[Functions](http://www.pythonforbeginners.com/python-functions-cheat-sheet/) ***2***

What is a function in Python?

A function is something you can call (possibly with some parameters, the things

you put in the parentheses), which performs an action and returns a value.

Why should I use functions?

Reduce code tasks into simples tasks

Can easier split up the code between developers

Elimination of duplicate code

Reuse code

Get a good structure of the code

Easier debugging.

What are the rules of functions?

A function in Python must be defined before it’s used.

Create a function by using the keyword **“def”** followed by the functions name and

the parentheses ().

The function has to be **named** plus specify what parameter it has (if any)

A function can use a number of arguments and every argument is responding to a

parameter in the function.

The **keyword** "def" is required and must be in lowercase.

The name can be anything you like.

The end of the line has to end with a **colon** (:)

The function often ends by returning a value using return.

The code inside the function must be **indented**

The function is used when called.

Parameters (Arguments)

Parameters (also known as arguments) are inputs to functions.

All parameters (arguments) in the Python language are passed by reference.

There are some different types of parameters, two of them are:

**Position**

Positional arguments do not have keywords and are assigned first.

**Keyword**

Keyword arguments have keywords and are assigned second, after positional

arguments.

When you call a function you make a decision to use position or keyword or

a mixture.

You can choose to do all keywords if you want.

Call

A call of a function a procedure or a function must have parenthesis.

Between the parenthesis, you can have one or more parameter values,

but it can also be empty.

The first thing that happens is that the functions parameters get their values,

and then continue with the rest of the code in the function.

When a functions value is done, it returns it to the call.

**Function call with one parameter**

normal = celsius\_to\_fahrenheit(c\_temp)

**Function call without parameters**

x = input()

**Procedure call with two parameters**

rectangle(20,10)

**Procedure call without parameters**

say\_hello()

Remember that when Python makes a call, the function must already be defined.

Return

While the parameters are the inputs to functions, the Return values are the

outputs.

The return keyword is used to return values from a function.

The function will exit upon the return command.

(all code after that will be ignored)

A function may or may not return a value.

If a function does not have a return keyword, it will send a None value.

Create functions in Python

First thing when creating a function in Python is to define it and give it a

name (possibly with some parameters between the parentheses)

**define it and give it a name**

>> def name()

**Create directions for the function**

>> commands

**Call the function**

>> name()

You can send values to your function, by creating variables in the definition.

(These variables only works inside this particular functions)

**Let's see an example:**

The first line defines the function numbers()

The function has two parameters **num1** and **num2**

The second lines makes the addition of num1 and num2

**def numbers(num1, num2):**

**print num1+num2**

If this definition is at the beginning of the program, all we have to do is

write **def numbers(1,2)** to send the values to the function.

We do that by placing values in the function call.

You can also define mathematical functions. This takes the square root of a

number:

**def square(x):**

**return x\*x**

Let's see an example how to create a simple function any parameters.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | def name():      # Get the user's name.      name = raw\_input('Enter your name: ')        # Return the name.      return name    name() |

In this second example show how an argument is passed to a function:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | def even(number):      if number % 2 == 0:          return True        else:          return False    print even(10) |

Examples

If you haven't read the [Non-Programmer's Tutorial for Python](http://en.wikibooks.org/wiki/Non-Programmer) yet, read it.

It's a great resource for learning Python.

This example which converts temperatures is a good example of how to use

functions.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | def print\_options():      print "Options:"      print " 'p' print options"      print " 'c' convert from celsius"      print " 'f' convert from fahrenheit"      print " 'q' quit the program"    def celsius\_to\_fahrenheit(c\_temp):      return 9.0 / 5.0 \* c\_temp + 32    def fahrenheit\_to\_celsius(f\_temp):      return (f\_temp - 32.0) \* 5.0 / 9.0    choice = "p"    while choice != "q":        if choice == "c":          temp = input("Celsius temperature: ")          print "Fahrenheit:", celsius\_to\_fahrenheit(temp)        elif choice == "f":          temp = input("Fahrenheit temperature: ")          print "Celsius:", fahrenheit\_to\_celsius(temp)        elif choice != "q":          print\_options()        choice = raw\_input("option: ") |

I hope that you like this cheat sheet and that you have learned something today.