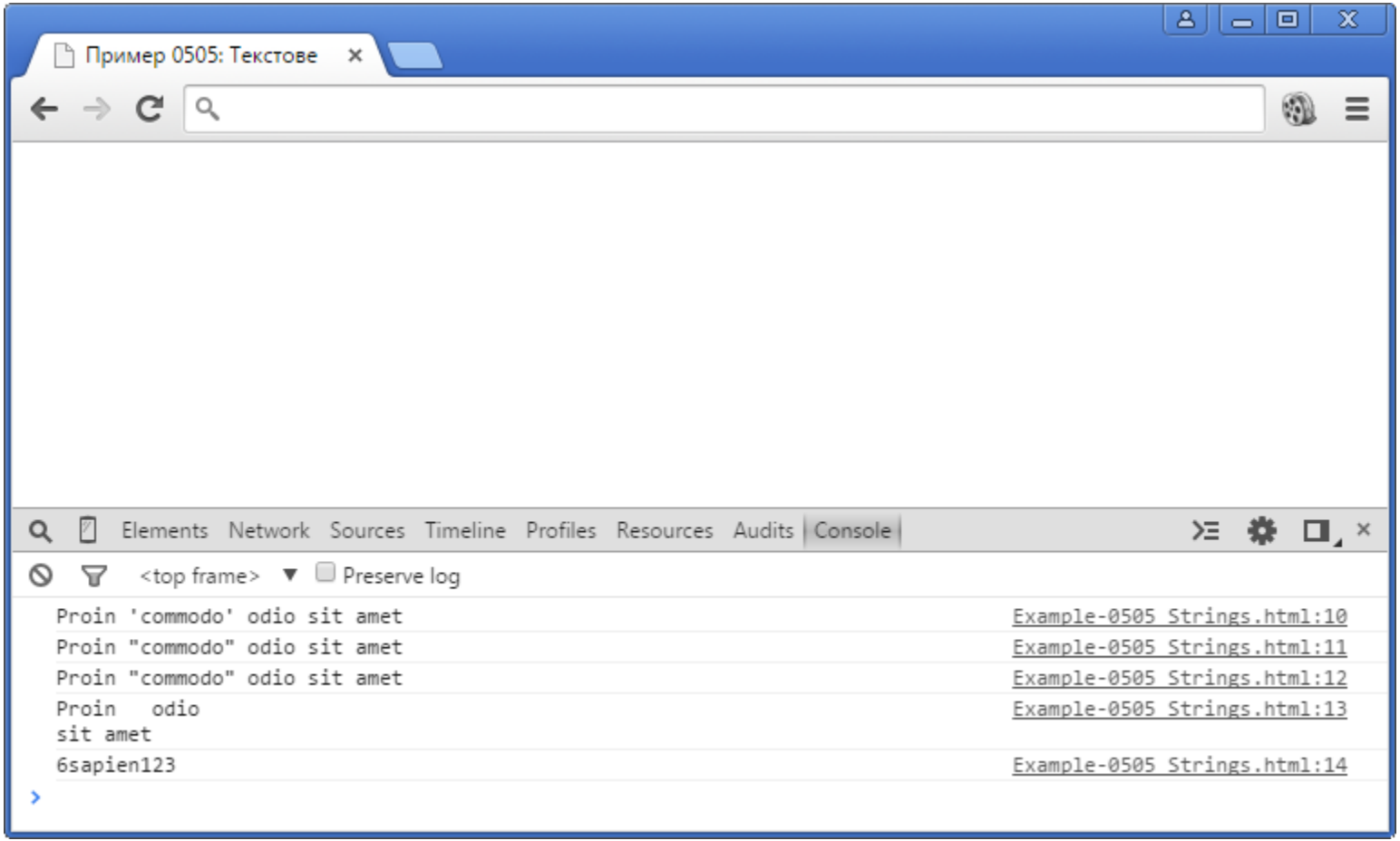

Пример 0504: Числа
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
12 34.5 670000000
Example-0504 Numbers.html:10
20 3.141592653589793 1.4142135623730951
Example-0504 Numbers.html:11
0.7071067811865475
Example-0504 Numbers.html:12
9
Example-0504 Numbers.html:13
1 2
Example-0504 Numbers.html:14

ПРОБА

Текстови данни (низове, стрингове)

- Константите се заграждат в кавички "…" или апострофи '…'
- Специалните символи се предхождат с \
- Поддържат се нов ред \n и табулация \t
- Автоматично конвертиране на число до стринг

```
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);
```

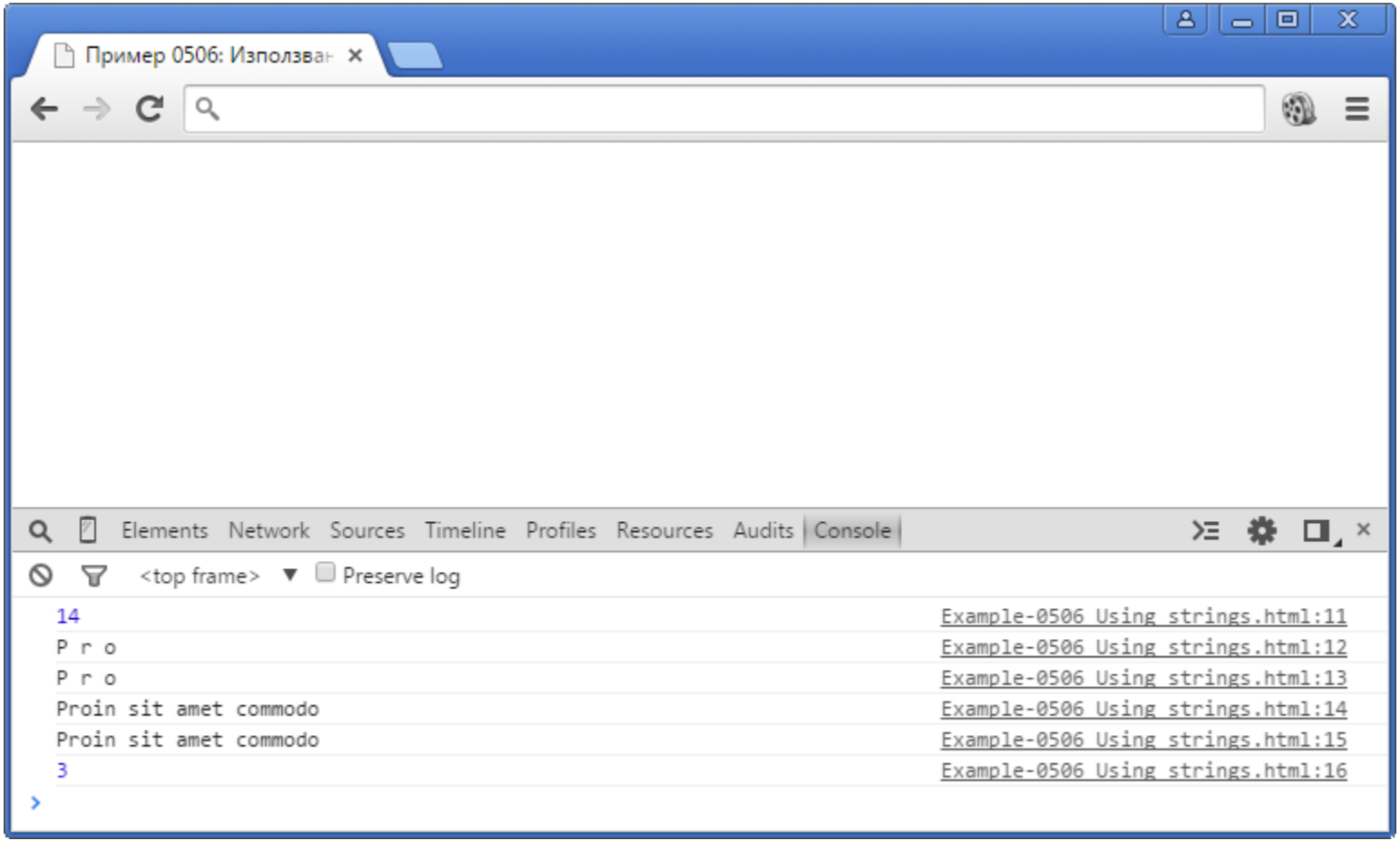

Пример 0505: Текстове
Elements Network Sources Timeline Profiles Resources Audits Console
<top frame>
Preserve log
Proin 'commodo' odio sit amet
Example-0505_Strings.html:10
Proin "commodo" odio sit amet
Example-0505_Strings.html:11
Proin "commodo" odio sit amet
Example-0505_Strings.html:12
Proin odio
sit amet
Example-0505_Strings.html:13
6sapien123
Example-0505_Strings.html:14

ПРОБА

Работа с текстови данни

- Свойство length за дължина
- Метод charAt или […] за вземане на символ
- Метод concat или + за конкатенация
- Методи indexOf и search за търсене

```
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'));
```

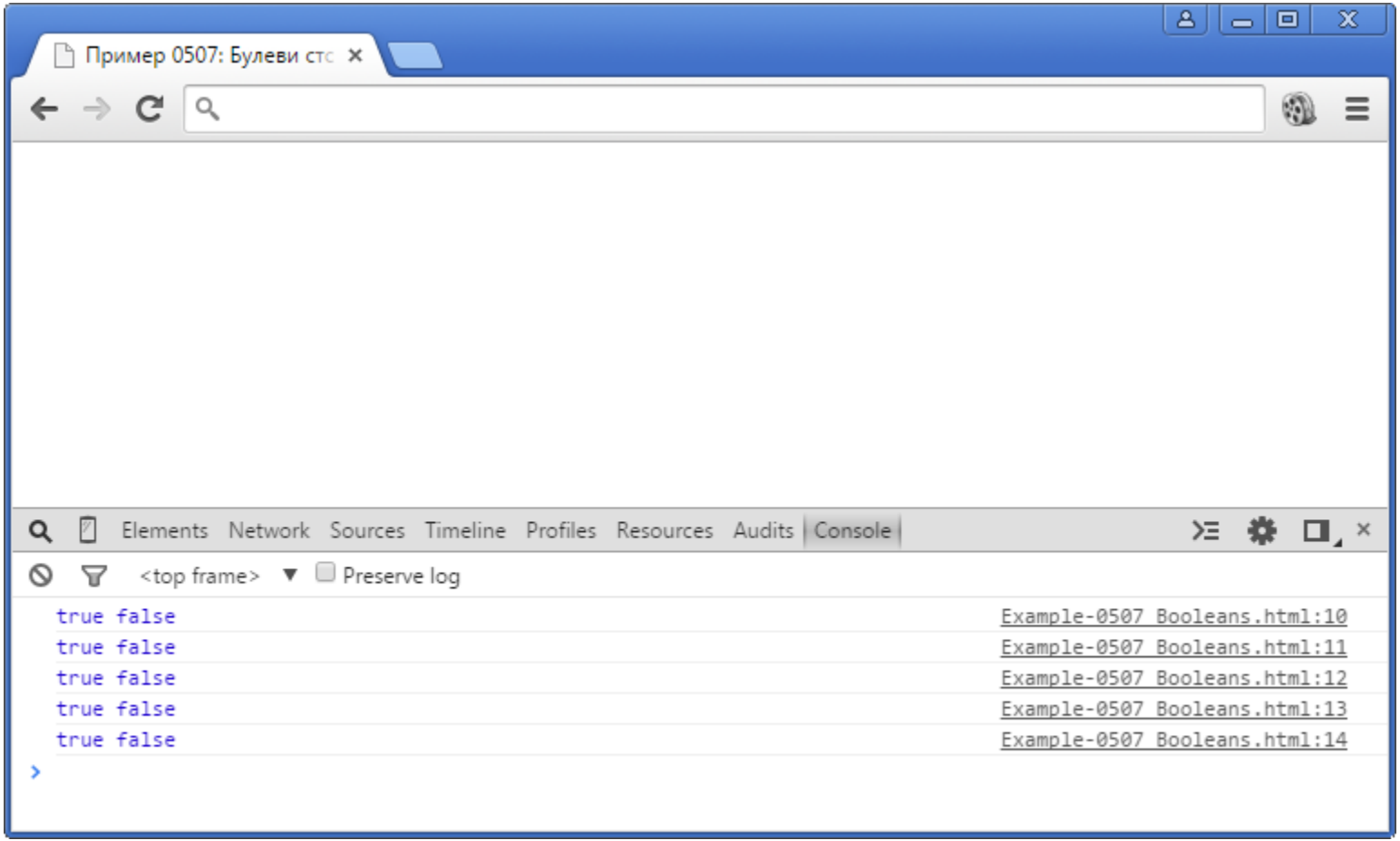

Пример 0506: Използван
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
14
Example-0506 Using strings.html:11
P r o
Example-0506 Using strings.html:12
P r o
Example-0506 Using strings.html:13
Proin sit amet commodo
Example-0506 Using strings.html:14
Proin sit amet commodo
Example-0506 Using strings.html:15
3
Example-0506 Using strings.html:16

ПРОБА

Булеви стойности

- Константи true и false
- Стандартни булеви оператори &&, ||, !
- Резултатите от сравнения с <, = и > са булеви
- Функция Boolean определя дали булева стойност би се възприела като true или 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(""));
```


Пример 0507: Булеви стс
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
true false
Example-0507 Booleans.html:10
true false
Example-0507 Booleans.html:11
true false
Example-0507 Booleans.html:12
true false
Example-0507 Booleans.html:13
true false
Example-0507 Booleans.html:14

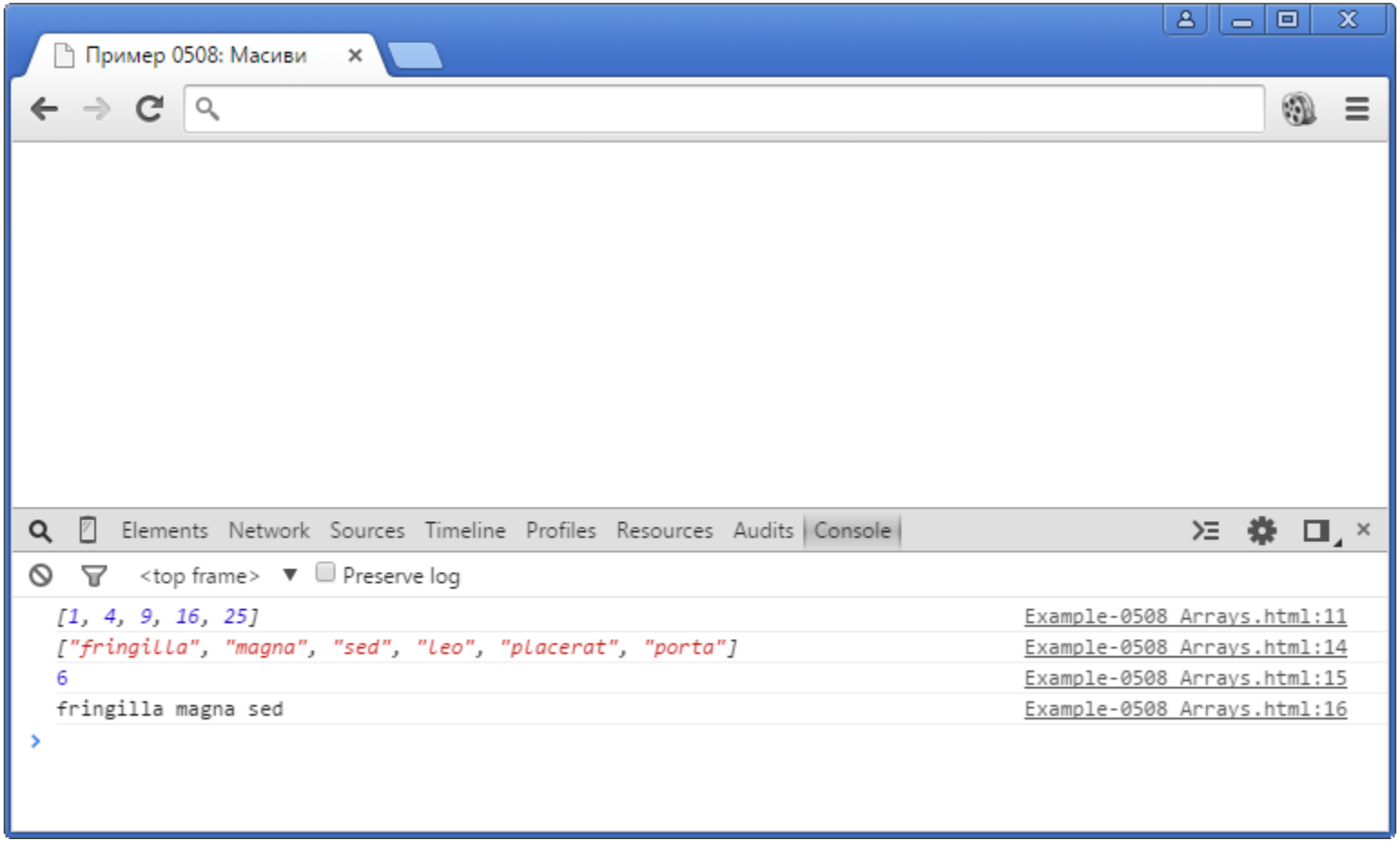
ПРОБА

Съставни типове

Масиви

- Достъпът до елементи е с [...]
- Индексите започват от 0, броят елементи е length
- Елементите могат да са от различни типове

```
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]);
```


Пример 0508: Масиви
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
[1, 4, 9, 16, 25]
Example-0508 Arrays.html:11
["fringilla", "magna", "sed", "leo", "placerat", "porta"]
Example-0508 Arrays.html:14
6
Example-0508 Arrays.html:15
fringilla magna sed
Example-0508 Arrays.html:16

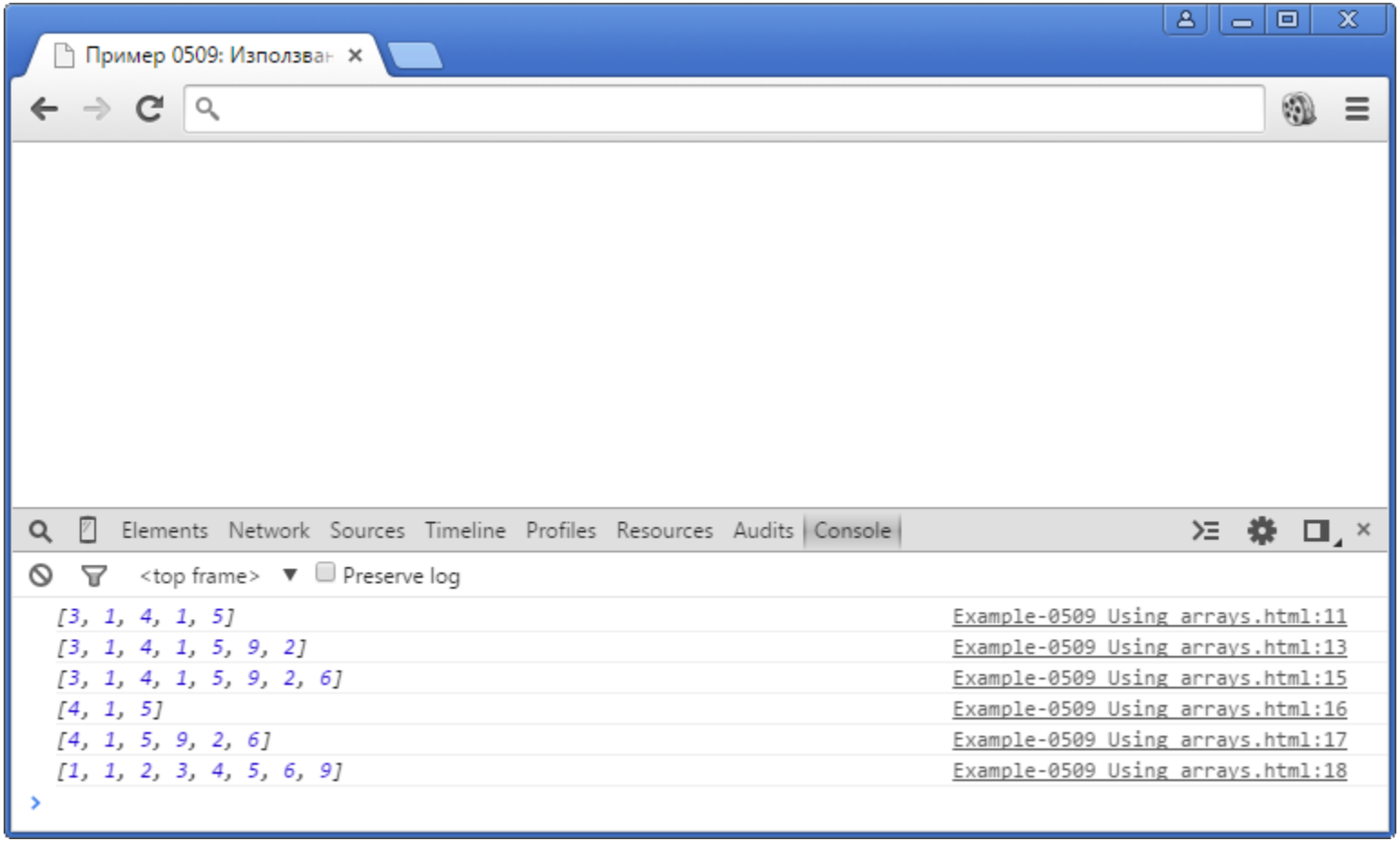
ПРОБА

Работа с масиви

- Добавяне към края на масив с push
- Сортиране със sort
- Извличане на част от масив със slice
  - от индекс до друг индекс, но без него
  - от индекс до края на масива

От индекс 2
до индекс 4
включително

```
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();
```


Пример 0509: Използван
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
[3, 1, 4, 1, 5]
Example-0509 Using arrays.html:11
[3, 1, 4, 1, 5, 9, 2]
Example-0509 Using arrays.html:13
[3, 1, 4, 1, 5, 9, 2, 6]
Example-0509 Using arrays.html:15
[4, 1, 5]
Example-0509 Using arrays.html:16
[4, 1, 5, 9, 2, 6]
Example-0509 Using arrays.html:17
[1, 1, 2, 3, 4, 5, 6, 9]
Example-0509 Using arrays.html:18

ПРОБА

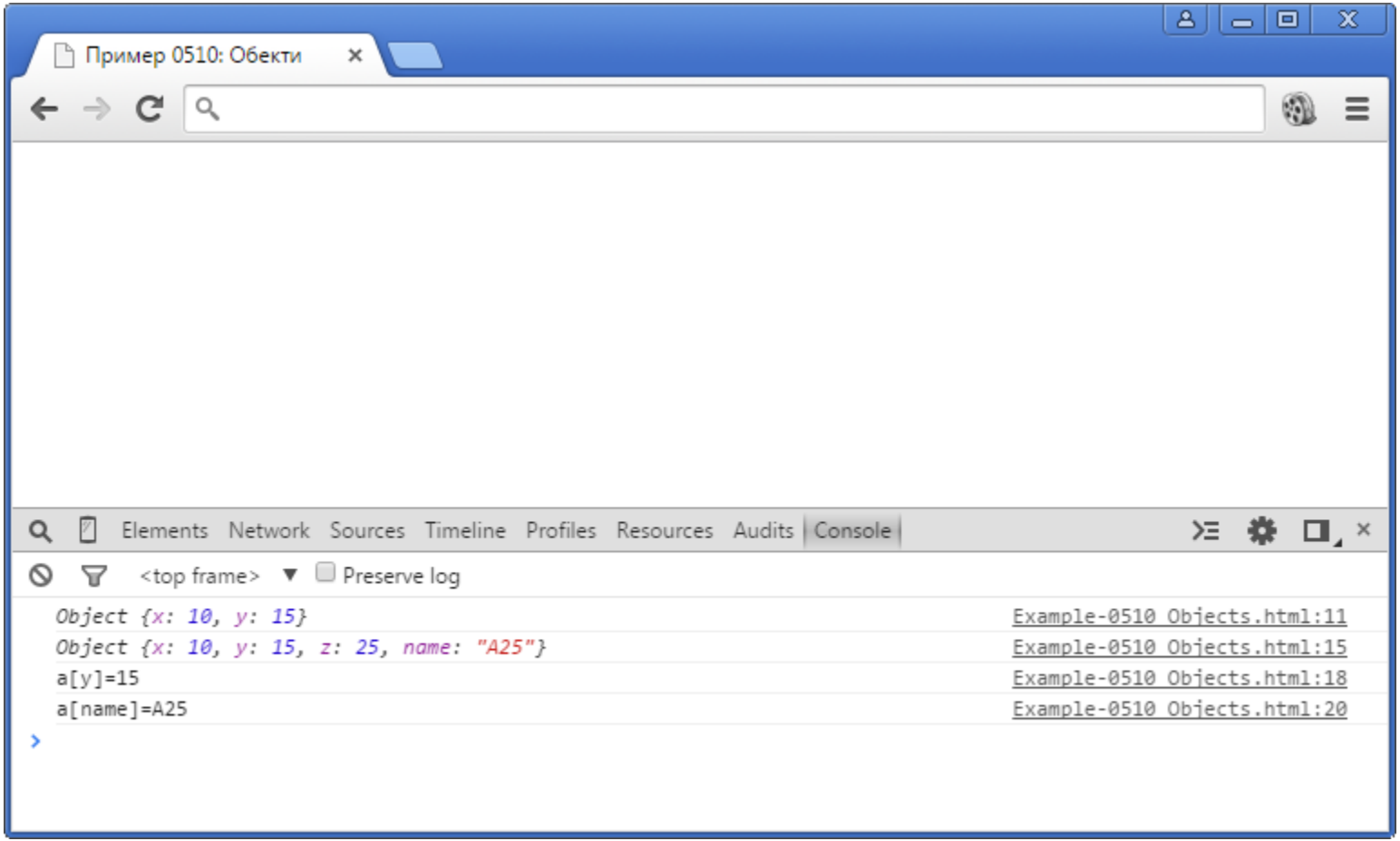
Обекти

- Константи {име:стойност, име:стойност, … }
- Достъп до елементите с обект.име или обект[име]

```
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]);
```


Пример 0510: Обекти
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
Object {x: 10, y: 15}
Example-0510 Objects.html:11
Object {x: 10, y: 15, z: 25, name: "A25"}
Example-0510 Objects.html:15
a[y]=15
Example-0510 Objects.html:18
a[name]=A25
Example-0510 Objects.html:20

ПРОБА

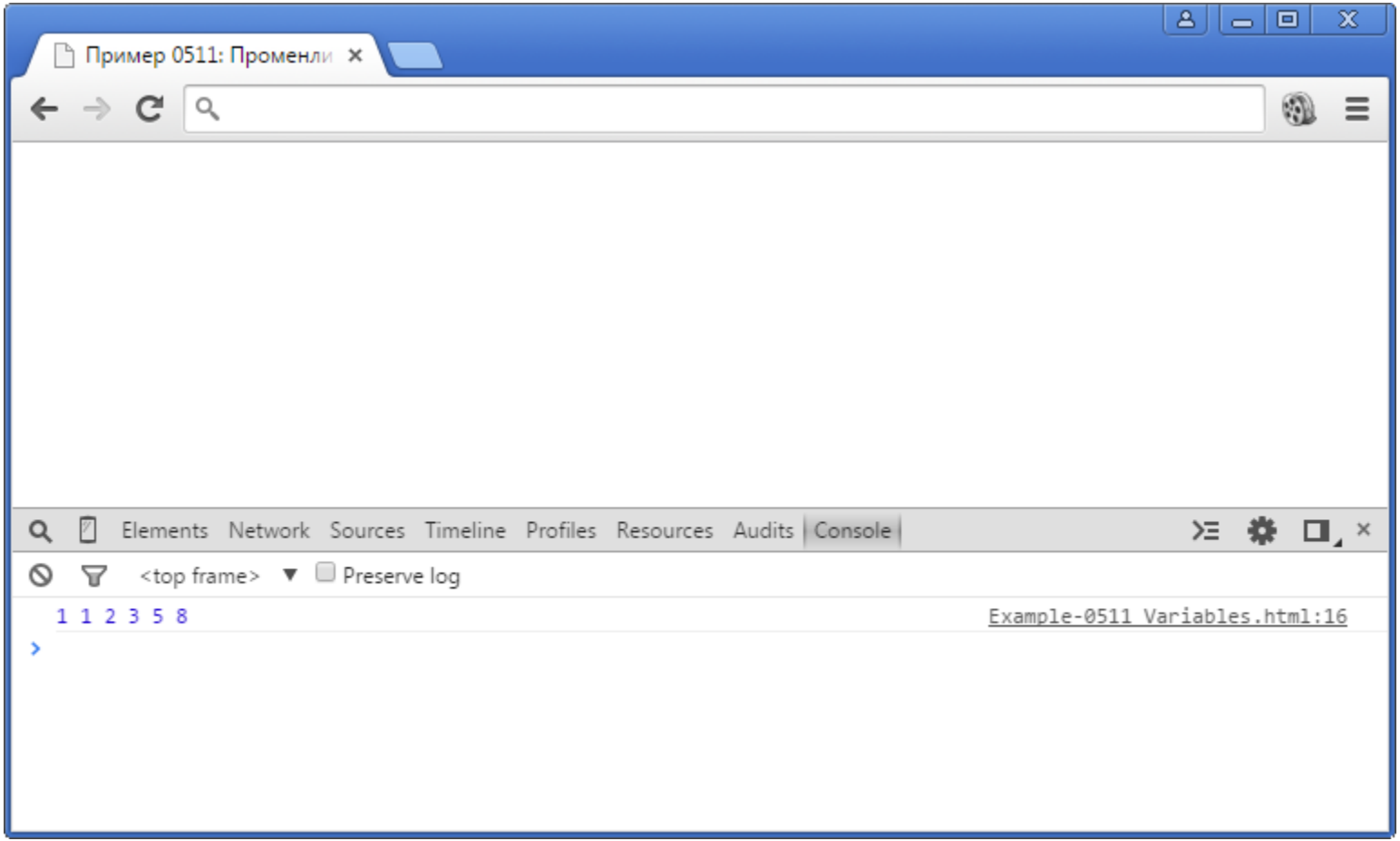
Синтаксис

Променливи

Променливи в JS

- Могат да си променят типа на стойността
- Има локални и глобални променливи
- Създаване на локална променлива с var име;
- Присвояване на стойност с име = стойност;
- Ако няма променлива с даденото име, създава се глобална

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


Пример 0511: Променли
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
1 1 2 3 5 8
Example-0511 Variables.html:16

ПРОБА

Условен оператор

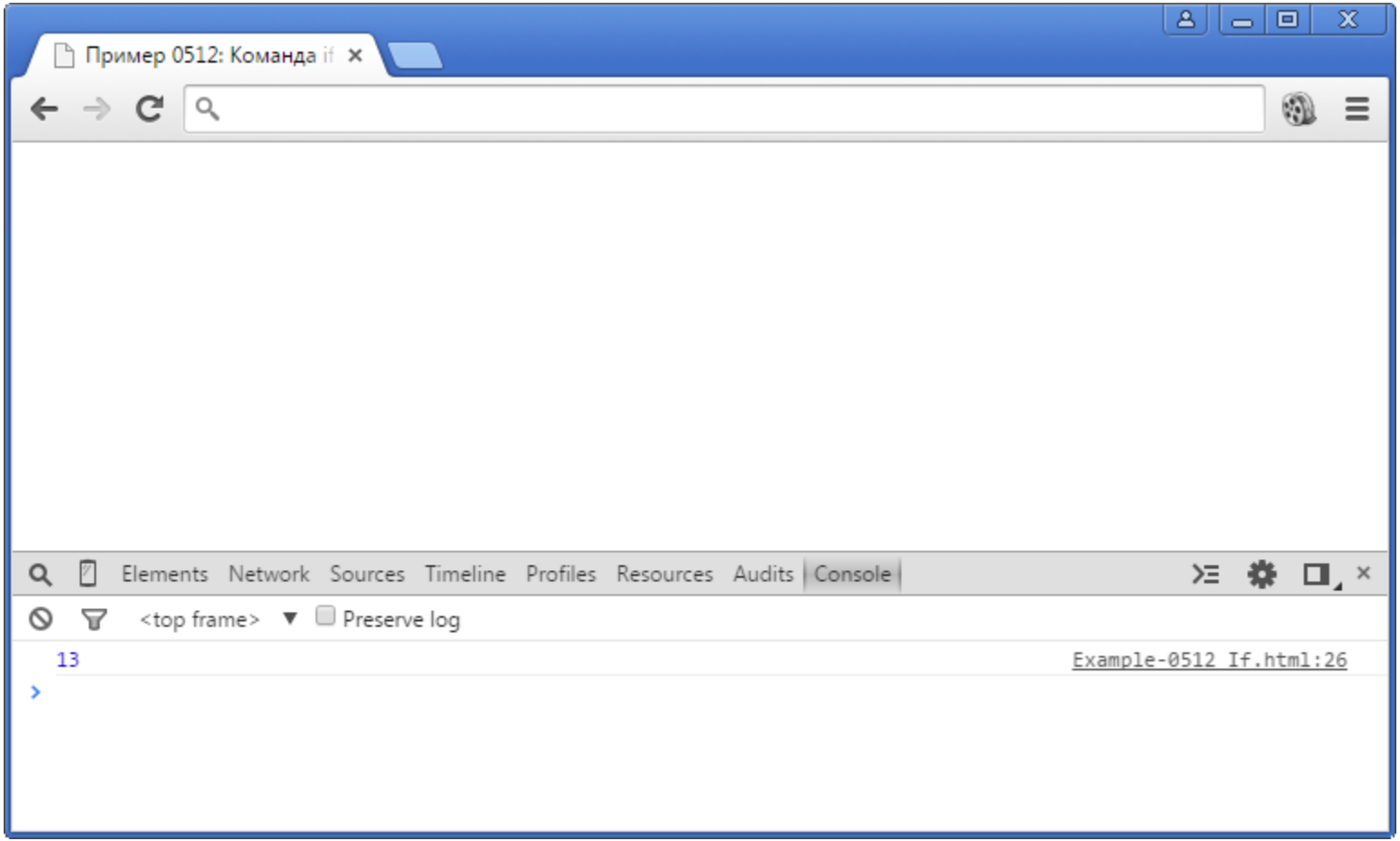
- Като команда

  if (условие) {команди}

  if (условие) {команди} else {команди}

  if (условие) {команди} else if (условие) {команди} …

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


Пример 0512: Команда if
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
13
Example-0512_If.html:26

ПРОБА

Внимание!

- Освен булеви стойности в условията могат да се ползват и стойности от друг тип
- Всичко, което не е „истински“ данни, се приема за false:
  - Самата стойност false
  - Числова нула 0
  - Празен низ ""
  - Недефинирана променлива undefined
  - Празна стойност null
  - Нечислово числов резултат NaN
- Подобно преобразуване на небулеви стойности се прави в командата if, функцията Boolean и оператора !

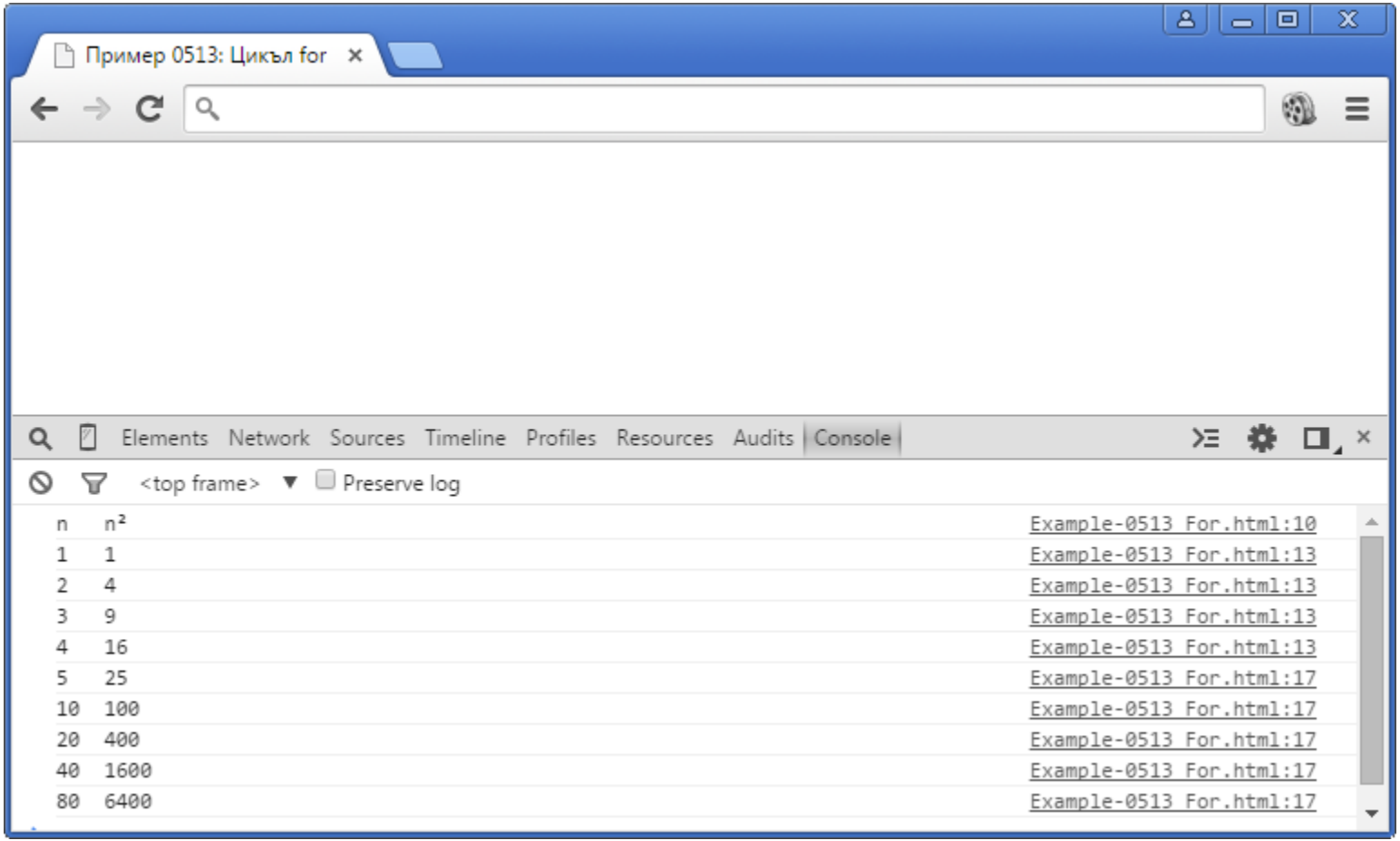
Цикъл

- Стандартен цикъл с начална стойност, условие и промяна на всяка стъпка for (начало; условие; стъпка) { команди }

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

- С условие на всяка стъпка while (условие) { команди }

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

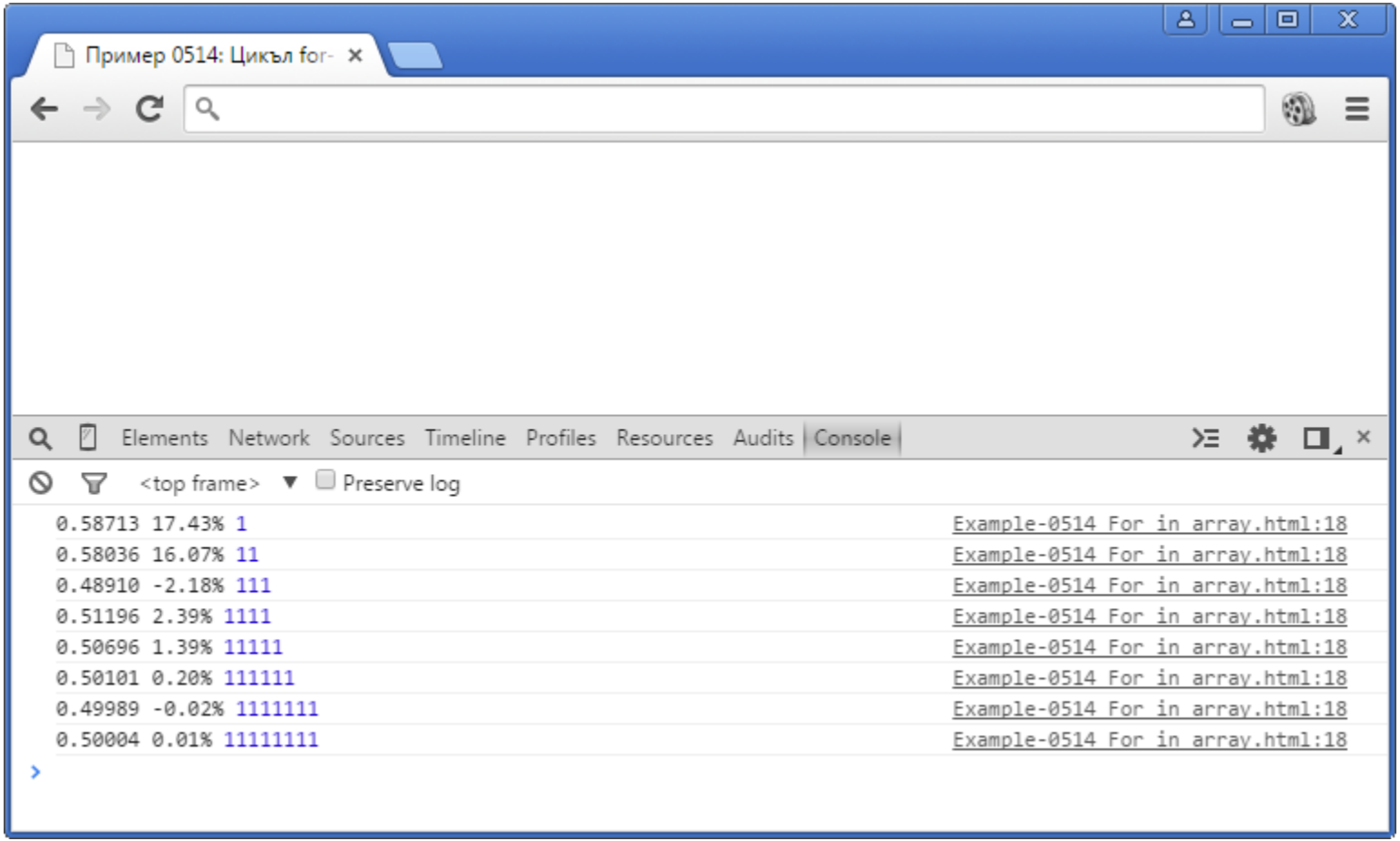

Пример 0513: Цикъл for
Elements Network Sources Timeline Profiles Resources Audits Console
<top frame>
Preserve log
n n²
1 1
2 4
3 9
4 16
5 25
10 100
20 400
40 1600
80 6400
Example-0513 For.html:10
Example-0513 For.html:13
Example-0513 For.html:13
Example-0513 For.html:13
Example-0513 For.html:13
Example-0513 For.html:17
Example-0513 For.html:17
Example-0513 For.html:17
Example-0513 For.html:17
Example-0513 For.html:17

ПРОБА

- Обхождане с for (променлива in масив) {…}
- Променливата приема за стойност индексите на елементите
- Пример за намиране на средна стойност на n числа в масив

```
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);
}
```

- При for (променлива in обект) {…} променливата приема за стойност имената на елементите в обекта


Пример 0514: Цикъл for-
Elements Network Sources Timeline Profiles Resources Audits Console
<top frame>
Preserve log
0.58713 17.43% 1 Example-0514 For in array.html:18
0.58036 16.07% 11 Example-0514 For in array.html:18
0.48910 -2.18% 111 Example-0514 For in array.html:18
0.51196 2.39% 1111 Example-0514 For in array.html:18
0.50696 1.39% 11111 Example-0514 For in array.html:18
0.50101 0.20% 111111 Example-0514 For in array.html:18
0.49989 -0.02% 1111111 Example-0514 For in array.html:18
0.50004 0.01% 11111111 Example-0514 For in array.html:18

ПРОБА

Функции

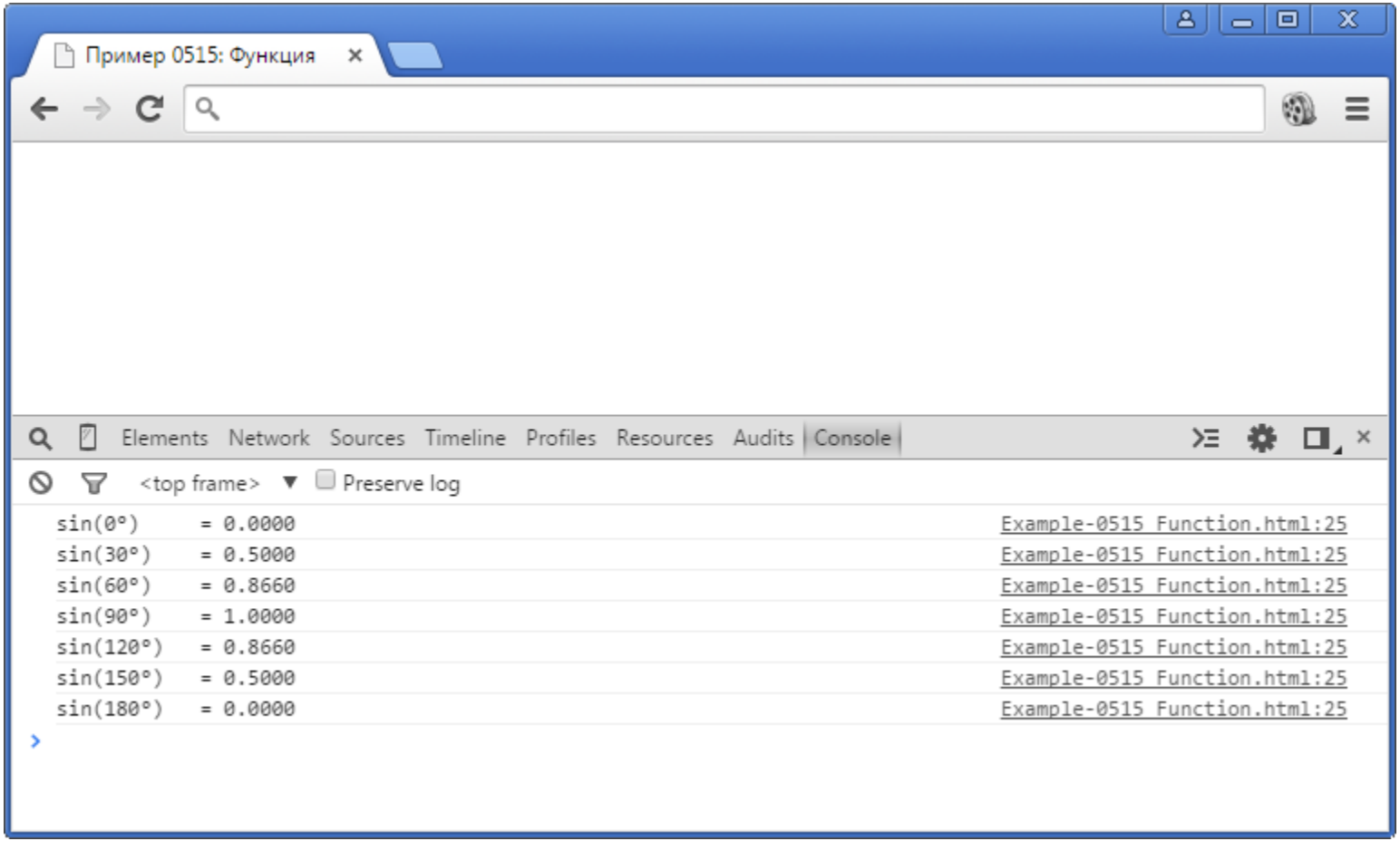
Функции в JS

Дефиниране на функция

- Запазена дума function
- Няма тип на функцията и на параметрите
- Резултат се връща с return

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

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

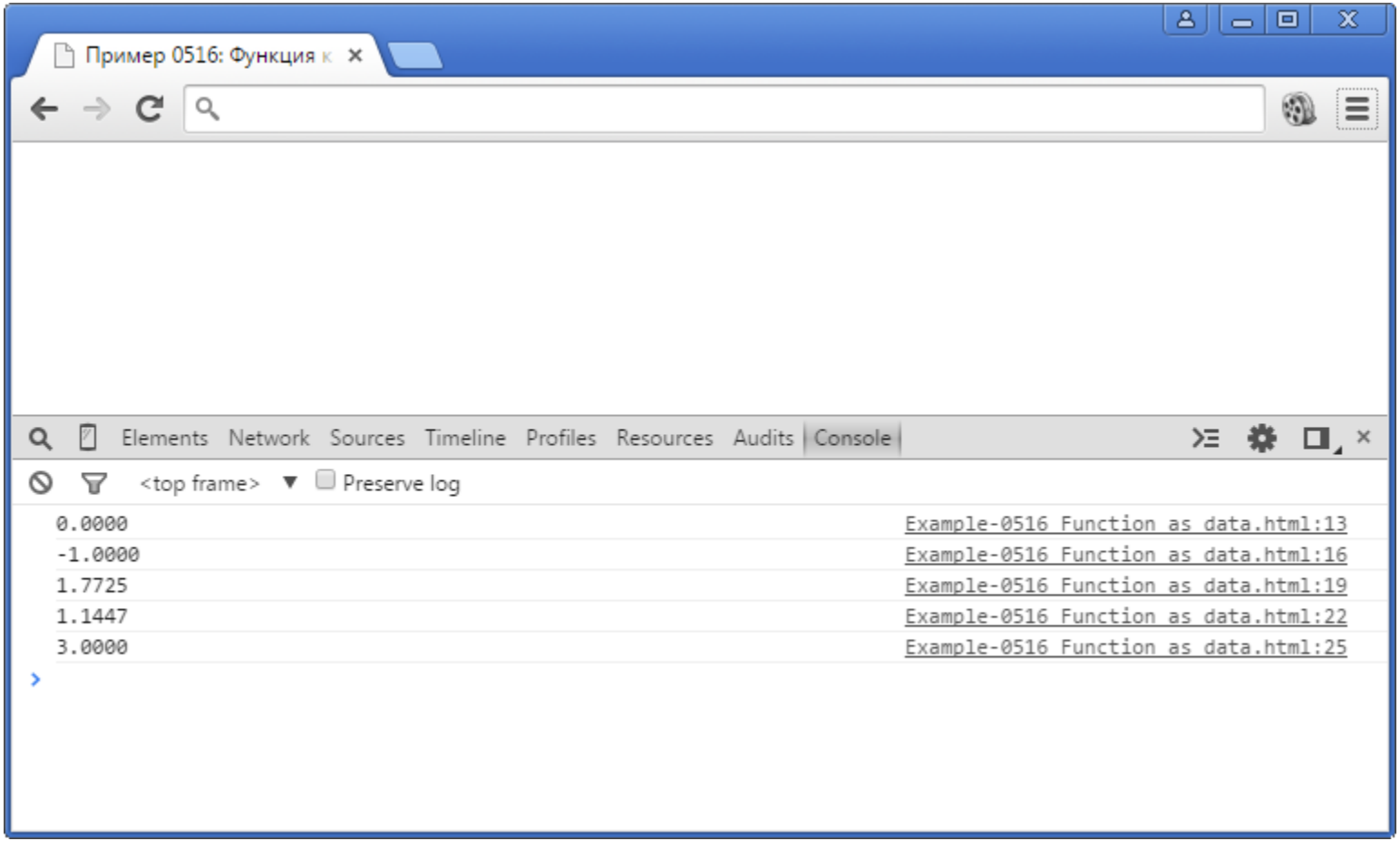

Пример 0515: Функция
Elements Network Sources Timeline Profiles Resources Audits Console
<top frame>
Preserve log
sin(0°) = 0.0000 Example-0515_Function.html:25
sin(30°) = 0.5000 Example-0515_Function.html:25
sin(60°) = 0.8660 Example-0515_Function.html:25
sin(90°) = 1.0000 Example-0515_Function.html:25
sin(120°) = 0.8660 Example-0515_Function.html:25
sin(150°) = 0.5000 Example-0515_Function.html:25
sin(180°) = 0.0000 Example-0515_Function.html:25

ПРОБА

Функции като данни

- В JS няма указатели
- Функции без (…) е данни от тип функция
- Могат да се присвояват на променливи и да се подават като параметри на функции

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


Пример 0516: Функция
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
0.0000
Example-0516 Function as data.html:13
-1.0000
Example-0516 Function as data.html:16
1.7725
Example-0516 Function as data.html:19
1.1447
Example-0516 Function as data.html:22
3.0000
Example-0516 Function as data.html:25

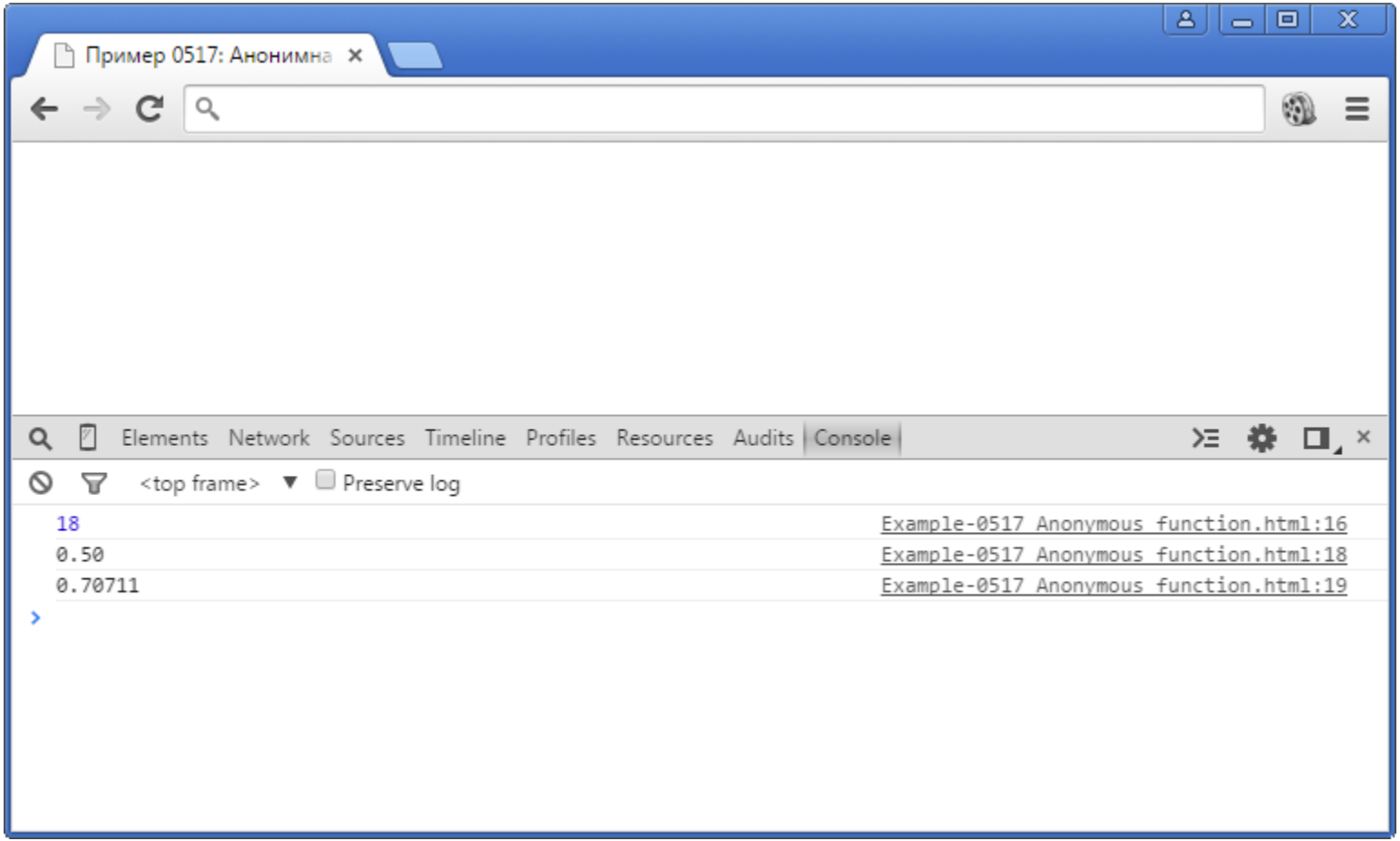
ПРОБА

Анонимни функции

- Функция без име, но с параметри и тяло
- Често се ползва еднократно за създаване на данна-функция

```
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 )
);
```


Пример 0517: Анонимна
Elements
Network
Sources
Timeline
Profiles
Resources
Audits
Console
<top frame>
Preserve log
18
Example-0517 Anonymous function.html:16
0.50
Example-0517 Anonymous function.html:18
0.70711
Example-0517 Anonymous function.html:19

ПРОБА

Обобщение

JavaScript

Езикът JavaScript

- Подобен на C
- Използва се в браузър

Типове данни

- Прости: числови, текстови, булеви
- Съставни: масиви, обекти, функции

Променливи и функции

- Нямат тип, тип имат само данните
- Може да се създава анонимна функция

Още

Допълнителна информация

- Ето там: http://www.w3schools.com/js/
- И там: http://www.w3schools.com/jsref

ИКТ в НОС

Край

Коментари, въпроси