

Shorthand for Defining Operations in Objects

'Concise method syntax' - a new feature of ES6

As the name suggests, function declaration using the concise method syntax requires less typing, but that's not the only advantage.

Let's declare a `logArea` method in our `shape` object:

```
let shapeName = 'Rectangle', a = 5, b = 3;

let shape = {
  shapeName,
  a,
  b,
  logArea() { console.log( 'Area: ' + (a*b) ); },
  id: 0
};

shape.logArea();
```

Concise Method Syntax

Notice that in ES5, we would have to write `: function` between `logArea` and `()` to make the same declaration work. This syntax is called the *concise method syntax*. We first used the concise method syntax in [Chapter 4 - Classes](#).

Concise methods have not made it to the specification just to shave off 10 to 11 characters from the code. They also make it possible to access prototypes more easily.

This leads us to the next section on object prototype extensions and super calls.

