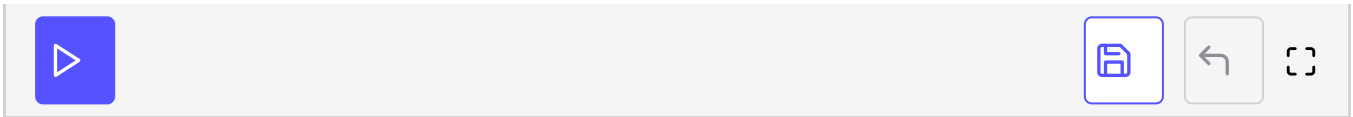
Arrow Functions in React

This lesson demonstrates the uses of Arrow Functions in React.

As mentioned in the previous lesson, Arrow Functions are a more concise way of writing a function in JavaScript. They are frequently used in React to make things more efficient and simpler (Event Handling, preventing bugs, etc.).

When teaching someone about React, I explain JavaScript arrow functions pretty early. They are one of JavaScript's language additions in ES6 which pushed JavaScript forward in functional programming.

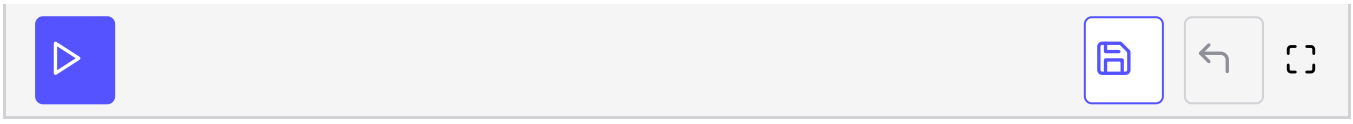
```
// JavaScript ES5 function
function getGreetingFunc() {
  return 'Welcome to JavaScript';
}

// JavaScript ES6 arrow function with body
const getGreetingArrow1 = () => {
  return 'Welcome to JavaScript';
}

// JavaScript ES6 arrow function without body and implicit return
const getGreetingArrow2 = () =>
  'Welcome to JavaScript';

console.log(getGreetingFunc());
console.log(getGreetingArrow1());
console.log(getGreetingArrow2());
```



Given below is an example to help you understand the importance of arrow functions in JavaScript. Let's say you have an array of student objects with two properties ID's and a boolean to check if that student is present or not. Let's say you want to filter out the students who are absent, one way to do this is:

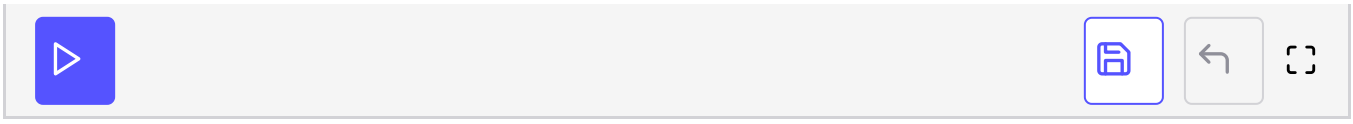
```
const students = [
  { ID: 1, present: true}
```

```
  { ID: 2, present: true},
  { ID: 3, present: false},
];

const presentStudents = students.filter(function(student){return student.present;});
console.log(presentStudents);
```



This statement can be easily transformed into a much a simpler and concise syntax by introducing an arrow function, like this:

```
const students = [
  { ID: 1, present: true},
  { ID: 2, present: true},
  { ID: 3, present: false},
];

const presentStudents = students.filter(student => student.present);
console.log(presentStudents);
```



Rules of thumb:

While writing an arrow function or transforming an existing function into arrow function, you must keep these points in mind:

- If there is only one statement inside function body then you can omit `return` keyword as well the curly braces
- Omit the `function` keyword in the beginning
- You can also get rid of parameter parentheses if you only have one parameter

> **Did you know?**
>
> *As arrow function do not rebind `this` so it makes it easier for developers to debug the code and prevent any errors caused by making use of `this` within callbacks. We will cover this in detail in the upcoming chapters.*

JavaScript arrow functions are often used in React applications for keeping the code concise and readable. I love them, teach them early, but always try to refactor my functions from JavaScript ES5 to ES6 functions along the way. At some point, when the differences between JavaScript ES5 functions and JavaScript ES6 functions are clear, I stick to the JavaScript ES6 way of doing it with arrow functions. However, I always see that too many different syntaxes can be overwhelming for React newcomers. So I try to make the different characteristics of JavaScript functions clear before going all-in using them in React.

Coding Challenge: Write a JavaScript arrow function

Rewrite the given square function using arrow functions. Again, remember to name it `square` or your code won't compile.

```
// Write-your-code-here

function square(n){
  return n*n;
}
```