Day 5: Arrow Functions

by AvmnuSng

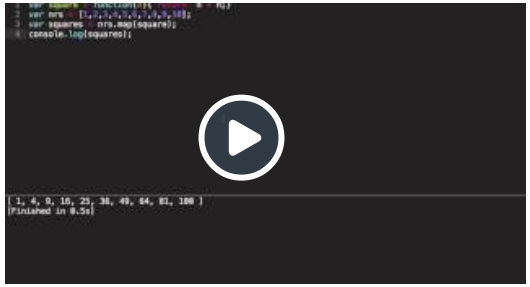
Problem Submissions Leaderboard Discussions Editorial Tutorial

Arrow Functions in JavaScript

Arrow Functions

Table Of Contents

Here are some basic examples of arrow function syntax:



```
(parameter) => {statements}
parameter => {statements}
parameter => expression
parameter => {return expression}

(param1, param2, ..., paramN) => {statements}
(param1, param2, ..., paramN) => expression
(param1, param2, ..., paramN) => {return expression}
```

- EXAMPLE

Let's look at some simple ways to apply this syntax:

```
1  'use strict';
2
3  const makeArray = (...values) => { return values };
4  console.log('Array:', makeArray(1, 2, 3, 4));
5  console.log('Array:', makeArray(1, 2, 3, 4, 5, 6));
6
7  const getSum = (a, b) => { return a + b };
8  console.log('1 + 2 =', getSum(1, 2));
9
10 const greeting = 'Hello, World.';
11 const capitalize = (myString) => { return myString.toUpperCase() };
12 console.log(greeting, '=>', capitalize(greeting));
```

OK

Output

Run

Using Arrow Functions

Let's look at some ways we can use arrow functions to make our code shorter.

- EXAMPLE

Sum the Elements of an Array

While we can certainly iterate over an array and sum each value, we can also use the *reduce* function.

```
1  'use strict';
2
3  const arr = [1, 2, 3, 4, 5];
4
5  const sum = arr.reduce(function (a, b) {
6      return a + b;
7  }, 0);
8
9  console.log('Array:', arr);
10 console.log('Sum:', sum);
```

Output

Run

Now, let's reduce the length of our code using an arrow function:

```
'use strict';

const arr = [1, 2, 3, 4, 5];

const sum = arr.reduce((a, b) => { return a + b }, 0);

console.log('Array:', arr);
console.log('Sum:', sum);
```

Output

Run

Let's further reduce it by getting rid of the `return`:

```
'use strict';

const arr = [1, 2, 3, 4, 5];

const sum = arr.reduce((a, b) => a + b, 0);

```

Output

Run

- EXAMPLE

Find the Length of Strings in an Array

Let's take an array of strings and use it to create a new array containing the respective lengths of its elements.

```
1 'use strict';
2
3 const arr = ['first', 'second', 'third', 'fourth', 'fifth'];
4
5 const len = arr.map(function(s) { return s.length });
6
7 console.log('Array:', arr);
8 console.log('Lengths:', len);
```

Output

Run

Now, let's reduce the length of our code using an arrow function:

```
const arr = ['first', 'second', 'third', 'fourth', 'fifth'];

const len = arr.map(s => s.length);

console.log('Array:', arr);
console.log('Lengths:', len);
```

Output

Run

- EXAMPLE

Find Array Elements Greater Than a Value

Let's find all the elements in an array that are greater than **3** and add them to a new array.

```
'use strict';

const arr = [1, 2, 3, 4, 5];

const greaterThan3 = arr.filter(function(a) {
    return a > 3;
});

```

Output

Run

Now, let's reduce the length of our code using an arrow function:

```
'use strict';

const arr = [1, 2, 3, 4, 5];

const greaterThan3 = arr.filter(a => a > 3);

console.log('Array:', arr);
console.log('Elements Greater Than 3:', greaterThan3);
```

Output

Run