



# Python Functions

[< Previous](#)[Next >](#)

A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

A function can return data as a result.

## Creating a Function

In Python a function is defined using the `def` keyword:

### Example

```
def my_function():  
    print("Hello from a function")
```

## Calling a Function

To call a function, use the function name followed by parenthesis:

### Example

```
def my_function():  
    print("Hello from a function")  
  
my_function()
```

[Run example »](#)

## Parameters

Information can be passed to functions as parameter.

Parameters are specified after the function name, inside the parentheses. You can add as many parameters as you want, just separate them with a comma.

The following example has a function with one parameter (fname). When the function is called, we pass along a first name, which is used inside the function to print the full name:

### Example

```
def my_function(fname):  
    print(fname + " Refsnes")  
  
my_function("Emil")  
my_function("Tobias")  
my_function("Linus")
```

[Run example »](#)

## Default Parameter Value

The following example shows how to use a default parameter value.

If we call the function without parameter, it uses the default value:

### Example

```
def my_function(country = "Norway"):  
    print("I am from " + country)  
  
my_function("Sweden")  
my_function("India")
```

```
my_function()  
my_function("Brazil")
```

Run example »

## Return Values

To let a function return a value, use the `return` statement:

### Example

```
def my_function(x):  
    return 5 * x  
  
print(my_function(3))  
print(my_function(5))  
print(my_function(9))
```

Run example »

## Recursion

Python also accepts function recursion, which means a defined function can call itself.

Recursion is a common mathematical and programming concept. It means that a function calls itself. This has the benefit of meaning that you can loop through data to reach a result.

The developer should be very careful with recursion as it can be quite easy to slip into writing a function which never terminates, or one that uses excess amounts of memory or processor power. However, when written correctly recursion can be a very efficient and mathematically-elegant approach to programming.

In this example, `tri_recursion()` is a function that we have defined to call itself ("recurse"). We use the `k` variable as the data, which decrements (`-1`) every time we recurse. The recursion ends when the condition is not greater than 0 (i.e. when it is 0).

To a new developer it can take some time to work out how exactly this works, best way to find out is by testing and modifying it.

## Example

### Recursion Example

```
def tri_recursion(k):  
    if(k>0):  
        result = k+tri_recursion(k-1)  
        print(result)  
    else:  
        result = 0  
    return result  
  
print("\n\nRecursion Example Results")  
tri_recursion(6)
```

Run example »

## Test Yourself With Exercises

### Exercise:

Create a function named `my_function`.

```
 :  
    print("Hello from a function")
```

Submit Answer »

[Start the Exercise](#)

◀ Previous

Next ▶

## COLOR PICKER



## HOW TO

Tabs  
Dropdowns  
Accordions  
Side Navigation  
Top Navigation  
Modal Boxes  
Progress Bars  
Parallax  
Login Form  
HTML Includes  
Google Maps  
Range Sliders  
Tooltips  
Slideshow  
Filter List  
Sort List

## SHARE



## CERTIFICATES

HTML  
CSS  
JavaScript  
PHP  
jQuery

Bootstrap  
XML

[Read More »](#)

---

---

[REPORT ERROR](#)

[PRINT PAGE](#)

[FORUM](#)

[ABOUT](#)

---

## Top 10 Tutorials

[HTML Tutorial](#)  
[CSS Tutorial](#)  
[JavaScript Tutorial](#)  
[How To Tutorial](#)  
[W3.CSS Tutorial](#)  
[Bootstrap Tutorial](#)  
[SQL Tutorial](#)  
[PHP Tutorial](#)  
[jQuery Tutorial](#)  
[Python Tutorial](#)

## Top 10 References

[HTML Reference](#)  
[CSS Reference](#)  
[JavaScript Reference](#)  
[W3.CSS Reference](#)  
[Bootstrap Reference](#)  
[SQL Reference](#)  
[PHP Reference](#)  
[HTML Colors](#)  
[jQuery Reference](#)  
[Python Reference](#)

## Top 10 Examples

[HTML Examples](#)  
[CSS Examples](#)  
[JavaScript Examples](#)  
[How To Examples](#)  
[W3.CSS Examples](#)  
[Bootstrap Examples](#)  
[PHP Examples](#)

[jQuery Examples](#)  
[Angular Examples](#)  
[XML Examples](#)

## Web Certificates

[HTML Certificate](#)  
[CSS Certificate](#)  
[JavaScript Certificate](#)  
[jQuery Certificate](#)  
[PHP Certificate](#)  
[Bootstrap Certificate](#)  
[XML Certificate](#)

---

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2019 by Refsnes Data. All Rights Reserved.

Powered by W3.CSS.

