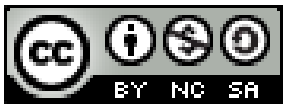
An Introduction to STEM Programming with Python 3 – Chapter 8 Definitions – AKA Functions

By
James M. Reneau, Ph.D.
jreneau@shawnee.edu

Shawnee State University

Version 2020-03-24a



In this video we will cover:

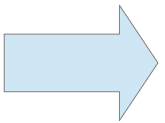
- Creating your own definition
- Passing values and references to a function
- Returning values from a function

Simple Definition

- A definition "def" allows you to create a block of code that:
  - Can be reused – in a program and possibly in many programs
  - Has it's own variables
  - Simplifies programming and maintenance
- Starts with a "def" followed by a name
  - Need parentheses after name
    - We will use them later
  - Needs a ":" after parentheses
  - Statements are an indented suite.

Simple Definition

```
1 def doit():
2     print("I did it.")
3
4 doit()
5 doit()
6 doit()
```



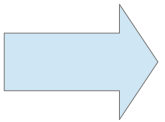
```
I did it.
I did it.
I did it.
```

Passing Values to Def

- We can also pass values from our code to the definition
- Values are passed as a reference (the variable points to the original value)
- Variables you create in a "def" are local to the function
  - They do not change the value at the calling level

Passing Values to Def

```
1  def calcprice(qty, base):
2      price = qty * base
3      if price >= 100:
4          price = price * .90
5      elif qty >= 10:
6          price = price * .95
7      print(price)
8
9  print("1 foozap costs")
10 calcprice(1, 23.45)
11 print("5 foozaps costs")
12 calcprice(5, 23.45)
13
14 print("10 tinyzips costs")
15 calcprice(10, .38)
16 print("100 tinyzips costs")
17 calcprice(100, .38)
```



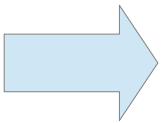
```
1 foozap costs
23.45
5 foozaps costs
105.525
10 tinyzips costs
3.61
100 tinyzips costs
36.1
```

Passing Values to Def – Mutable References

- Values are passed as a reference (the variable points to the original value)
- Lists, objects, dictionaries and many types the changes made in the def will change value outside.
- Be careful when passing this type of thing to a def.
- Integer, float, string, Boolean, and tuples are immutable (they do not change outside if changed inside)

Passing Values to Def - Mutable

```
1  def f(s):
2      s.upper()
3
4  def a(l):
5      l.append("hi")
6
7  x = "hello"
8  f(x)
9  print(x)
10
11 y = [1,2,3]
12 a(y)
13 print(y)
14
```



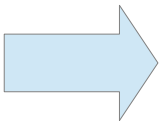
```
hello
[1, 2, 3, 'hi']
```

Returning Values from Def

- Definitions can also return a value using the "return" statement.
- Can return almost anything:
  - Integer, float, string, list, dictionary, object...
- You can:
  - save the value in a variable
  - use it in an expression
  - It is lust like any other value

Returning Values from Def

```
1  def calcprice(qty, base):
2      price = qty * base
3      if price >= 100:
4          price = price * .90
5      elif qty >= 10:
6          price = price * .95
7      return round(price,2)
8
9  total = 0
10 this = calcprice(1, 23.45)
11 total = total + this
12 print("1 foozap costs", this)
13
14 this = calcprice(12, 1.23)
15 total = total + this
16 print("12 zidoflaps costs", this)
17
18 print("your total order is", total)
```



```
1 foozap costs 23.45
12 zidoflaps costs 14.02
your total order is 37.47
```

Thank you

- This presentation is Copyright 2020 by James M. Reneau PhD.
- Contact me at jreneau@shawnee.edu
- This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.