**Functions**

Quite often we need to perform a similar action in many places of the script.

For example, we need to show a nice-looking message when a visitor logs in, logs out and maybe somewhere else.

Functions are the main “building blocks” of the program. They allow the code to be called many times without repetition.

We’ve already seen examples of built-in functions, like alert(message), prompt(message, default) and confirm(question). But we can create functions of our own as well.

**[Function Declaration](https://javascript.info/function-basics" \l "function-declaration)**

To create a function we can use a *function declaration*.

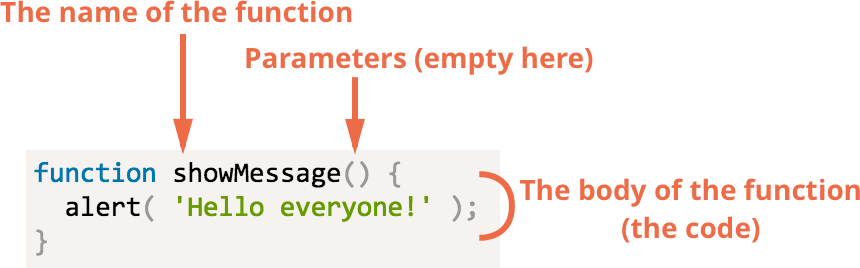
It looks like this:

function showMessage() {

alert( 'Hello everyone!' );

}

The function keyword goes first, then goes the *name of the function*, then a list of *parameters* in the brackets (empty in the example above) and finally the code of the function, also named “the function body”.



Our new function can be called by its name: showMessage().

For instance:

function showMessage() {

alert( 'Hello everyone!' );

}

showMessage();

showMessage();

The call showMessage() executes the code of the function. Here we will see the message two times.

This example clearly demonstrates one of the main purposes of functions: to evade code duplication.

If we ever need to change the message or the way it is shown – it’s enough to modify the code in one place: the function which outputs it.

**[Local variables](https://javascript.info/function-basics" \l "local-variables)**

A variable declared inside a function is only visible inside that function.

For example:

function showMessage() {

let message = "Hello, I'm JavaScript!"; // local variable

alert( message );

}

showMessage(); // Hello, I'm JavaScript!

alert( message ); // <-- Error! The variable is local to the function

**[Outer variables](https://javascript.info/function-basics" \l "outer-variables)**

A function can access an outer variable as well, for example:

let userName = 'John';

function showMessage() {

let message = 'Hello, ' + userName;

alert(message);

}

showMessage(); // Hello, John

The function has full access to the outer variable. It can modify it as well.

For instance:

let userName = 'John';

function showMessage() {

userName = "Bob"; // (1) changed the outer variable

let message = 'Hello, ' + userName;

alert(message);

}

alert( userName ); // John before the function call

showMessage();

alert( userName ); // Bob, the value was modified by the function

The outer variable is only used if there’s no local one. So an occasional modification may happen if we forget let.

If a same-named variable is declared inside the function then it *shadows* the outer one. For instance, in the code below the function uses the local userName, the outer one is ignored:

let userName = 'John';

function showMessage() {

let userName = "Bob"; // declare a local variable

let message = 'Hello, ' + userName; // Bob

alert(message);

}

// the function will create and use it's own userName

showMessage();

alert( userName ); // John, unchanged, the function did not access the outer variable

**Global variables**

Variables declared outside of any function, such as the outer userName in the code above, are called *global*.

Global variables are visible from any function (unless shadowed by locals).

Usually, a function declares all variables specific to its task, and global variables only store project-level data, so important that it really must be seen from anywhere. Modern code has few or no globals, most variables reside in their functions.

**Questions and Exercises**

1. Create a program that has two textboxes. One is labelled cm and the other feet. Add a button. When clicked call two functions that will convert and display the cm value to inches and the feet to yards. Use labels or appropriate html tags to display your answer.
2. Create a program that has three textboxes and a button. In the first textbox the user will enter a number, in the second another number and in the third a letter. When the button is clicked 3 functions will be called that determine if the first number is between 1 and 5, the second determines if the second number entered is between 1 and 10.5 and the last determines if the letter is between A and D.