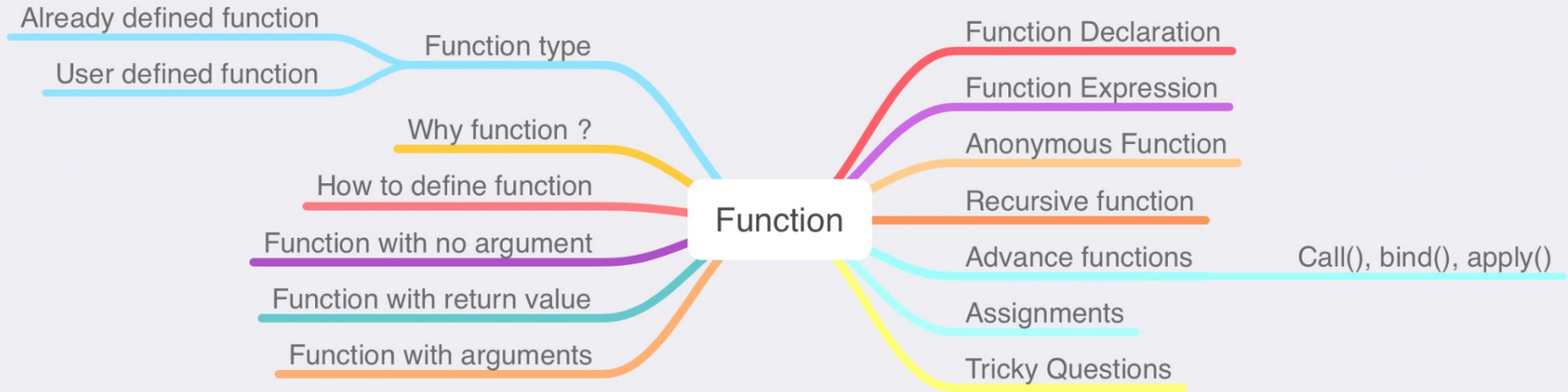


Functions

By Gajanan Kharat - CODEMIND Technology

Contact us 8600 2550 66

Functions



What is Block

Piece of code written inside { }

What is Function?

A block of code designed to perform a particular task.

Why Function?

- To improve readability of code
- It helps for reusability of code

Function type

-
- Already defined function
 - User defined function

1. Function with no argument and no return type

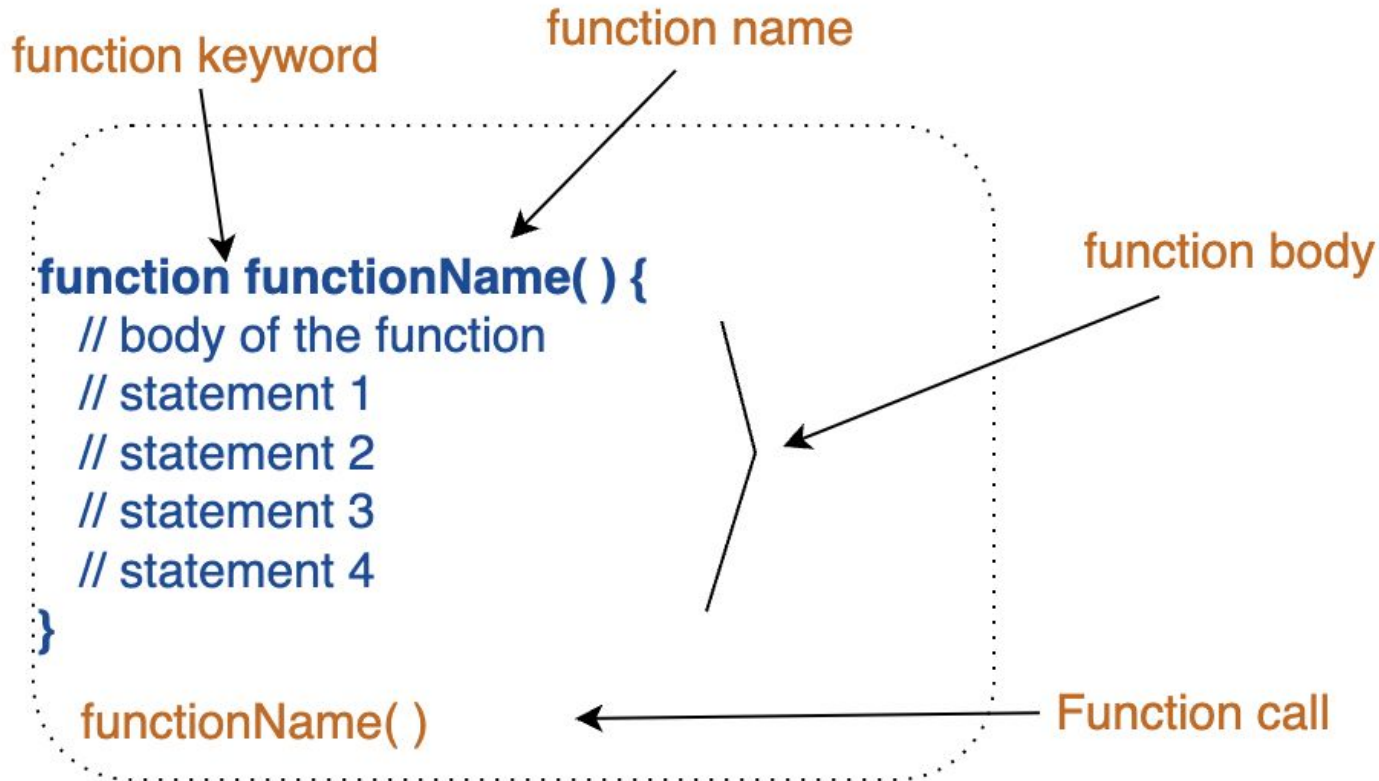
- Keyword 'function' is used to define function
- A Function is executed when we call or invoke it

Function body: Anything that is return inside { } → curly braces

Rule:

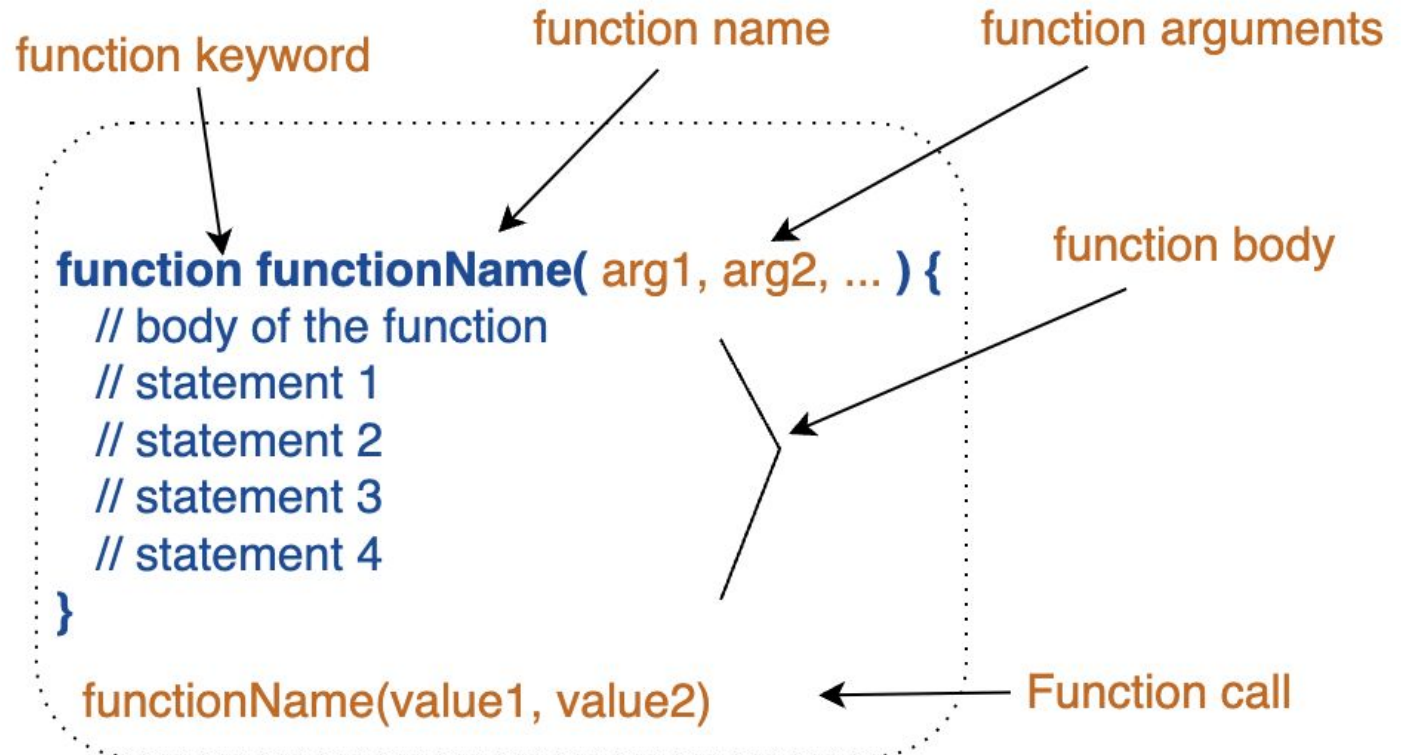
- Function name - Camel case
- Function arguments: Should not have more than 6 to 7
- Total number of lines inside function: should not have more than screen size

1. Function with no argument and no return type



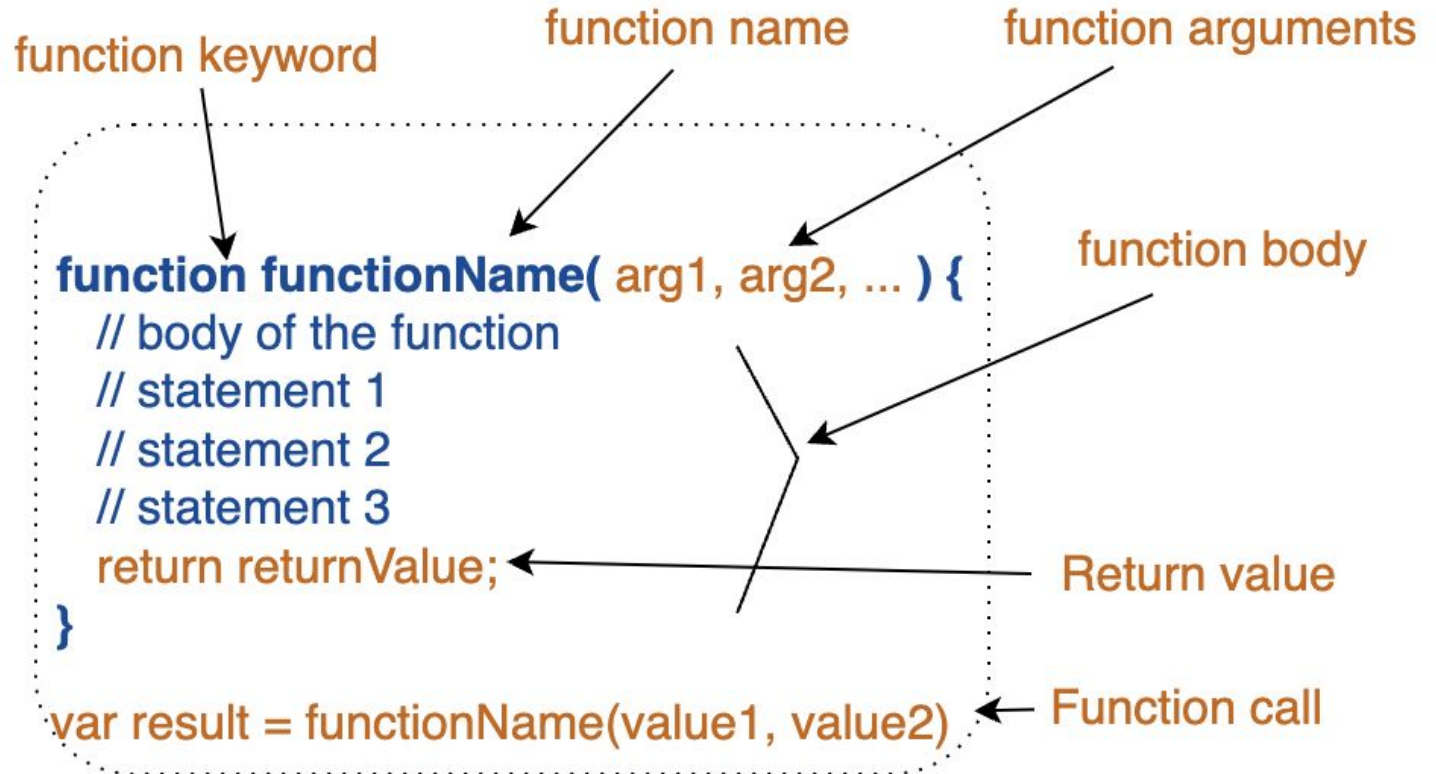
2. Function with arguments and no return value

Function arguments are mentioned inside () parenthesis



3. Function with arguments and return type

'return' keyword is used to return value



4 Function

A function is simply a bunch of code bundled in a section. This bunch of code ONLY runs when the function is called. Functions allow for organizing code into sections and code reusability.

Using a function has ONLY two parts. (1) Declaring/defining a function, and (2) using/running a function.

Name of function

That's it, it's just a name you give to your function.

Tip: Make your function names descriptive to what the function does.

Return (optional)

A function can optionally spit-out or "return" a value once it's invoked. Once a function returns, no further lines of code within the function run.

Invoke a function

Invoking, calling or running a function all mean the same thing. When we write the function name, in this case `someName`, followed by the brackets symbol `()` like this `someName()`, the code inside the function gets executed.

// Function declaration / Function statement

```
function someName(param1, param2){
```

```
// bunch of code as needed...
```

```
var a = param1 + "love" + param2;
```

```
return a;
```

```
}
```

// Invoke (run / call) a function

```
someName("Me", "You")
```

Parameters / Arguments (optional)

A function can optionally take parameters (a.k.a arguments). The function can then use this information within the code it has.

Code block

Any code within the curly braces `{ ... }` is called a "block of code", "code block" or simply "block". This concept is not just limited to functions. "if statements", "for loops" and other statements use code blocks as well.

Passing parameter(s) to a function (optional)

At the time of invoking a function, parameter(s) may be passed to the function code.

Possible combination

1. Function with no arguments and return value
2. Function with arguments and no return value
3. Function with arguments and return value

Assignments 02: File→ 03_funAssig01.js Please use meaningful message while logging on console

1. Write any two functions with no arguments and no return type as per your choice & log message on console
2. Write a function → personalDetails with arguments namely firstName, lastName, collegeName arguments and log on console inside a function. Don't forget to call it please
3. Write A function with two args and should swap the passed values and log on console before swap and after swap values inside function itself. (Func with args and no return)
 - 3.1. Function name: swapValues()
 - 3.2. Invoke or call the function swapValues() with values "Virat" and "Anushka";
 - 3.3. Invoke or call the same function swapValues() with values 1000 and 2000;
4. WAF with 3 parameters or arguments and it should log the addition.
 - 4.1. Function name: addThreeValues()
 - 4.2. Call the function for values → 10.23, 600, 40
 - 4.3. Invoke the same function for values "Hello", " Good", " Morning"

Assignments: Normal Function

1. Write a function 'bankDetails' with 4 arguments bankName, accountNum, location, pinCode
 - 1.1. Log these values on same line on console as it with meaningful message
 - 1.2. Invoke function as
 - 1.2.1. `bankDetails("CITI Bank", 3456782345, 'Pune', 431202);`
 - 1.2.2. `bankDetails("Axis Bank", 7856782345, Mumbai, 441202);`
 - 1.2.3. `bankDetails("HDFC Bank", 8956782345, 'Pune', 631202, "Open");`