

# Arrow Functions

Arrow functions were introduced in ECMAScript 6. They offer some shorthands and conveniences compared to “normal” functions. We will see the notation in this section, and we will discuss the differences at a later time.

## Relevant Links

- Exploring ES6, chapter 13

## Notes

**Arrow functions** are just another way to define a function. They are especially convenient for defining one-time functions, as we will explore in the next section. Their syntax is inspired from CoffeeScript.

Arrow functions differ from “normal” functions in how they treat the `this` variable, which we will discuss in a future section. This makes them extremely valuable as “callback functions”.

For now, just consider them as a shorthand notation for functions.

Here is a small example of an arrow function:

```
let sq1 = function(x) { return x * x; }; // A normal function
let sq2 = (x) => x * x; // Arrow function
let sq3 = x => x * x; // Can omit the parentheses if there is a single argument.
// Can do multiple steps, but then need braces and to "return"
let sq4 = x => { x = x + 1; return x * x; };
sq1(5);
sq2(5);
sq3(5);
sq4(4);
```