

Functions in a Nutshell

We found more than one version of [Functions in a Nutshell](#). You can view the alternate [here](#).



A **function** is reusable code that takes arguments and returns results. There are two main types of functions (aside from recursive and iterative):

1. **Built-in Functions** are functions that are already built into python. They can be invoked without any pre definition.
2. **User Defined Functions** are functions that are defined and created by the users. They are created by using the def keyword in the following manner:

```
def Userdefined_Func():
```

check functions.py for more examples

What can you do with functions?

You basically can recall the same code at different intervals passing different arguments for different instances. For example, let's say for some reason your code requires you to add two numbers A LOT. You could simply create the function below, invoke it and pass those two numbers through it every time you see fit:

```
def Add2Nums(A,B)
    s = A+B
    return s
Add2Nums(5,6)
>>>11
```

Function Rules

1. The same name ruling that applies to variables applies to functions.
2. You can not use a variable defined inside a function on a global scope (outside the function)
3. Variables defined outside the function can be used inside the function if you define them as global within the function.

(I doubt this will be asked on exams but just incase)

There is also something insane called "recursive functions" which literally makes 0 absolute sense but makes sense because it makes no sense...if that makes sense. It was not explained in class but was in the 2019 Or 2016 past questions. Learning it was a headache and Lord knows I can't explain it. But yeah. That way.