Python 101 Tutorials

Complete each section with activity

Contents

[1: Setup Python 1](#_Toc111535592)

[2: Setup VSCode + Github Repo + Hello Python 2](#_Toc111535593)

[3: Variables in Python 3](#_Toc111535594)

[4: looping Python 4](#_Toc111535595)

[5. Loading up libraries and taking script input 5](#_Toc111535596)

[6: Functions : back to Fib 6](#_Toc111535597)

[7: Bonus Activity: CLI Library fun 7](#_Toc111535598)

# 1: Setup Python

**Required Knowledge:**  
Follow instructions to install and setup Python

<https://www.python.org/downloads/>

<https://docs.python.org/3.9/using/windows.html>

**Activities:**

Test is installed correctly

$ python --version

Python 3.9.10

# 2: Setup VSCode + Github Repo + Hello Python

**Required Knowledge:**  
Follow instructions to setup Python plugins in VSCode

<https://code.visualstudio.com/docs/languages/python>

Activities:  
Setup github repo for all your tutorials (if you haven’t already)

**Activities:**  
Complete the VSCode hello world python activity

<https://code.visualstudio.com/docs/python/python-tutorial>

Troubleshooting:

Ps1 files might be disabled on your locally machine  
Text

Description automatically generated  
  
Run powershell as a admin

And run  
Set-ExecutionPolicy -ExecutionPolicy Undefined -Scope LocalMachine  
Graphical user interface, text, email

Description automatically generated

# 3: Variables in Python

**Required Knowledge:**Read through documentation on how variables / operators work

<https://python.swaroopch.com/basics.html>

<https://python.swaroopch.com/op_exp.html>

Watch the brilliant Corey Schafer data types videos (feel free to follow what he’s doing on screen)  
<https://www.youtube.com/watch?v=k9TUPpGqYTo> Text Data

<https://www.youtube.com/watch?v=khKv-8q7YmY> Number Data

<https://www.youtube.com/watch?v=W8KRzm-HUcc> List Data

<https://www.youtube.com/watch?v=daefaLgNkw0> dictionary key-value data

**Activities:**

Implement simple “what age were you in when the London eye opened” python program

Steps:

* ask the user for their name\*
* ask the user for their age\*
* prints out “Hello <name> (age <age>)”
* calculate the age the person was in 1999  
  (can assume year today 2022)
* prints out “You were <xxx> years old when the london eye opened in 1999”

\* use the function “xxx = input()”

<https://docs.python.org/3/library/functions.html#input>   
<https://docs.python.org/3/tutorial/inputoutput.html>

# 4: looping Python

**Required Knowledge:**  
Read through documentation on how variables / operators and loops work

<https://python.swaroopch.com/control_flow.html>

Read through documentation on how for loops work  
<https://wiki.python.org/moin/ForLoop>   
  
Watch the brilliant Corey Schafer control videos

<https://www.youtube.com/watch?v=DZwmZ8Usvnk> If / conditionals  
<https://www.youtube.com/watch?v=6iF8Xb7Z3wQ> For/While loops

**Activities:**

Implement a wonderful game of Fizz buzz

print out the numbers from 1 to 20, with Three exceptions :

when the number is dividable by 3 print out ‘fizz’ instead of the number

when the number is dividable by 4 print out ‘buzz’ instead of the number

and when the number is dividable by 3 and 4 print out ‘fizzbuzz’ instead of the number

<https://en.wikipedia.org/wiki/Fizz_buzz>

Hints:  
Especially look at the modulus % operator (6 % 3 == 0, 7 % 3 == 1)

<https://python.swaroopch.com/op_exp.html>

$ ./fizz-buzz.py

1

2

fizz

buzz

5

fizz

7

buzz

fizz

10

11

fizzbuzz

13

14

fizz

buzz

17

fizz

19

20

# 5. Loading up libraries and taking script input

**Required Knowledge:**  
Read through modules document  
<https://www.youtube.com/watch?v=CqvZ3vGoGs0>

Read through **argv** entry in **sys** inbuilt documentation  
<https://docs.python.org/3/library/sys.html>   
(  
ps **sys** is one of many libraries inbuilt into python   
see <https://docs.python.org/3/library/index.html>   
)

Read through tutorial how to use **argv**

<https://www.tutorialspoint.com/python/python_command_line_arguments.htm>

**Activities:**

Update your fizz buzz application to accept three cli parameters, 1st is the number to loop two, 2nd which will be the dividable number that print out fizz, and the 3rd the dividable number to print out buzz

Aka

$ ./fizz-buzz.py 20 6 7

1

2

fizz

buzz

5

fizz

7

buzz

fizz

10

11

fizzbuzz

13

14

fizz

buzz

17

fizz

19  
20

# 6: Functions : back to Fib

**Required Knowledge:**  
Read functions guides, functions 101  
<https://www.freecodecamp.org/news/functions-in-python-a-beginners-guide/>   
<https://www.tutorialspoint.com/python/python_functions.htm>   
  
Watch the brilliant Corey Schafer function video  
<https://www.youtube.com/watch?v=9Os0o3wzS_I>

**Activities:**  
  
Using functions, and all you’re learnt so far

Create a script that takes number as the first argument, and use that number to return the Fibonacci number of that list position, so for 5 would be 8 aka fib number 5 (where fib numbers are 1,2,3,5,8,13,21…)

Aka 3rd Fibonacci number is 3, 7th Fibonacci number is 21

<https://www.mathsisfun.com/numbers/fibonacci-sequence.html>

**# ./fibonacci.py 7**

21

# 7: Bonus Activity: CLI Library fun

BONUS ACTIVITY, ATTEMPT THIS TIME BOXED TO NO MORE THAN 3-4 HOURS, DO NOT WORRY IF YOU FAIL, IT IS VERY HARD BUT IMPORTANT TO TRY CLI LIBRARY CODING  
  
**Required Knowledge:**

Install pip if not already installed (python library manager)  
$ pip --version

pip 21.2.4

<https://pip.pypa.io/en/stable/cli/pip_install/>

Read pip documentation

<https://packaging.python.org/en/latest/tutorials/installing-packages/>   
  
Watch the brilliant Corey Schafer libraries video  
<https://www.youtube.com/watch?v=CqvZ3vGoGs0>   
  
  
Read through the click library documentation

<https://click.palletsprojects.com/en/8.1.x/>

<https://pypi.org/project/click/>

**Activities:**

Update your previous **fibonacci.py** to use click for command line flags

1) Install the excellent click library

$ pip install click  
  
2) Update Fibonacci.py to import and use the excellent click library

3) Update the arguments to be flags, aka --count=X --fizz=X --buzz=X

4) Update the arguments to have defaults, so if flags aren’t send it’ll default to count=20, fizz=3, buzz=4

4 bonus) Add --help and --version flags to help users of your application

5) Test out your new click app  
$ ./**Fibonacci.py** --count=100 --fizz=21 --buzz=11