Day 5: Arrow Functions



Arrow Functions in JavaScript

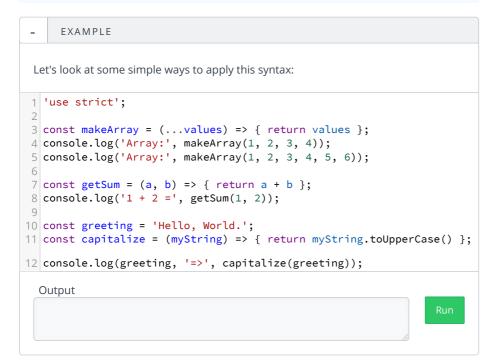
Arrow Functions

These expressions lexically bind the this value while using less syntax than a typical function expression. Arrow functions are always anonymous.

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}
```



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';
```

Table Of Contents

Arrow Functions

Using Arrow Functions



```
3 \text{ const arr} = [1, 2, 3, 4, 5];
5 const sum = arr.reduce(function (a, b) {
      return a + b;
7 }, 0);
8
9 console.log('Array:', arr);
10 console.log('Sum:', sum);
  Output
 Now, let's reduce the length of our code using an arrow function:
1 'use strict';
2
3 \text{ const arr} = [1, 2, 3, 4, 5];
5 const sum = arr.reduce((a, b) => { return a + b }, 0);
 7 console.log('Array:', arr);
8 console.log('Sum:', sum);
  Output
 Let's further reduce it by getting rid of the return:
 1 'use strict';
3 \text{ const arr} = [1, 2, 3, 4, 5];
5 const sum = arr.reduce((a, b) => a + b, 0);
7 console.log('Array:', arr);
8 console.log('Sum:', sum);
  Output
     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';
3 const arr = ['first', 'second', 'third', 'fourth', 'fifth'];
 5 const len = arr.map(function(s) { return s.length });
7 console.log('Array:', arr);
8 console.log('Lengths:', len);
  Output
```

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

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

Run
```

