

# Fat Arrow Syntax

introduction to fat arrow syntax and its advantages

An arrow function expression has a shorter syntax than a regular function expression and does not have its own `this`, arguments, `super`, or `new.target`. Let's write an ES5 function to sum two numbers.

```
var sum = function( a, b ) {  
    return a + b;  
};  
console.log(sum(2, 3));
```



Using fat arrows ( `=>` ), we will rewrite the same function in two steps.

Step 1: replace the `function` keyword with a fat arrow.

```
var sum = ( a, b ) => {  
    return a + b;  
};  
console.log(sum(2, 3));
```



Step 2: if the return value of the function can be described by one expression, and the function body has no side-effects, then we can omit the braces and the `return` keyword.

```
var sum = ( a, b ) => a + b;  
console.log(sum(2, 3));
```



If a function has only one argument, parentheses are not needed on the left of the **fat arrow**:

```
var square = a => a * a;  
console.log(square(2));
```



Use cases of fat arrows: syntactic sugar, more compact way of writing functions.

In the next lesson, let's discuss context binding in ES6.