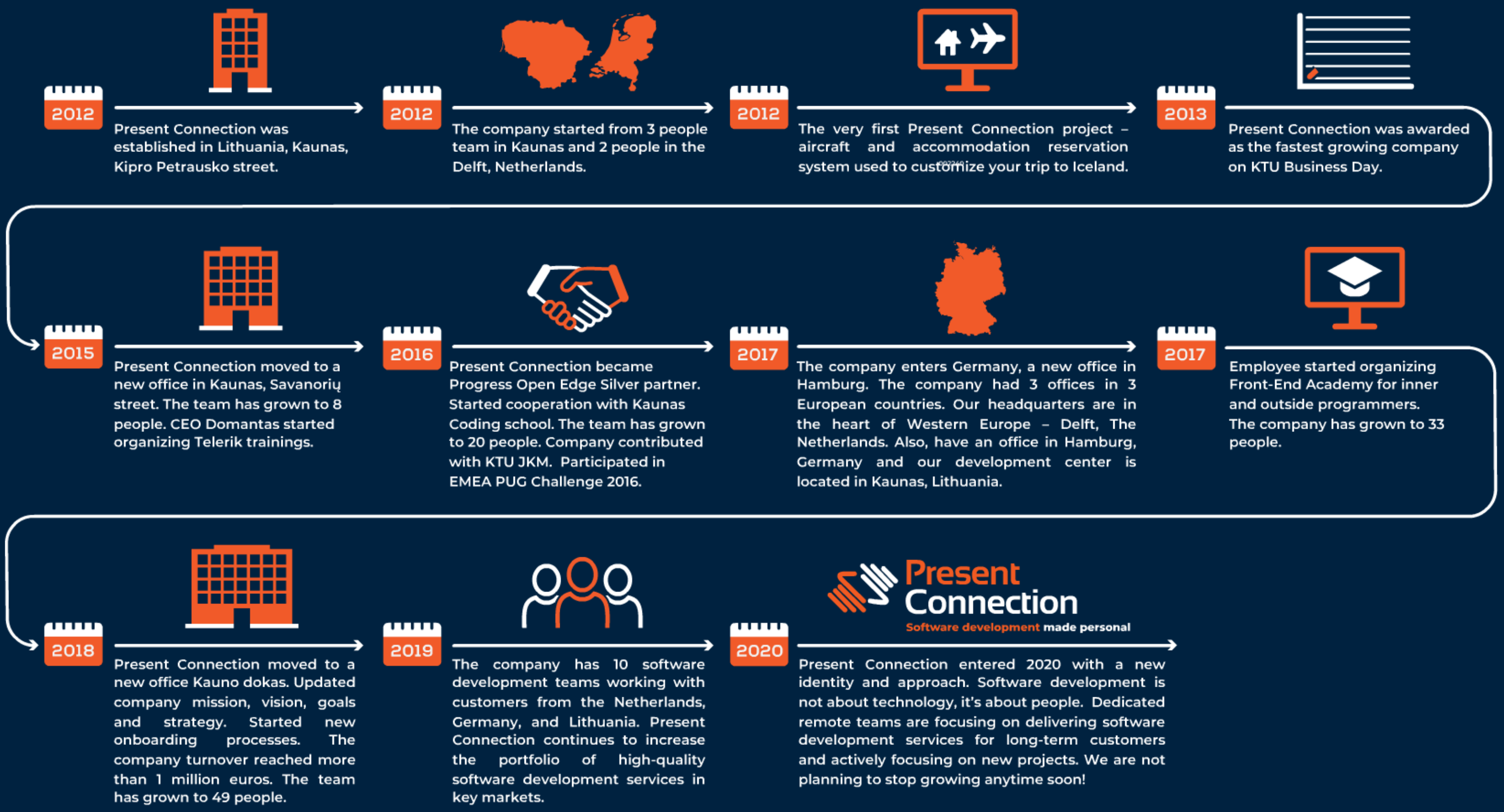
Present Connection

FRONT-END Academy

by Present Connection

Present Connection



2012

Present Connection was established in Lithuania, Kaunas, Kipro Petrausko street.

2012

The company started from 3 people team in Kaunas and 2 people in the Delft, Netherlands.

2012

The very first Present Connection project – aircraft and accommodation reservation system used to customize your trip to Iceland.

2013

Present Connection was awarded as the fastest growing company on KTU Business Day.

2015

Present Connection moved to a new office in Kaunas, Savanorių street. The team has grown to 8 people. CEO Domantas started organizing Telerik trainings.

2016

Present Connection became Progress Open Edge Silver partner. Started cooperation with Kaunas Coding school. The team has grown to 20 people. Company contributed with KTU JKM. Participated in EMEA PUG Challenge 2016.

2017

The company enters Germany, a new office in Hamburg. The company had 3 offices in 3 European countries. Our headquarters are in the heart of Western Europe – Delft, The Netherlands. Also, have an office in Hamburg, Germany and our development center is located in Kaunas, Lithuania.

2017

Employee started organizing Front-End Academy for inner and outside programmers. The company has grown to 33 people.

2018

Present Connection moved to a new office Kauno dokas. Updated company mission, vision, goals and strategy. Started new onboarding processes. The company turnover reached more than 1 million euros. The team has grown to 49 people.

2019

The company has 10 software development teams working with customers from the Netherlands, Germany, and Lithuania. Present Connection continues to increase the portfolio of high-quality software development services in key markets.

2020

Present Connection

Software development made personal

Present Connection entered 2020 with a new identity and approach. Software development is not about technology, it's about people. Dedicated remote teams are focusing on delivering software development services for long-term customers and actively focusing on new projects. We are not planning to stop growing anytime soon!

Present Connection

Ernestas Burokas

- 30 metų
- ~8 metai programavimo patirtis
- ernestas.burokas@presentconnection.eu
- Full-stack programuotojas

HTML JS CSS

SQL

.NET MVC

Knockout.

Entity Framework

APACHE CORDOVA™

mongoDB

Akademija

📅 2017

👥 ~30

👤 12

🎓 7

📅 2019

👥 34

👤 10 + 4

🎓 9

📅 2018

👥 109

👤 12

🎓 9

📅 2021

👥 67

👤 12

🎓 ?

Tikslas

Planas

JavaScript
Lapkričio 17

TypeScript
Lapkričio 24

Back-end + projektas
Gruodžio 1

React
Gruodžio 8

React
Gruodžio 15

React
Gruodžio 22

React
Gruodžio 29

React
Sausio 5

React
Sausio 12

React native
Sausio 19

React native
Sausio 26

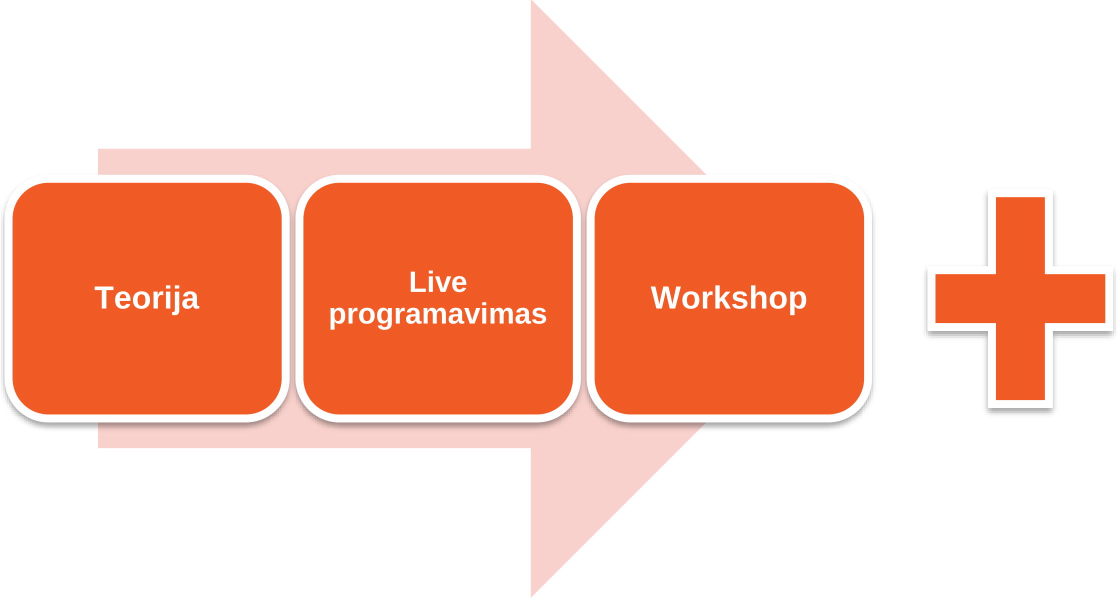
Finišas
Vasario 2

Lapkritis | Gruodis | Sausis | Vasaris

2021

2022

Paskaitų formatas



Namų darbai

Įrankiai

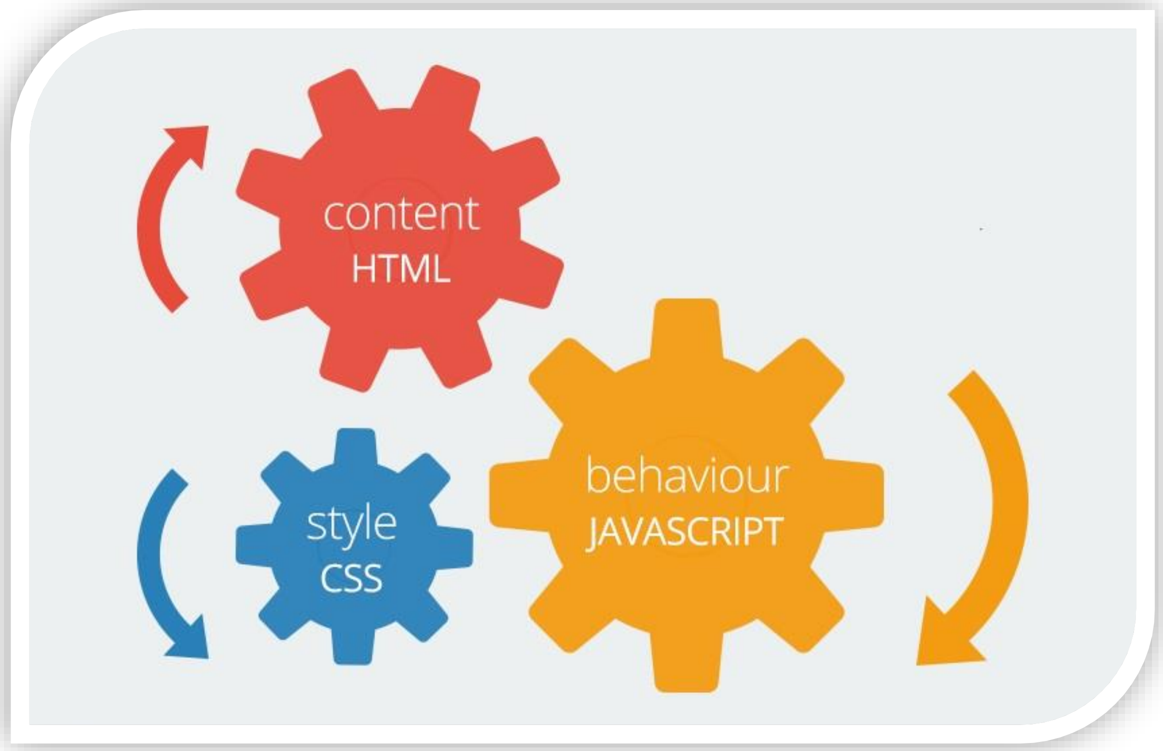
- Visual studio code
- Github
- Discord: https://discord.gg/GE9wsqyt
- Youtube: https://www.youtube.com/channel/UCfMJ83c_jaQLBq4vGfGvXiQ/videos

FRONT-END Academy

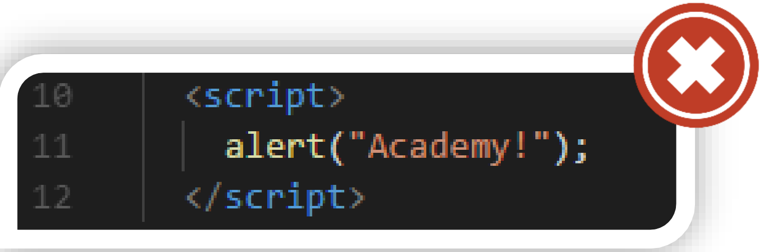
by Present Connection

Kas yra JavaScript?

- Programavimo kalba
- 1995m, Netscape
- Java != JavaScript

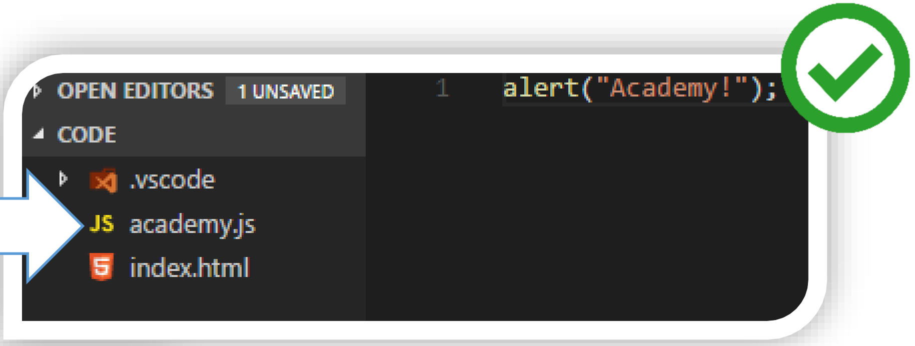


JavaScript naudojimas



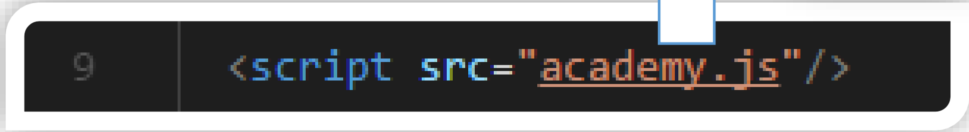
```
<script>
  alert("Academy!");
</script>
```

✗

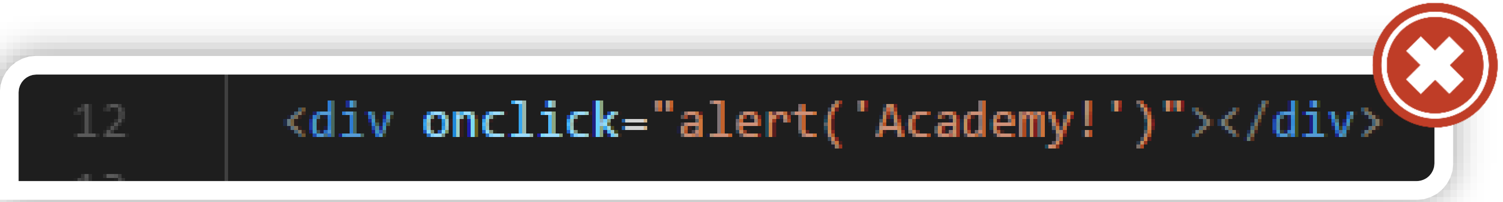


```
alert("Academy!");
```

✓



```
<script src="academy.js"/>
```



```
<div onclick="alert('Academy!')"></div>
```

✗

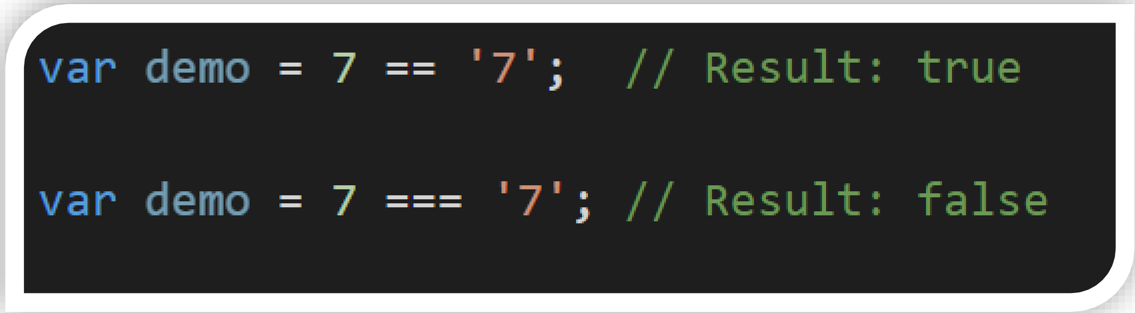
Kodėl nenaudoti inline JS?

- Negražus kodas
- Daugiau sveriantis html failas
- Toks JS kodas nebus cach’inamas
- Sunkesnis kodo palaikymas

JS įdomybės

===

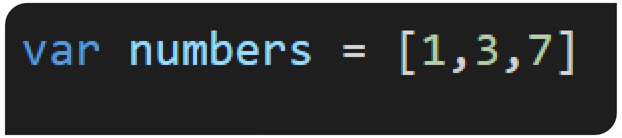
- Palygina ne tik reikšmę bet ir tipą
- Veikia greičiau negu dviguba lygybė
- Good practice ☺



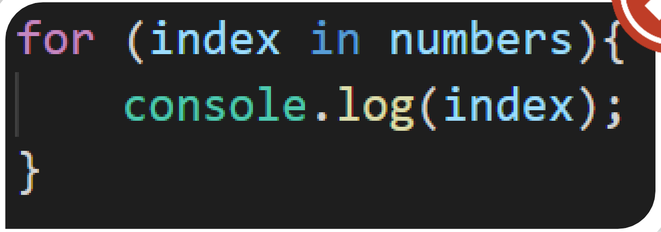
```
var demo = 7 == '7';  // Result: true

var demo = 7 === '7'; // Result: false
```

JS įdomybės

For in

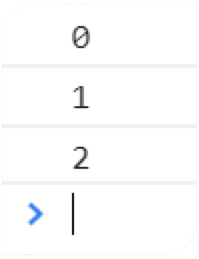


```
var numbers = [1,3,7]
```

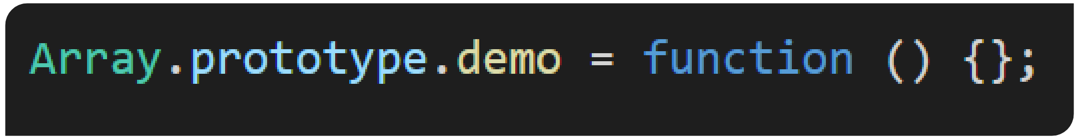


```
for (index in numbers){
    console.log(index);
}
```

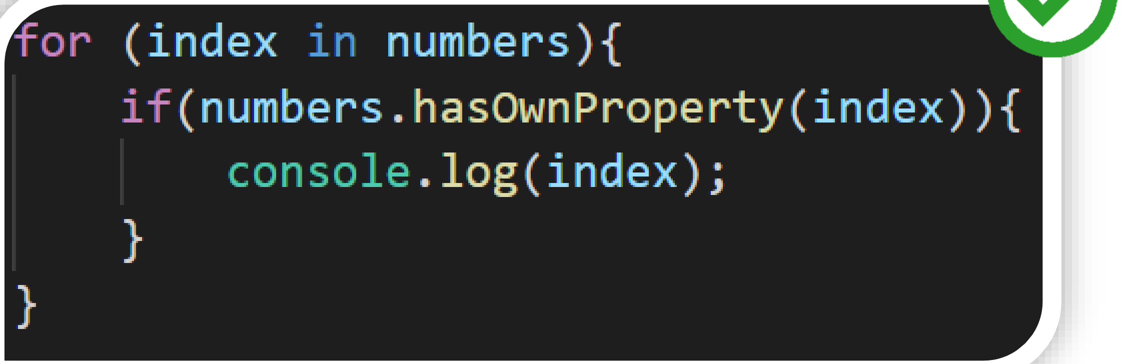
✖



```
0
1
2
> |
```



```
Array.prototype.demo = function () {};
```

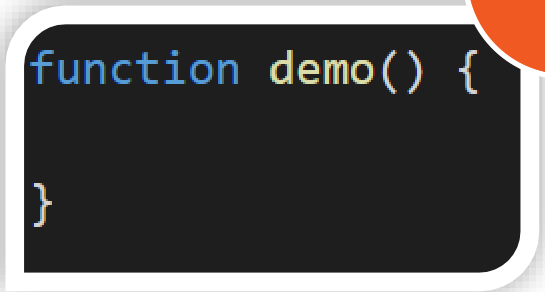


```
for (index in numbers){
    if(numbers.hasOwnProperty(index)){
        console.log(index);
    }
}
```

✓

JS įdomybės

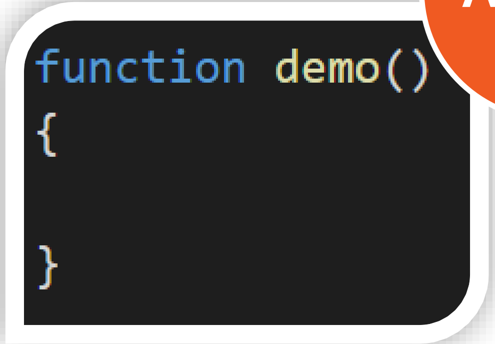
K&R



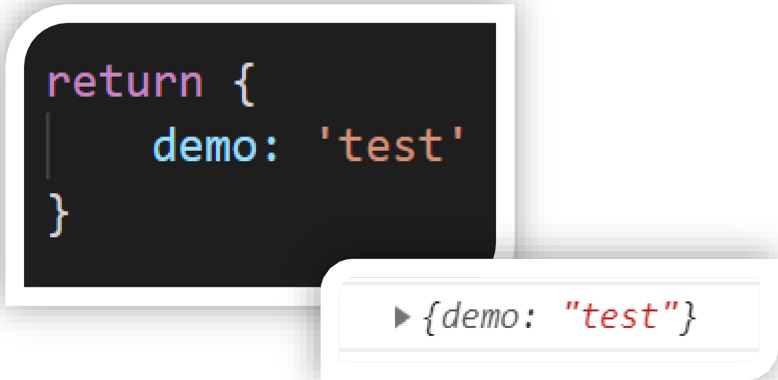
```
function demo() {

}
```

ar

Allman

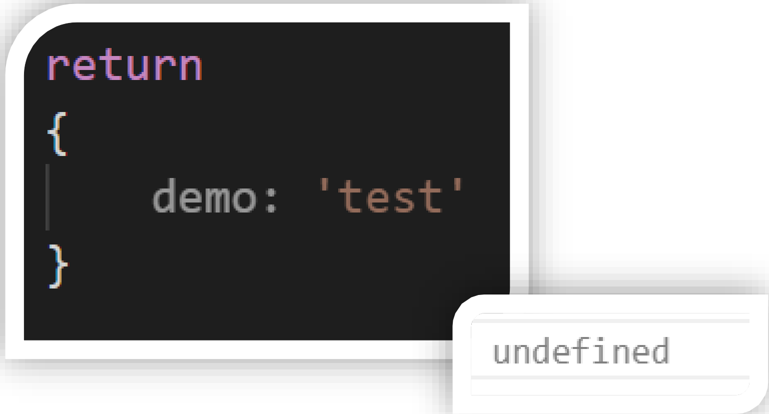


```
function demo()
{

}
```



```
return {
    demo: 'test'
}
```

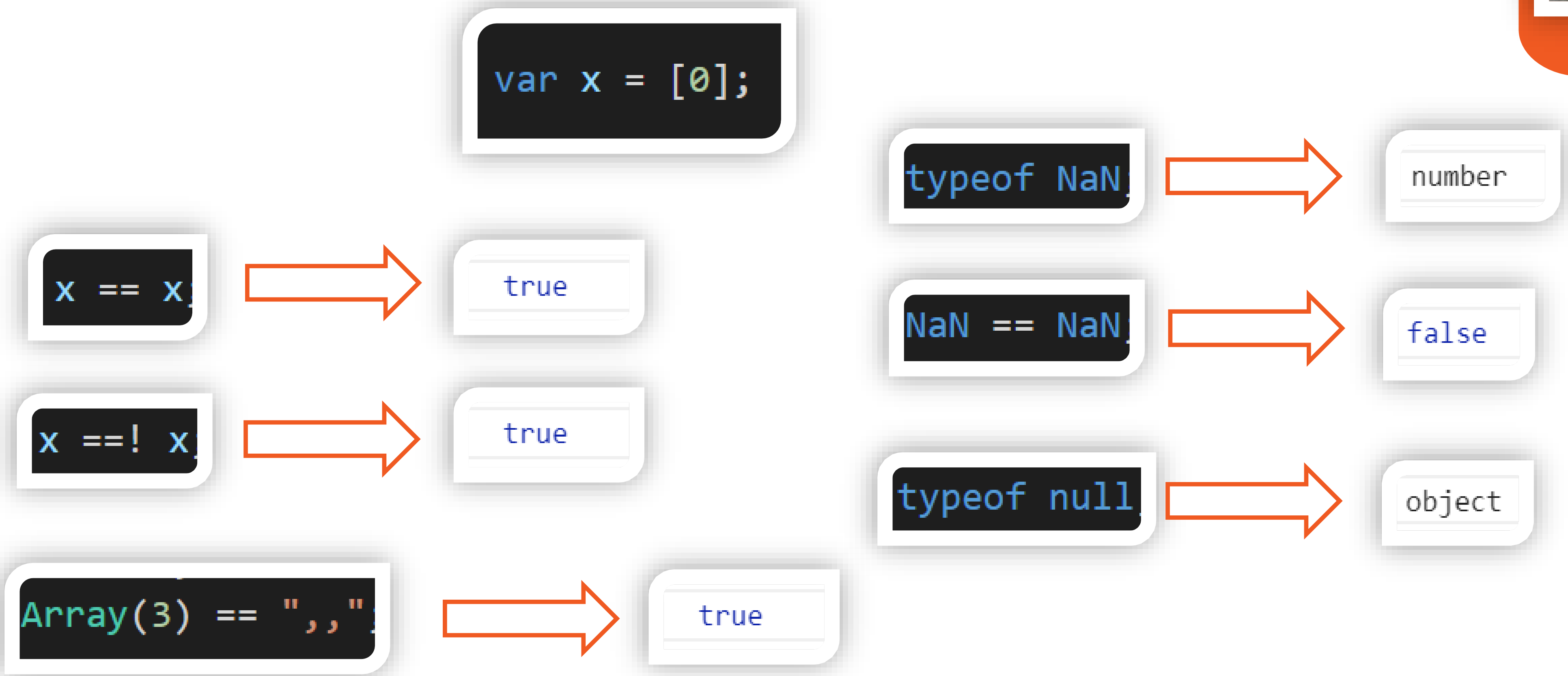
```
▸ {demo: "test"}
```



```
return
{
    demo: 'test'
}
```

```
undefined
```

JS įdomybės



```
var x = [0];
```

```
x == x
```

```
true
```

```
x ==! x
```

```
true
```

```
Array(3) == ",,"
```

```
true
```

```
typeof NaN
```

```
number
```

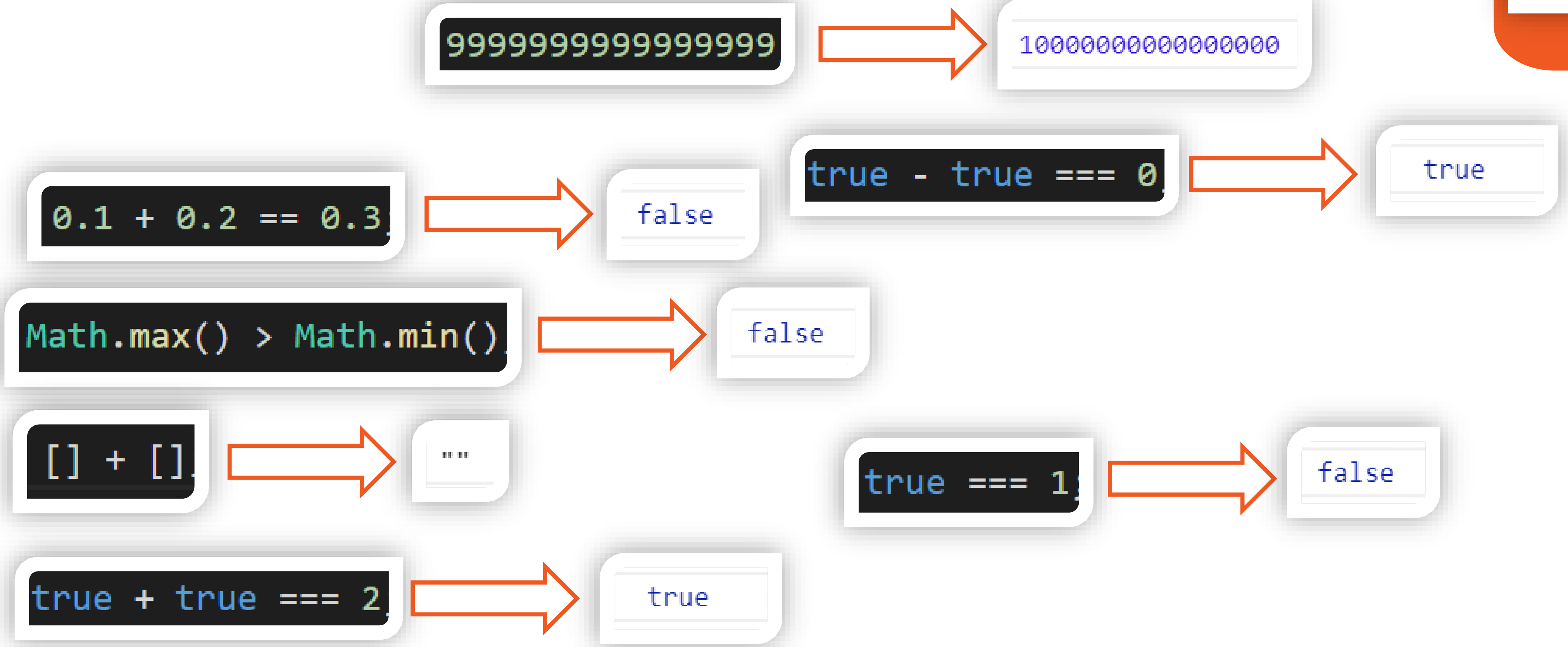
```
NaN == NaN
```

```
false
```

```
typeof null
```

```
object
```

JS įdomybės



```
9999999999999999
```

```
10000000000000000
```

```
0.1 + 0.2 == 0.3
```

```
false
```

```
true - true === 0
```

```
true
```

```
Math.max() > Math.min()
```

```
false
```

```
[] + []
```

```
""
```

```
true === 1
```

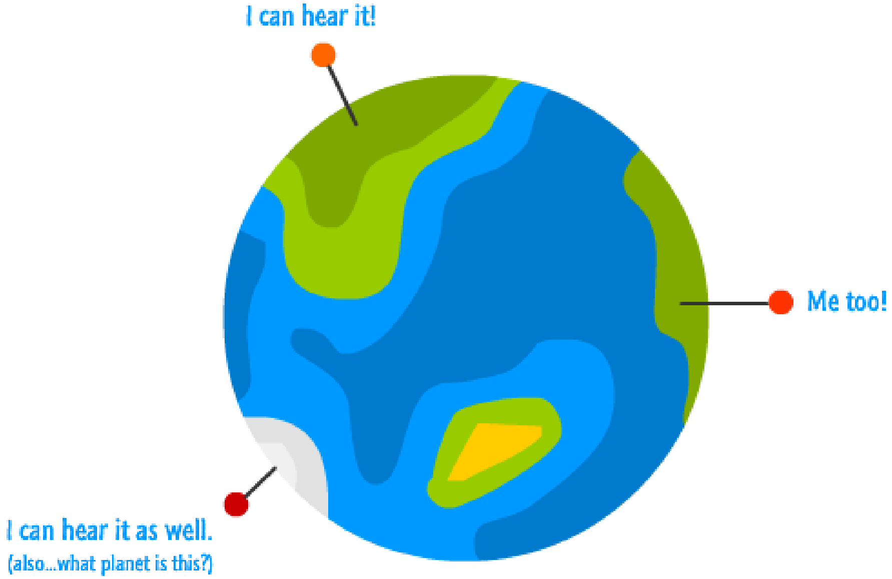
```
false
```

```
true + true === 2
```

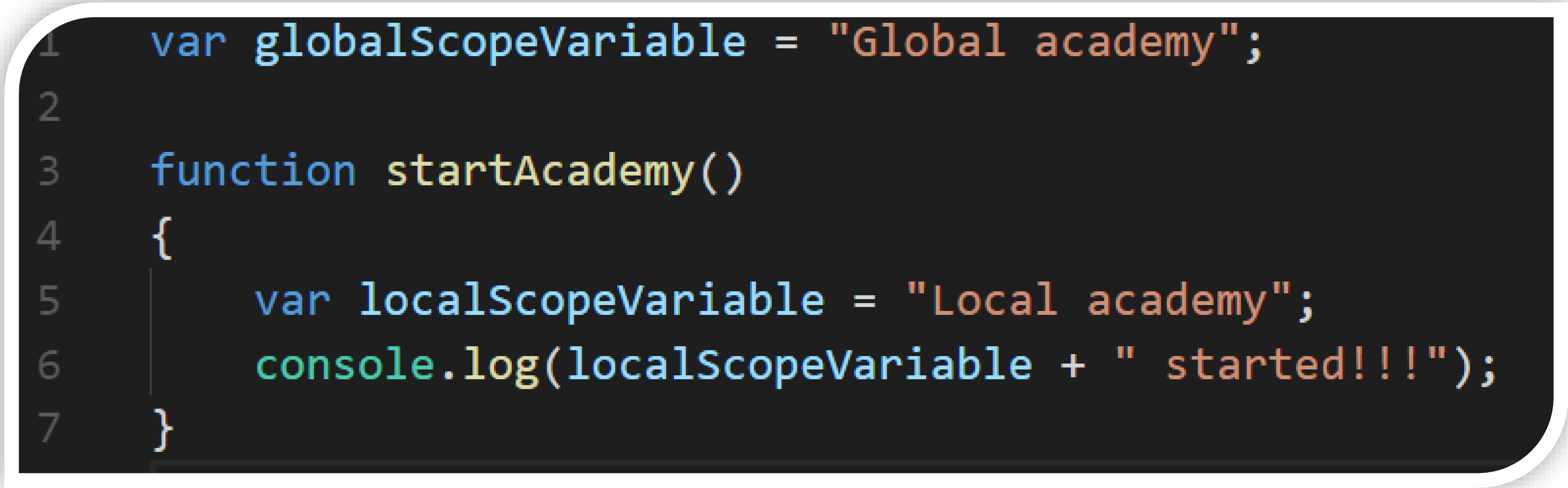
```
true
```

Scope

- Nusako kaip galime pasiekti kintamuosius, funkcijas, objektus
- Tipai:
  - Globalus
  - Lokalus
- Jeigu neinicijuojate kintamojo, jis tampa globalus.
- Nenaudokite globalių kintamųjų



Lokalus ir globalus scope



```
var globalScopeVariable = "Global academy";

function startAcademy()
{
    var localScopeVariable = "Local academy";
    console.log(localScopeVariable + " started!!!");
}
```

DEMO

Global Scope

https://toddmotto.com/everything-you-wanted-to-know-about-javascript-scope

Debuginimas

- console.log()
- debugger
- break point

DEMO

Debugging:
Console

https://raygun.com/javascript-debugging-tips

Linting

- Procesas skirtas analizuoti kodui bei nustatyti galimas problemas
- Sutaupo laiko
- Kodas tampa geresnis
- Pagrindiniai įrankiai:
    - JSHint
    - ESLint
    - JSLint

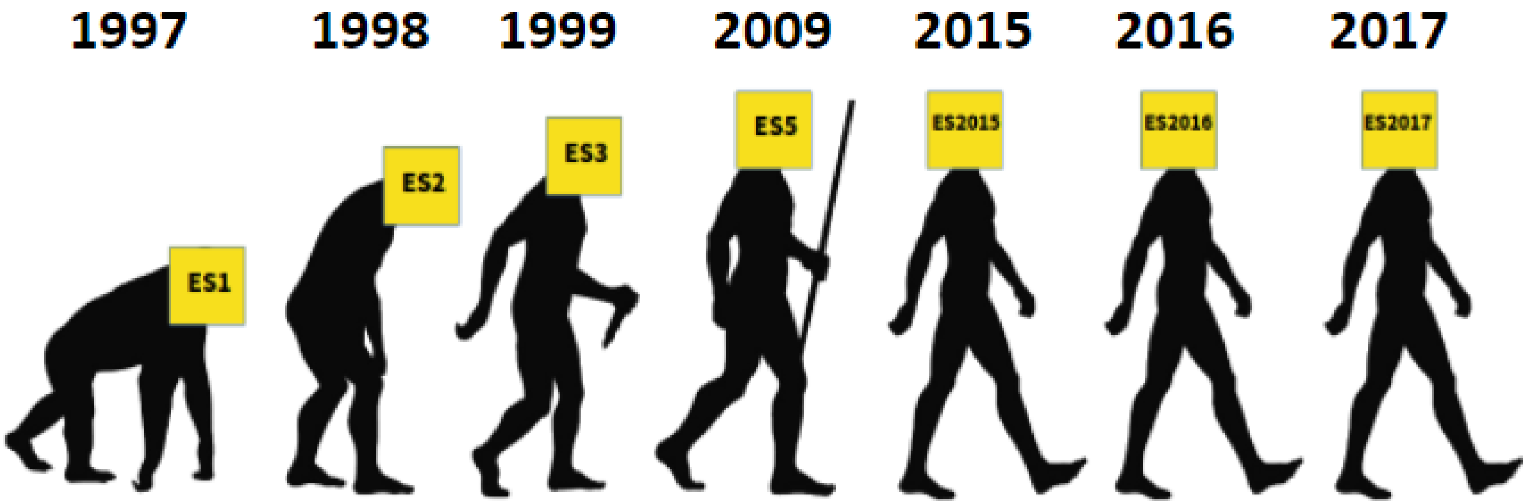
DEMO

ESLint:
Install &
Basic usage

https://eslint.org/docs/rules/

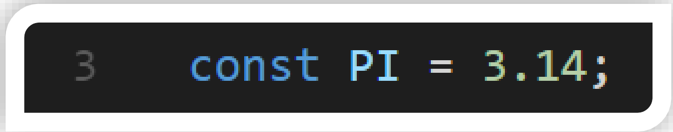
Ecma

- Standartas kurį naudoja JavaScript
- Naujesni negu ecma5 nėra visiškai palaikomi

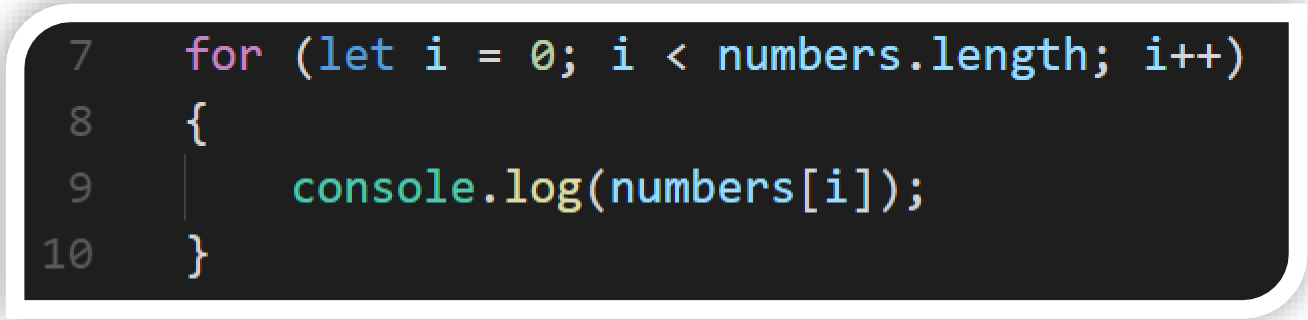


const ir let

- Block-scoped
- Negalioja hoisting
- Negalima iš naujo paskelbti



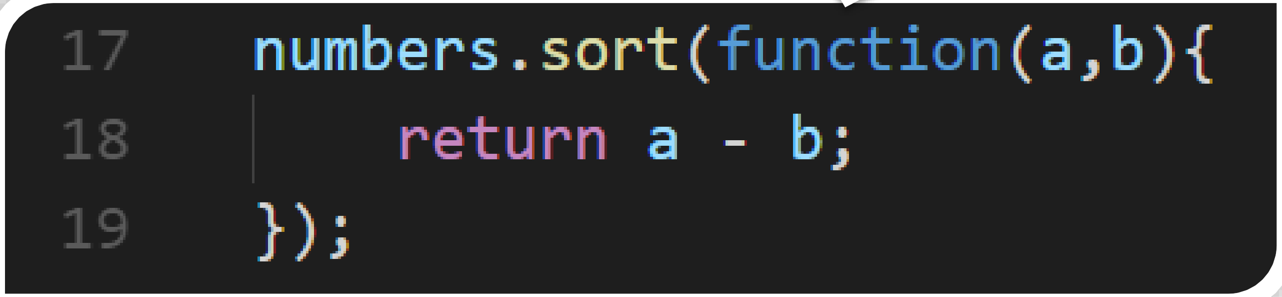
```
const PI = 3.14;
```



```
for (let i = 0; i < numbers.length; i++)
{
    console.log(numbers[i]);
}
```

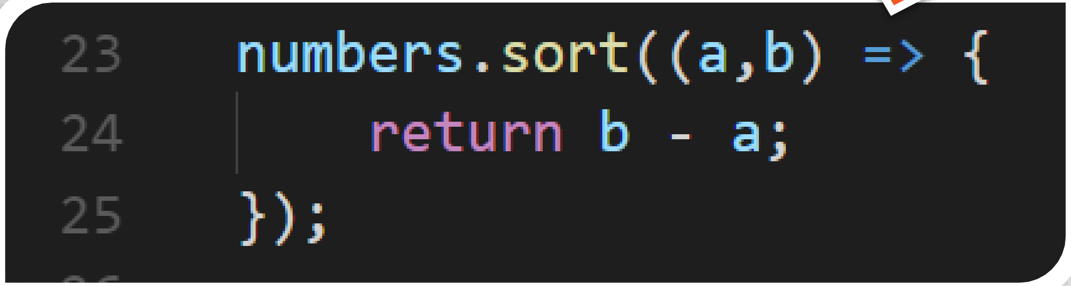
Arrow funkcijos

ecma5



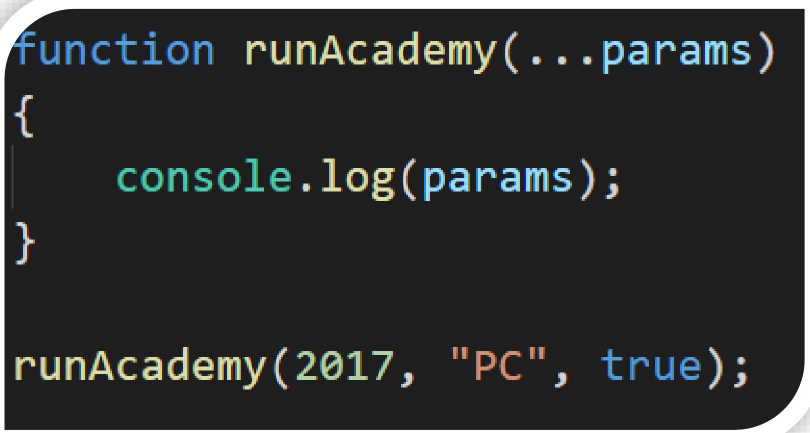
```
numbers.sort(function(a,b){
    return a - b;
});
```

ecma6

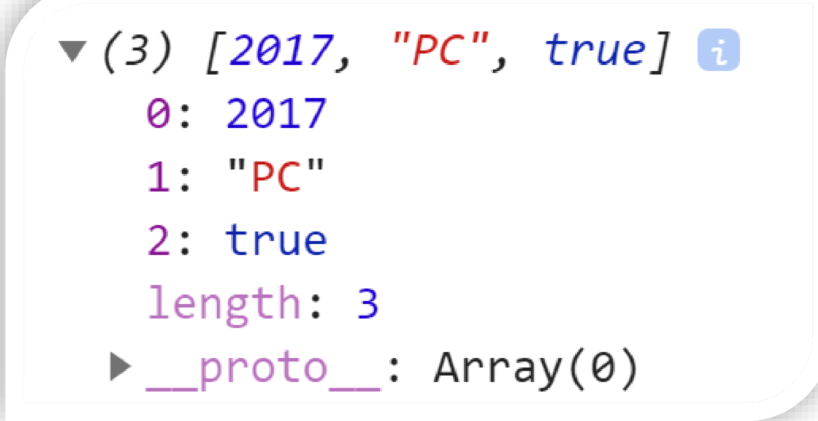


```
numbers.sort((a,b) => {
    return b - a;
});
```

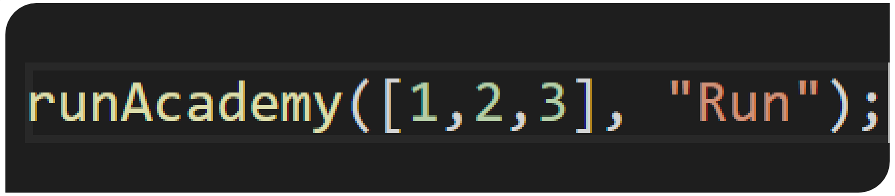
Rest parameter



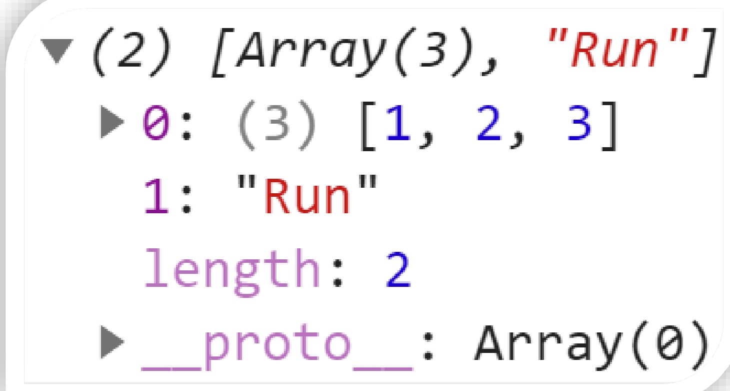
```
function runAcademy(...params)
{
    console.log(params);
}

runAcademy(2017, "PC", true);
```



```
▼(3) [2017, "PC", true] i
    0: 2017
    1: "PC"
    2: true
    length: 3
  ▶ __proto__: Array(0)
```

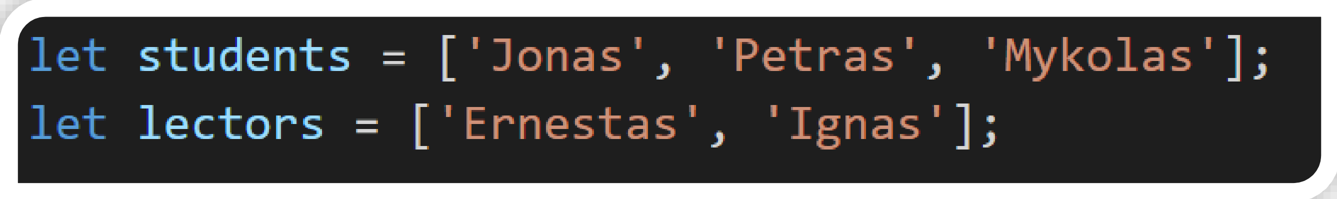


```
runAcademy([1,2,3], "Run");
```



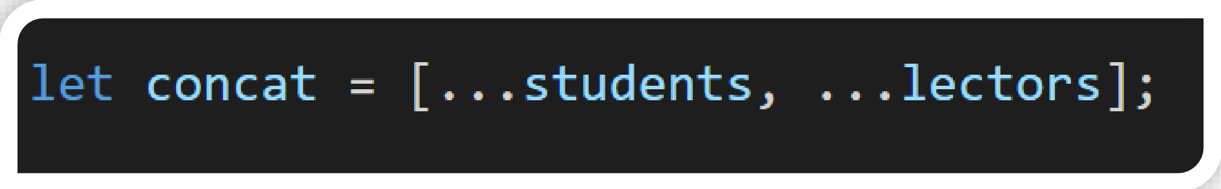
```
▼(2) [Array(3), "Run"]
  ▶0: (3) [1, 2, 3]
   1: "Run"
   length: 2
  ▶__proto__: Array(0)
```

Spread operatorius

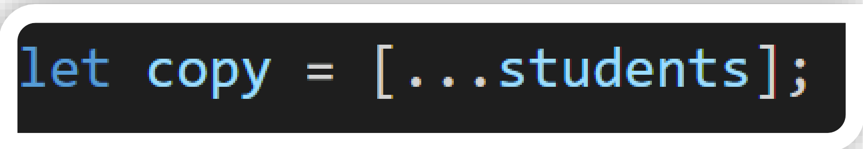


```
let students = ['Jonas', 'Petras', 'Mykolas'];
let lectors = ['Ernestas', 'Ignas'];
```

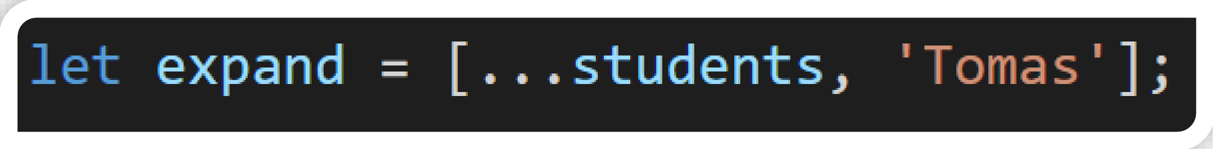
- Daugiausiai naudojamas masyvams, kad išskleisti elementus
- Sujungti masyvams
- Kopijuoti masyvus
- Plėsti masyvus



```
let concat = [...students, ...lectors];
```

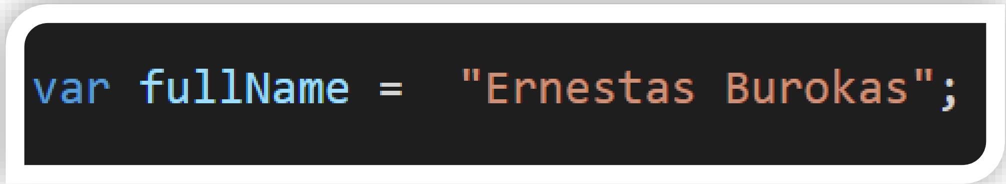


```
let copy = [...students];
```



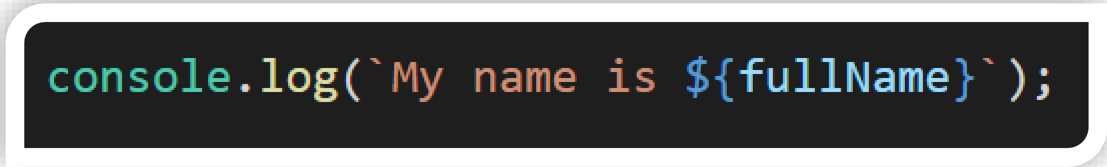
```
let expand = [...students, 'Tomas'];
```

Tekstų formatavimas



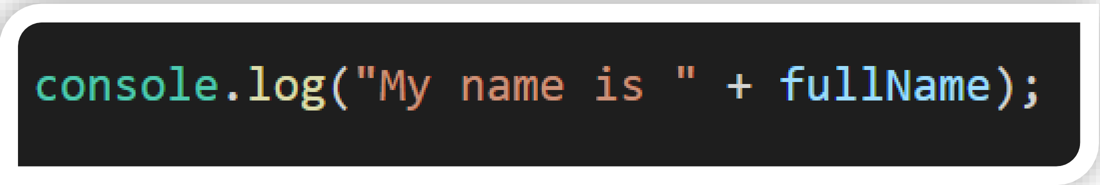
```
var fullName =  "Ernestas Burokas";
```

ecma6



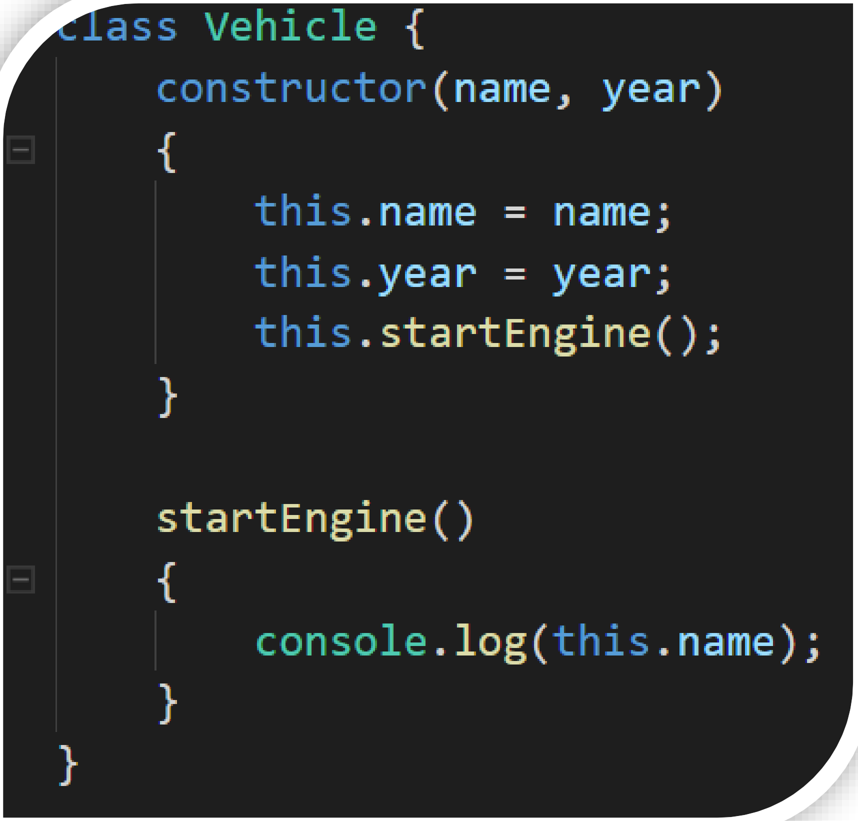
```
console.log(`My name is ${fullName}`);
```

ecma5

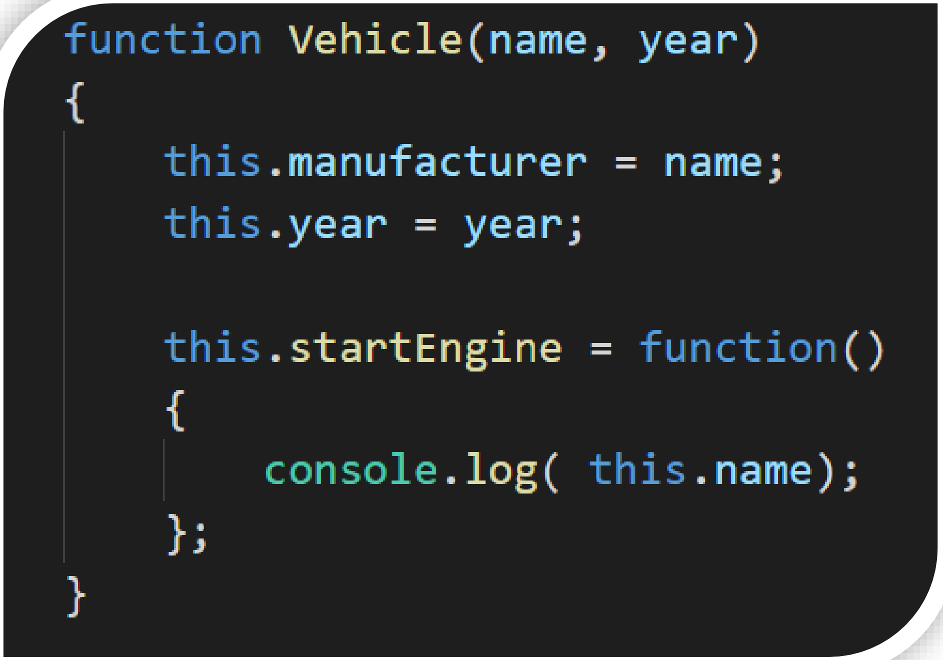


```
console.log("My name is " + fullName);
```

Klasės



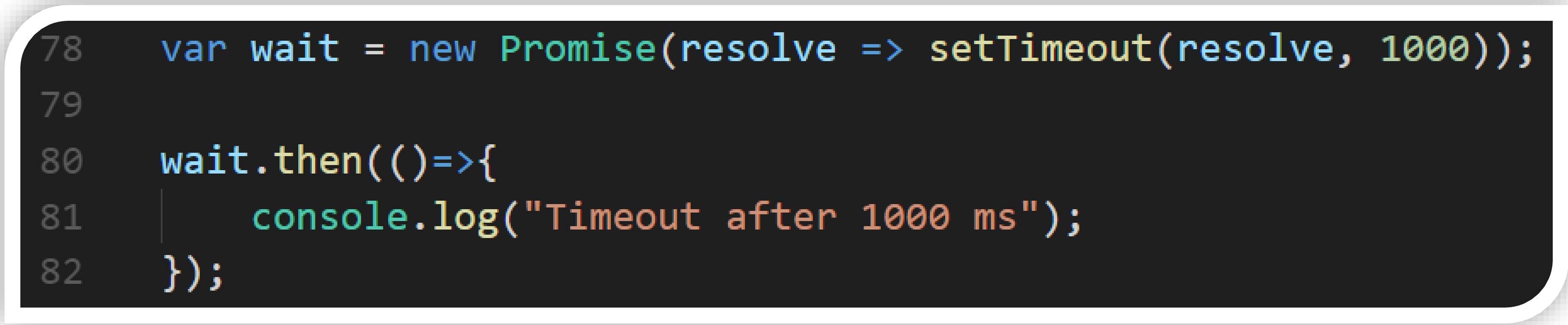
```
class Vehicle {
    constructor(name, year)
    {
        this.name = name;
        this.year = year;
        this.startEngine();
    }

    startEngine()
    {
        console.log(this.name);
    }
}
```



```
function Vehicle(name, year)
{
    this.manufacturer = name;
    this.year = year;

    this.startEngine = function()
    {
        console.log( this.name);
    };
}
```

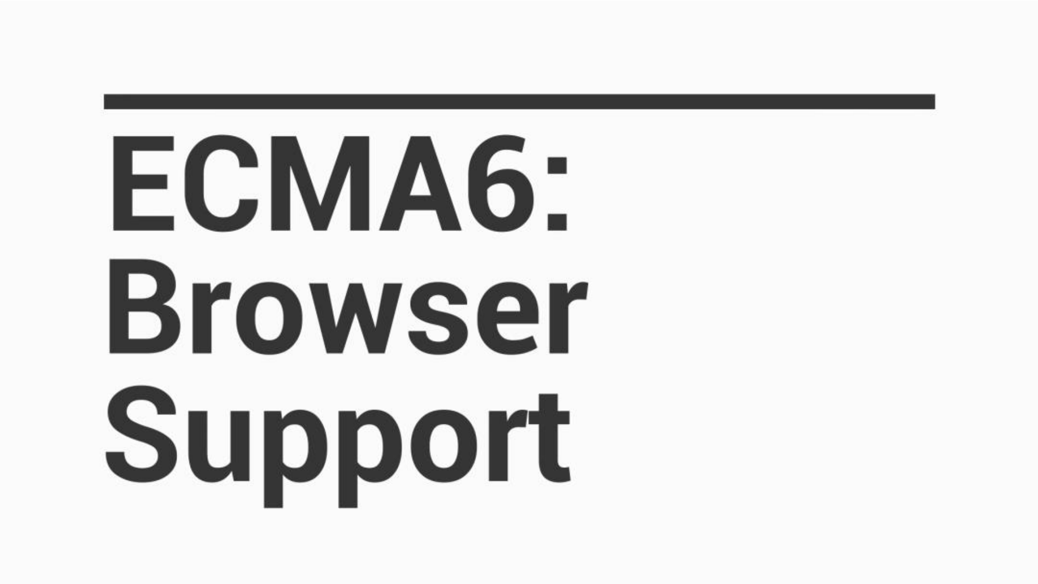
Promise

Apibūdina galimą asinchroninės operacijos užbaigimą (arba nesėkmę) bei grąžina rezultatą.



```
var wait = new Promise(resolve => setTimeout(resolve, 1000));

wait.then(()=>{
    console.log("Timeout after 1000 ms");
});
```

DEMO



ECMA6:
Browser
Support

http://es6-features.org/

Namų darbai

- Pakeisti textArea kad atitiktų šiuos reikalavimus:
  - Yra pateikti du parametrai: minRows, maxRows
  - Kai tuščia textArea turi būti min rows aukščio
  - Kai tekstas viršija min rows text area auga iki max rows aukščio
  - Kai tekstas viršija max rows rodyti vertikalų scrollbar
  - Neleisti vartotojui “tampyti” text area
  - Naudoti tik JS, HTML ir CSS
  - Panaudoti EsLint ir Babel (jeigu naudosit ECMA6)