

(WRITE **LESS** JAVASCRIPT)

9 JAVASCRIPT SHORTHANDS!

Using for Loops

Math Shorthands

Find Function

Join & Clone Arrays

String to Number

Destructuring

Short Circuit

Template Literals



Learn all in one place



FOR LOOP SHORTHAND

LONGHAND

```
const fruits = ['mango', 'peach', 'banana'];  
for (let i = 0; i < fruits.length; i++)  
    app.js
```

SHORTHAND

```
for (let fruit of fruits)  
    app.js
```

If you just wanted to access index, do:

```
for (let index in fruits)  
    app.js
```


SHORT CIRCUIT EVALUATION

Long Hand

app.js

```
let dbHost;  
if (process.env.DB_HOST) {  
    dbHost = process.env.DB_HOST;  
} else {  
    dbHost = 'localhost';  
}
```

Short Hand

```
const dbHost = process.env.DB_HOST || 'localhost';
```

FIND FUNCTION SHORTHAND

Long Hand

```
const pets = [
  { type: 'Dog', name: 'Max' },
  { type: 'Cat', name: 'Karl' },
  { type: 'Dog', name: 'Tommy' },
]

function findDog(name) {
  for(let i = 0; i < pets.length; ++i) {
    if(pets[i].type === 'Dog' &&
pets[i].name === name) {
      return pets[i];
    }
  }
}
```

app.js

Short Hand

```
pet = pets.find(pet => pet.type === 'Dog' && pet.name === 'Tommy');
console.log(pet); // { type: 'Dog', name: 'Tommy' }
```

app.js

MATH SHORTHANDS

Shorthand for **Math.floor()**

app.js

```
//Longhand  
Math.floor(4.9) === 4 //true  
  
//Shorthand  
~~4.9 === 4 //true
```

Shorthand for **Math.pow()**

app.js

```
//Longhand  
Math.pow(2,3); // 8  
Math.pow(2,2); // 4  
  
//Shorthand  
2**3 // 8  
2**4 // 4
```

JOINING & CLONING ARRAYS

Longhand

```
app.js
// joining arrays
const odd = [1, 3, 5];
const nums = [2, 4, 6].concat(odd);
console.log(nums); // [ 2, 4, 6, 1, 3, 5 ]

// cloning arrays
const arr = [1, 2, 3, 4];
const arr2 = arr.slice()
```

Shorthand (Using Spread operator)

```
app.js
// joining arrays
const odd = [1, 3, 5];
const nums = [2, 4, 6, ...odd];
console.log(nums); // [ 2, 4, 6, 1, 3, 5 ]

// cloning arrays
const arr = [1, 2, 3, 4];
const arr2 = [...arr];
```


DESTRUCTURING ASSIGNMENT

Longhand (*Not properly using ES6*)

```
const observable = require('mobx/observable');  
const action = require('mobx/action');  
const runInAction = require('mobx/runInAction');
```

```
const store = this.props.store;  
const form = this.props.form;  
const loading = this.props.loading;  
const errors = this.props.errors;  
const entity = this.props.entity;
```

app.js

Shorthand (ES6 used properly)

```
import { observable, action, runInAction } from 'mobx';
```

app.js

```
const { store, form, loading, errors, entity } = this.props;
```

//You can even assign your own variable names:

```
const { store, form, loading, errors, entity:contact } = this.props;
```


STRING INTO A NUMBER

Long Hand

app.js

```
const num1 = parseInt("100");  
const num2 = parseFloat("100.01");
```

Short Hand

app.js

```
const num1 = +"100";  
// converts to int data type  
  
const num2 = +"100.01";  
// converts to float data type
```


TEMPLATE LITERALS

Longhand

```
const welcome = 'You have logged in as ' + first + ' ' + last + '.'  
const db = 'http://' + host + ':' + port + '/' + database;
```

app.js

Shorthand

```
const welcome = `You have logged in as ${first} ${last}`;  
const db = `http://${host}:${port}/${database}`;
```

app.js

Other Shorthands " I'm assuming you all might know "

- **Arrow Functions**
- **Implicit return Shorthand**
- **Object Property Shorthand**
- **Decimal Base etc...**