Springboard

# Arrow Functions

## Arrow Functions

- Arrow functions are shorthand for function expressions
- They cannot be named and they only work as function expressions.
- They are ideal for shortening callbacks.

```
[1, 2, 3].forEach(function (n, idx) { console.log(n, idx); });
```

is the same as

```
[1, 2, 3].forEach((n, idx) => { console.log(n, idx); });
```

## Another Arrow Function!

```
[1, 2, 3, 4, 5].filter(function(n, idx) { return n % 2 === 0; });
```

is the same as

```
[1, 2, 3, 4, 5].filter((n, idx) => { return n % 2 === 0; });
```

## More Arrow Functions!

Arrow Functions have an implicit return if you leave out the curly braces

```
/* square everything */ let nums = [1, 2, 3]; let arrSquared = nums.map(n => n ** 2); //
[1, 4, 9]
```

## Gotcha with Arrow Function

You still need a return if it’s not on one line!

```
const multiply = (a,b) => { return a * b; }
```

You will sometimes see () around an arrow function - esspecially with modern frameworks!

```
const multiply = (a,b) => ( a * b );
```

If you want to return an object, make sure it's wrapped in () or on more than one line!

```
const makeInstructor = (name) => {name:"Colt"} makeInstructor() // undefined const
makeInstructor2 = (name) => ({name:"Colt"}) makeInstructor2() // {name: "Colt"}
```

## Arrow Functions & *this*

Arrow Functions do not have their own ***this*** context. If your function uses the keyword ***this***, be wary!

You should not be using arrow functions:

- In an object method
- When you need your own keyword ***this***

## Arrow Functions & *this* - no no's

Do not use arrow functions inside objects!

```
const student = { firstName: "Melissa", sayHi: () => { return "Hello " + this.firstName }
}
```

student.sayHi() // Hello undefined

The keyword ***this*** refers to the global object (window / global) not the student object - that's not good!

## Arrow Functions usefullness

Use arrow functions when you **don't** want the keyword ***this*** that a function normally creates.

Arrow functions are a replacement for the ***bind*** function in the example below.

```
const student = { firstName: "Melissa", sayHi: function() { setTimeout(() => {
console.log("Hello " + this.firstName) }, 1000) } }
```

student.sayHi() // Hello Melissa

## Arrow Functions Summary

- Can only be used as shorthand for anonymous function expressions
- Must put parentheses around parameters if there are 0 or 2+ parameters
- Return statement is implied if you leave out curly braces
- They do not make their own ***this***