## Defining Functions in LISP

A function is a group of statements that together perform a task.

You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division usually is so each function performs a specific task.

The macro named **defun** is used for defining functions. The **defun** macro needs three arguments −

* Name of the function
* Parameters of the function
* Body of the function

Syntax for defun is −

(defun name (parameter-list) "Optional documentation string." body)

Let us illustrate the concept with simple examples.

### Example 1

Let's write a function named *averagenum* that will print the average of four numbers. We will send these numbers as parameters.

Create a new source code file named main.lisp and type the following code in it.

[Live Demo](http://tpcg.io/h23YdU" \t "https://www.tutorialspoint.com/lisp/_blank)

(defun averagenum (n1 n2 n3 n4)

(/ ( + n1 n2 n3 n4) 4))(write(averagenum 10 20 30 40))

When you execute the code, it returns the following result −

25

### Example 2

Let's define and call a function that would calculate the area of a circle when the radius of the circle is given as an argument.

Create a new source code file named main.lisp and type the following code in it.

[Live Demo](http://tpcg.io/iHu0yn" \t "https://www.tutorialspoint.com/lisp/_blank)

(defun area-circle(rad)

"Calculates area of a circle with given radius"

(terpri)

(format t "Radius: ~5f" rad)

(format t "~%Area: ~10f" (\* 3.141592 rad rad)))(area-circle 10)

When you execute the code, it returns the following result −

Radius: 10.0

Area: 314.1592

Please note that −

You can provide an empty list as parameters, which means the function takes no arguments, the list is empty, written as ().

LISP also allows optional, multiple, and keyword arguments.

The documentation string describes the purpose of the function. It is associated with the name of the function and can be obtained using the **documentation** function.

The body of the function may consist of any number of Lisp expressions.

The value of the last expression in the body is returned as the value of the function.

You can also return a value from the function using the **return-from** special operator.

Let us discuss the above concepts in brief. Click following links to find details −

[Optional Parameters](https://www.tutorialspoint.com/lisp/lisp_optional_parameters.htm)

[Rest Parameters](https://www.tutorialspoint.com/lisp/lisp_rest_parameters.htm)

[Keyword Parameters](https://www.tutorialspoint.com/lisp/lisp_keyword_parameters.htm)

[Returning Values from a Function](https://www.tutorialspoint.com/lisp/lisp_returning_values_functions.htm)

[Lambda Functions](https://www.tutorialspoint.com/lisp/lisp_lambda_functions.htm)

[Mapping Functions](https://www.tutorialspoint.com/lisp/lisp_mapping_functions.htm)