Functions are one of the primary data structures in JavaScript; they've been around *forever*.

**Arrow functions**

ES6 introduces a new kind of function called the **arrow function**. Arrow functions are very similar to regular functions in behavior, but are quite different syntactically. The following code takes a list of names and converts each one to uppercase using a regular function:

**const** upperizedNames = ['Farrin', 'Kagure', 'Asser'].map(**function**(name) {

**return** name.toUpperCase();

});

The code below does the same thing *except* instead of passing a regular function to the map() method, it passes an arrow function. Notice the *arrow* in the arrow function ( => ) in the code below:

**const** upperizedNames = ['Farrin', 'Kagure', 'Asser'].map(

name => name.toUpperCase()

);

The only change to the code above is the code inside the map() method. It takes a regular function and changes it to use an arrow function.

***NOTE:****Not sure how map() works? It's a method on the Array prototype. You pass a function to it, and it calls that function once on every element in the array. It then gathers the returned values from each function call and makes a new array with those results. For more info, check out*[*MDN's documentation*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map)*.*

**Convert a function to an arrow function**

**const** upperizedNames = ['Farrin', 'Kagure', 'Asser'].map(**function**(name) {

**return** name.toUpperCase();

});

With the function above, there are only a few steps for converting the existing "normal" function into an arrow function.

* remove the function keyword
* remove the parentheses
* remove the opening and closing curly braces
* remove the return keyword
* remove the semicolon
* add an arrow ( => ) between the parameter list and the function body

Replay

Mute

Loaded: 100.00%

Remaining Time -0:00

1x

Playback Rate

Captions

Picture-in-Picture

Fullscreen

Converting a normal function into an arrow function.

**QUIZ QUESTION**

Take a look at the following code:

**const** names = ['Afghanistan', 'Aruba', 'Bahamas', 'Chile', 'Fiji', 'Gabon', 'Luxembourg', 'Nepal', 'Singapore', 'Uganda', 'Zimbabwe'];

**const** longNames = names.filter(**function**(name) {

**return** name.length > 6;

});

Which of the following choices does the same thing, but replaces .filter()'s function with an arrow function?

* 

const longNames = names.filter( function(name) => return name.length > 6; );

* 

const longNames = names.filter( return name.length > 6 );

* 

const longNames = names.filter( name => {name.length > 6} );

* const longNames = names.filter( name => name.length > 6 );