JavaScript

special features

PHP web development 2019/2020

Milena Tomova Vratsa Software

https://vratsasoftware.com/

Table of Contents



- 1. ES6
 - a. arrow functions
 - b. default parameter values
 - c. Array.find()
 - d. Array.findIndex()
 - e. browsers' support
- 2. use strict



ES6 arrow functions

ES6

Arrow functions allows a short syntax for writing function expressions.

You don't need the

- function keyword,
- the return keyword,
- and the curly brackets.

ES6 arrow functions



ES5

```
var x = function(x, y) {
   return x * y;
}
x(5,2)//10
```

ES6

```
const x = (x, y) \Rightarrow x * y;
x(5,2)//10
```

ES6 arrow functions - 2



ES5

```
var x = function(x, y) {
  //some complicated multiline code concerning x and y
  return x * y;
  }
  x(5,2)//10
```

ES6

```
const x = (x, y) => {
//some complicated multiline code concerning x and y
return x * y;
}
x(5,2)//10
```



ES6 default parameter values

ES6

ES6 allows function parameters to have default values.

ES6 default parameter values



ES6

```
var x = function(x, y = 10, z = 10) {
   return x * y / z;
x(5)//30
X(5, 7)//
```

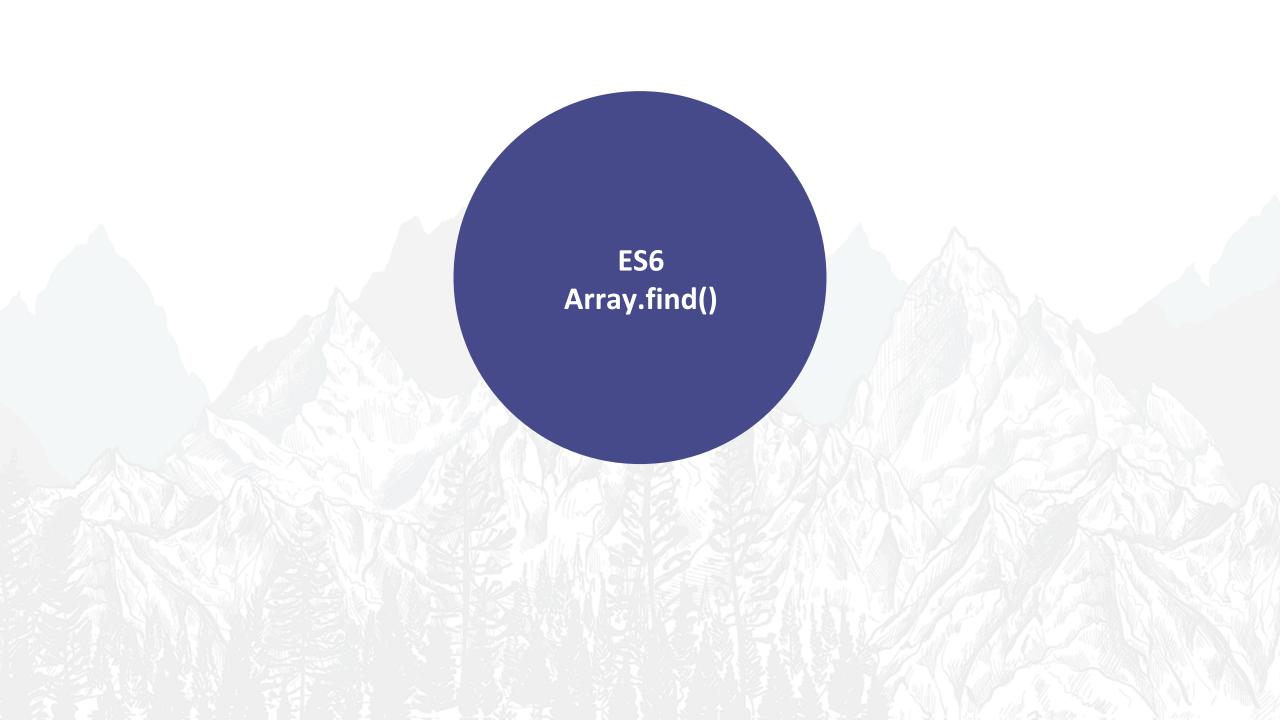


ES6 Array.findIndex()



The findIndex() method returns the index of the first array element that passes a test function.

```
var numbers = [4, 9, 16, 25, 29];
var first = numbers.findIndex(myFunction);//3
function myFunction(value, index, array) {
  return value > 18;
```



ES6 Array.find()



The find() method returns the value of the first array element that passes a test function.

```
var numbers = [4, 9, 16, 25, 29];
var first = numbers.find(myFunction);//25
function myFunction(value, index, array) {
  return value > 18;
```



ES6 browsers' support





Always check browsers` support!

ES6 browsers` support







The "use strict" directive was new in ECMAScript version 5.

The purpose of "use strict" is to indicate that the code should be executed in "strict mode".

With strict mode, you can not, for example, use undeclared variables.

All modern browsers support "use strict" except Internet Explorer 9 and lower:



Strict mode is declared by adding "use strict"; to the beginning of a script or a function.



Strict mode is declared by adding "use strict"; to the beginning of a script or a function.

```
"use strict";
myFunction();
function myFunction() {
   y = 3.14; // This will also cause an error because y is not declared
```



Declared inside a function, it has local scope (only the code inside the function is in strict mode):

```
x = 3.14; // This will not cause an error.
myFunction();
function myFunction() {
 "use strict";
          // This will cause an error
 y = 3.14;
```

Why "use strict"?





- Strict mode makes it easier to write "secure" JavaScript.
- Strict mode changes previously accepted "bad syntax" into real errors.
- As an example, in normal JavaScript, mistyping a variable name creates a new global variable. In strict mode, this will throw an error, making it impossible to accidentally create a global variable.
- In normal JavaScript, a developer will not receive any error feedback assigning values to non-writable properties.
- In strict mode, any assignment to a non-writable property, a getter-only property, a non-existing property, a non-existing variable, or a non-existing object, will throw an error.



Deleting a variable (or object) is not allowed.

```
"use strict";
var x = 3.14;
delete x; // This will cause an error
```



Deleting a function is not allowed.

```
"use strict";
function x(p1, p2) {};
delete x; // This will cause an error
```



Duplicating a parameter name is not allowed:

```
"use strict";
function x(p1, p1) {}; // This will cause an error
```

more info

Questions?



Partners















Trainings @ Vratsa Software



- Vratsa Software High-Quality Education, Profession and Jobs
 - www.vratsasoftware.com
- The Nest Coworking
 - www.nest.bg
- Vratsa Software @ Facebook
 - www.fb.com/VratsaSoftware
- Slack Channel
 - www.vso.slack.com



