# Arrow functions, the basics

There’s another very simple and concise syntax for creating functions, that’s often better than Function Expressions.

It’s called “arrow functions”, because it looks like this:

1. let func = (arg1, arg2, ..., argN) => expression;

This creates a function func that accepts arguments arg1..argN, then evaluates the expression on the right side with their use and returns its result.

In other words, it’s the shorter version of:

1. let func = function(arg1, arg2, ..., argN) {
2. return expression;
3. };

Let’s see a concrete example:

1. let sum = (a, b) => a + b;
2. /\* This arrow function is a shorter form of:
3. let sum = function(a, b) {
4. return a + b;
5. };
6. \*/
7. alert( sum(1, 2) ); // 3

As you can see, (a, b) => a + b means a function that accepts two arguments named a and b. Upon the execution, it evaluates the expression a + b and returns the result.

* If we have only one argument, then parentheses around parameters can be omitted, making that even shorter.

For example:

1. let double = n => n \* 2;
2. // roughly the same as: let double = function(n) { return n \* 2 }
3. alert( double(3) ); // 6

* If there are no arguments, parentheses are empty, but they must be present:

1. let sayHi = () => alert("Hello!");
2. sayHi();

Arrow functions can be used in the same way as Function Expressions.

For instance, to dynamically create a function:

1. let age = prompt("What is your age?", 18);
2. let welcome = (age < 18) ?
3. () => alert('Hello!') :
4. () => alert("Greetings!");
5. welcome();

Arrow functions may appear unfamiliar and not very readable at first, but that quickly changes as the eyes get used to the structure.

They are very convenient for simple one-line actions, when we’re just too lazy to write many words.

## [Multiline arrow functions](https://javascript.info/arrow-functions-basics" \l "multiline-arrow-functions)

The arrow functions that we’ve seen so far were very simple. They took arguments from the left of =>, evaluated and returned the right-side expression with them.

Sometimes we need a more complex function, with multiple expressions and statements. In that case, we can enclose them in curly braces. The major difference is that curly braces require a return within them to return a value (just like a regular function does).

Like this:

1. let sum = (a, b) => { // the curly brace opens a multiline function
2. let result = a + b;
3. return result; // if we use curly braces, then we need an explicit "return"
4. };
5. alert( sum(1, 2) ); // 3

**More to come**

Here we praised arrow functions for brevity. But that’s not all!

Arrow functions have other interesting features.

To study them in-depth, we first need to get to know some other aspects of JavaScript, so we’ll return to arrow functions later in the chapter [Arrow functions revisited](https://javascript.info/arrow-functions).

For now, we can already use arrow functions for one-line actions and callbacks.

## [Summary](https://javascript.info/arrow-functions-basics" \l "summary)

Arrow functions are handy for simple actions, especially for one-liners. They come in two flavors:

1. Without curly braces: (...args) => expression – the right side is an expression: the function evaluates it and returns the result. Parentheses can be omitted, if there’s only a single argument, e.g. n => n\*2.
2. With curly braces: (...args) => { body } – brackets allow us to write multiple statements inside the function, but we need an explicit return to return something.

## [Tasks](https://javascript.info/arrow-functions-basics#tasks)

### [Rewrite with arrow functions](https://javascript.info/arrow-functions-basics" \l "rewrite-with-arrow-functions)

Replace Function Expressions with arrow functions in the code below:

1. function ask(question, yes, no) {
2. if (confirm(question)) yes();
3. else no();
4. }
5. ask(
6. "Do you agree?",
7. function() { alert("You agreed."); },
8. function() { alert("You canceled the execution."); }
9. );

**solution**

1. function ask(question, yes, no) {
2. if (confirm(question)) yes();
3. else no();
4. }
5. ask(
6. "Do you agree?",
7. () => alert("You agreed."),
8. () => alert("You canceled the execution.")
9. );

Looks short and clean, right?

## [Comments](https://javascript.info/arrow-functions-basics" \l "comments)

read this before commenting…