

Functions

Reusable blocks of code.

The anatomy of a function declaration:

```
function keyword  function name  function parameters
      ↓           ↓             ↓
function name(param1, param2, ...) {
    statement 1;
    statement 2;
    statement 3;
}
return output;
```

function body

return keyword function output

Description of function syntax

- The function name and parameters must abide by JavaScript variable naming rules
- The list of parameters is optional
- The `return` statement is optional
- If a function doesn't have a `return` statement, its return value will be `undefined`
- The function body isn't run until you call the function.
Syntax: `functionName(param1, param2, ...)`

Examples of functions

Example 1: Function with no parameters

```
// declare the function
function greetUser() {
  console.log("Hello, user!");
}

// call the function
greetUser();

// 'Hello, user!'
```

Example 2: Function with a parameter

```
// declare the function
function greetUser(name) {
  console.log(`Hello, ${name}!`);
}

// arguments passed to the function are substituted for parameters
greetUser("Elise");
// 'Hello, Elise!'

// if no argument is passed, the parameter will be undefined
greetUser();
// 'Hello, undefined!'
```

Example 3: Function with no return value

```
// declare the function
function getFullName(firstName, lastName) {
  return `${firstName} ${lastName}`;
}

// assigning the return value of the function to a variable
const name = getFullName("Elise", "Bouer");

console.log(name);
// 'Elise Bouer'
```