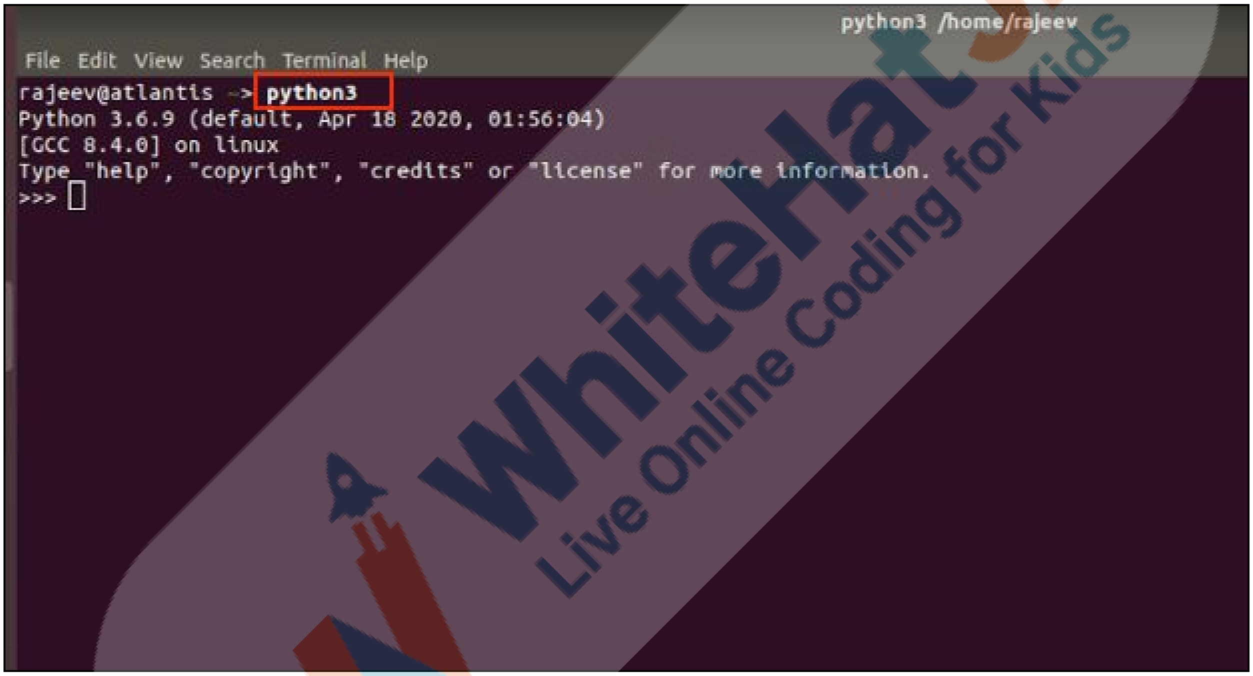
Topic	Introduction to Python	
Class Description	**Students install python on their system. They learn the basic syntax of the python programming language and how to run a python program.** **Student writes code that takes some input from the user and tells the number of words written by the user.**	
Class	**C97**	
Class time	**45 mins**	
Goal	● Learn basic syntax of python language ● Write a program to count the number of words input by the user	
Resources Required	● Teacher Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ● Student Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	**Warm Up** **Teacher-led Activity** **Student-led Activity** **Wrap up**	**15 min** **15 min** **5 min**

Class Steps	Say	Do
Step 1: **Warm Up** **(15 min)**	Javascript which we have used so far is mostly used to render frontend or what you see in an application. For example, we used javascript to render games in the browser and the app interface elements for mobile apps. Python, on the other hand, can be used to do almost anything which a	-

	computer can do! It can also run on any kind of computer. It is called a General Purpose Programming Language. We will learn how to leverage the power of Python in this and upcoming classes.	
Teacher Initiates Screen Share		
Step 2: Teacher-led Activity (15 min)	Excellent! Python has similar constructs to js. It just has a different syntax - or way of writing these constructs. However, now that you know js - it is going to be super easy to pick up python. Few hours of getting familiar with python will be enough to get you to start doing super awesome stuff using python.	-
	Let's learn how to install and run python programs. *<Instructions for installing Python for Windows, MacOS and Linux>* *<Teacher can download python from Teacher Activity 1>* *<Teacher opens the link from Teacher Activity 4>* *<Student Activity 1 & Student Activity 3>* *Teacher helps the student install python on their machine.*	*The student installs python on their system.*

<table>
<tr><td></td><td>There are two ways to run a python program. One is through a python shell (also called repl: read, evaluate, print and loop).
Python shell only takes a single expression, evaluates it, prints it as a result to the user and then gets ready for another input.

<i>Teacher opens the terminal and types python3 to enter into the python shell.</i></td><td><i>The student learns how to open a python shell.</i></td></tr>
<tr><td colspan="3">

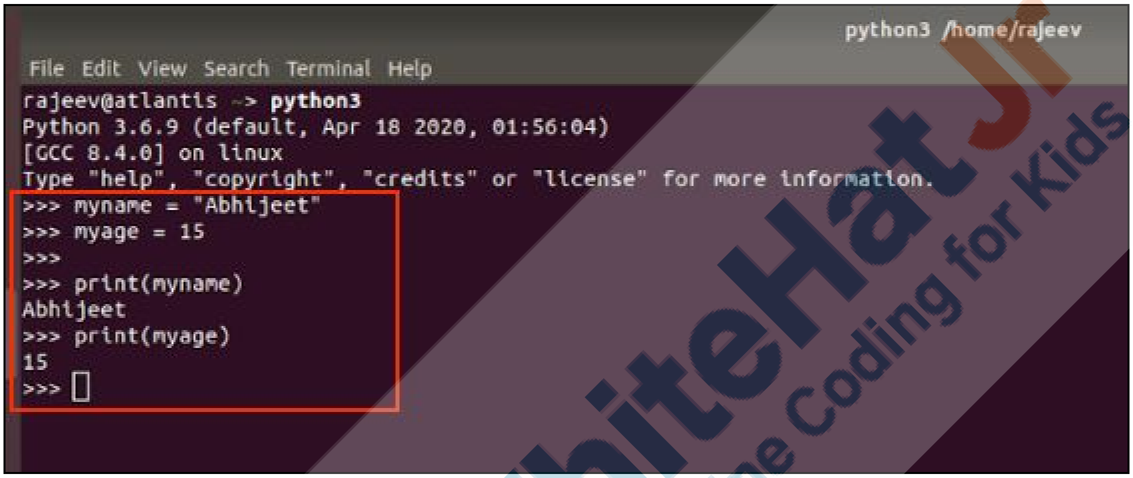


```
rajeev@atlantis -> python3
Python 3.6.9 (default, Apr 18 2020, 01:56:04)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

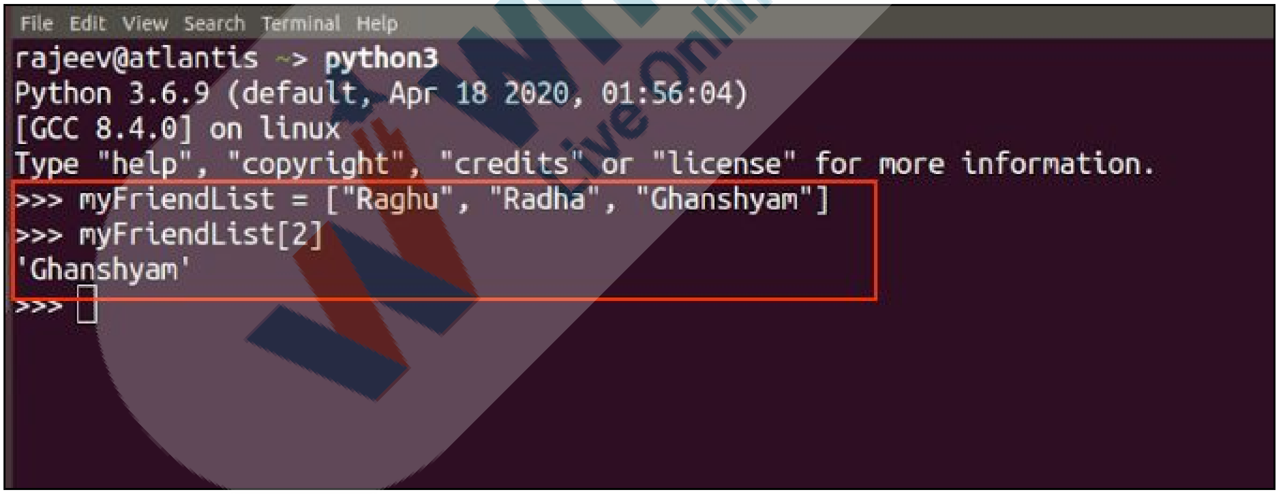
</td></tr>
<tr><td></td><td>Let's learn some python syntax in the shell.
<i>&lt;Teacher can explore the syntax from Teacher Activity 2&gt;</i>
<i>&lt;Students can refer to Student Activity 2&gt;</i></td><td>-</td></tr>
<tr><td></td><td><b>Variable</b>
You do not need to declare variables before using them, or declare their type.</td><td><i>The student learns to declare a variable, store string and numbers inside a variable.</i></td></tr>
</table>

	Let's see how to use variables in python. To define a variable, we just write the name of the variable. A variable in Python can store any kind of data - number, string, list etc. For example: Create a variable which stores student name as string. myname = "StudentName" Create a variable which stores student age as number. myage = 15 We can also print variables: print(myname) print(myage) There is another data type in python called list. It is similar to arrays of javascript. A list is created in python by placing all the items (elements) inside square brackets [], separated by commas(,) Example: myFriendList = ["Raghu", "Radha", "Ghanshyam"] A list can store different types of data. To access values in lists, use the	*The student learns to create a list and access its elements.*

	index of the list inside the square brackets. The index starts from 0 (just as in javascript). For example: myFriendList = ["Raghu", "Radha", "Ghanshyam"] myFriendList[2] The output will be Ghanshyam	



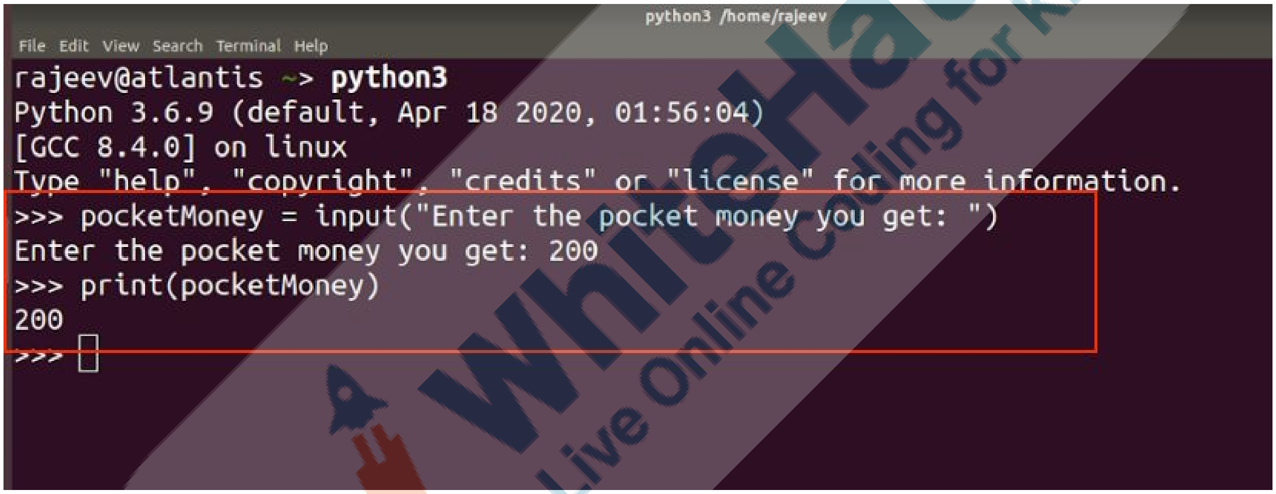
```
rajeev@atlantis ~> python3
Python 3.6.9 (default, Apr 18 2020, 01:56:04)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> myname = "Abhijeet"
>>> myage = 15
>>>
>>> print(myname)
Abhijeet
>>> print(myage)
15
>>> 
```



```
rajeev@atlantis ~> python3
Python 3.6.9 (default, Apr 18 2020, 01:56:04)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> myFriendList = ["Raghu", "Radha", "Ghanshyam"]
>>> myFriendList[2]
'Ghanshyam'
>>> 
```

	print() which we used above is a predefined function in Python. It is used to print the value passed to it as argument on the console.	*The student learns to take input and stores it in a variable.*

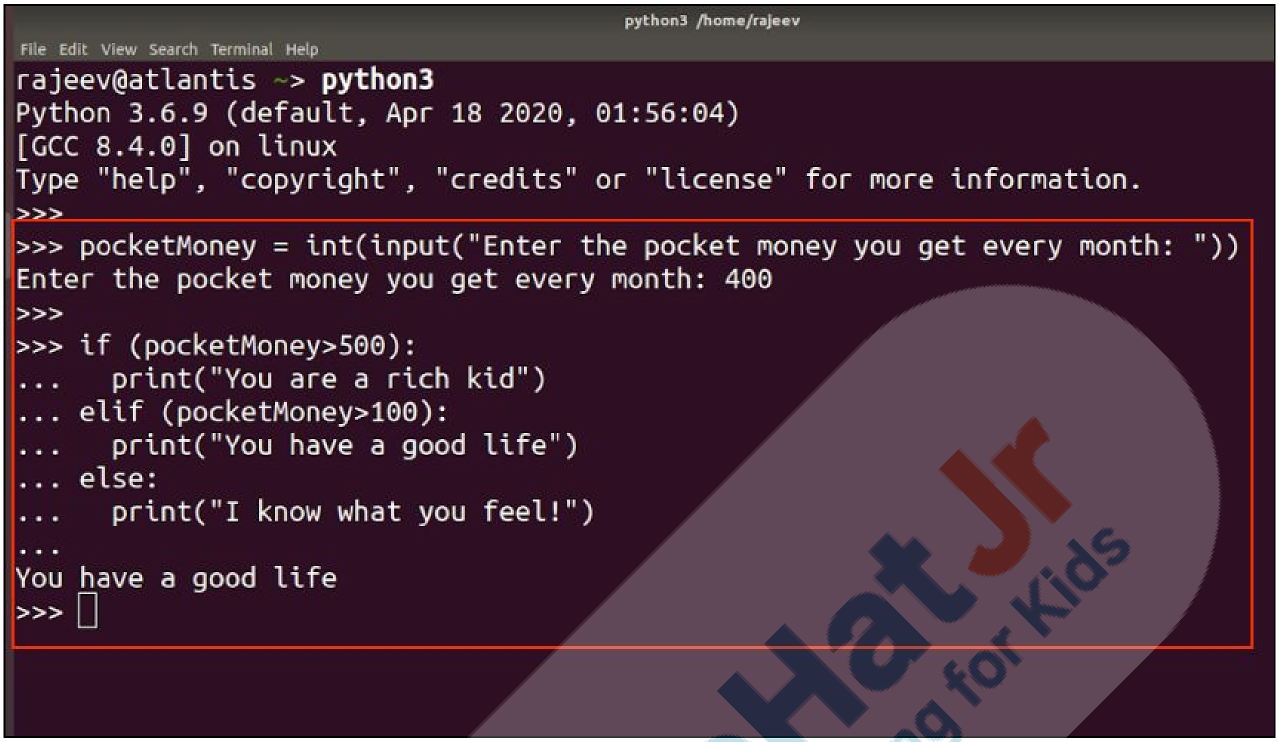
	Python has other pre-defined functions too. Function input() is used to take input from the user. **input ( )**: This function first takes the input from the user and then evaluates the expression entered. Python automatically identifies whether a user entered a string or a number or list. FOR EXAMPLE: pocketMoney = input("Enter the pocket money you get: ") print(pocketMoney)	



```
rajeev@atlantis ~> python3
Python 3.6.9 (default, Apr 18 2020, 01:56:04)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> pocketMoney = input("Enter the pocket money you get: ")
Enter the pocket money you get: 200
>>> print(pocketMoney)
200
>>> 
```

	Python has conditional programming too. It uses if-else to make decisions and execute a block of code only if a certain condition is satisfied. (similar to python) The **if…elif…else statement** is used in Python for decision making. **elif** is a combination of **else and if.**	*The student observes and learns about conditional programming in python.*

	The elif keyword is pythons way of saying "if the previous conditions were not true, then try this condition"
	Example:
	age = int(input("Enteryour age: "))
	if (age>18):
	  print("You are an adult. You can vote!")
	elif (age>12):
	  print("You are a teenager and a rebel!")
	else:
	  print("You are still a kid. There is so much in the world for you to see.")
	
	Note that python doesn't use { } to write the block of code which needs to be executed when a certain condition is true. Instead it uses indentation to describe those blocks of code. To indent a code, you can use Tab Key (like before!) Tab key indents by 4 spaces.
	Indentation, as we will see in python, is very important.	



```
rajeev@atlantis ~> python3
Python 3.6.9 (default, Apr 18 2020, 01:56:04)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> pocketMoney = int(input("Enter the pocket money you get every month: "))
Enter the pocket money you get every month: 400
>>>
>>> if (pocketMoney>500):
...   print("You are a rich kid")
... elif (pocketMoney>100):
...   print("You have a good life")
... else:
...   print("I know what you feel!")
...
You have a good life
>>>
```

Let's learn about loops in Python.

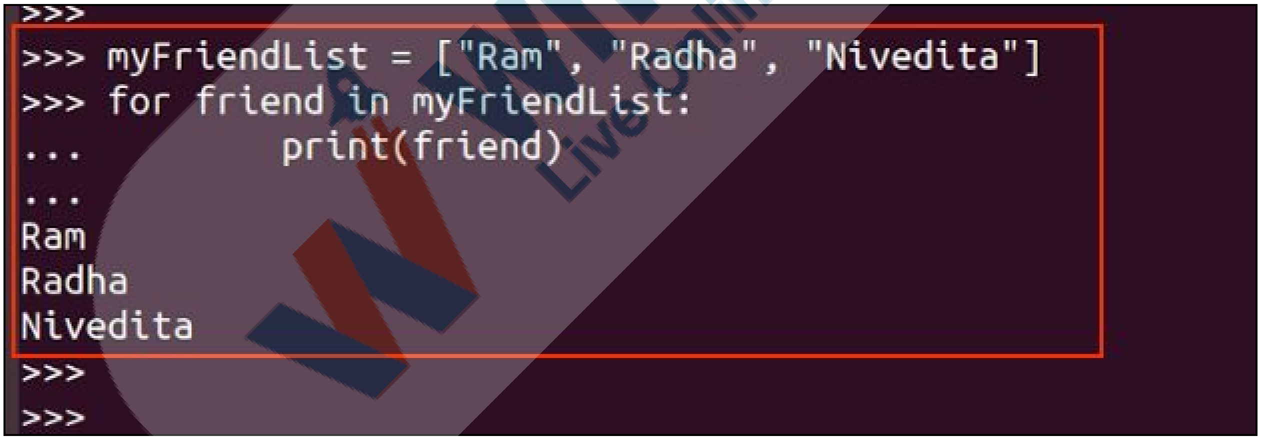
(There are two types of loops, For loop and While loop)
-The for loop in Python is used to iterate over a sequence (list or string) or other iterable objects.

Let's see with an example:

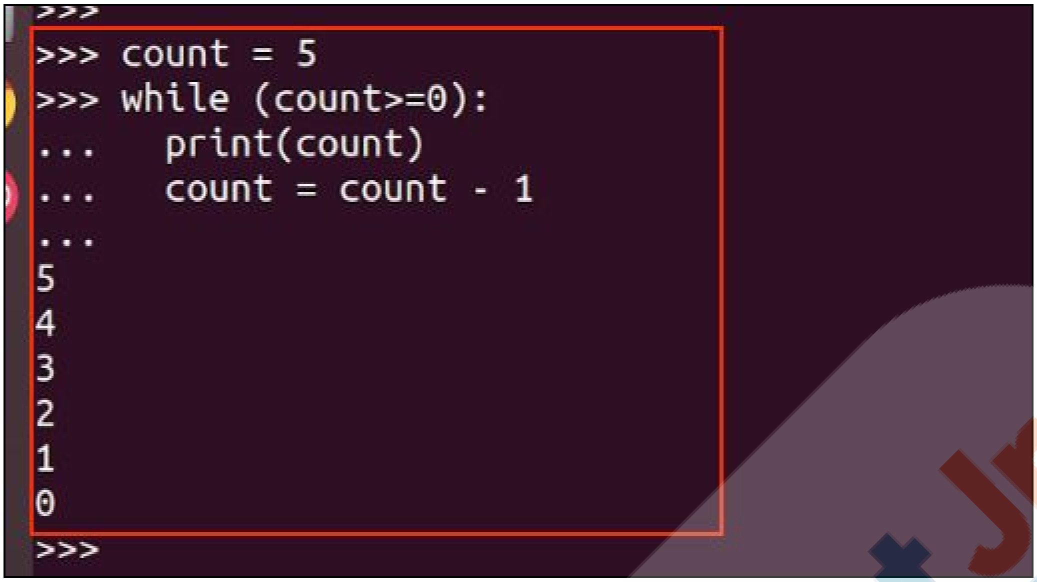
```
myFriendList = ["Ram", "Radha", "Nivedita"]
for friend in myFriendList:
        print(friend)
```

-The while loop in Python is used to iterate over a block of code as long as the test expression (condition) is true.

	We generally use this loop when we don't know the number of times to iterate beforehand.
	
	The format of while code is
	while (condition is true):
	   keep doing stuff inside this block
	
	The code inside while block keeps getting executed till the condition inside while is true. As soon as the condition becomes false, the next line of code after the while block is executed.
	
	Let's see with an example:
	count = 5
	while (count>=0):
	  print(count)
	  count = count - 1	



```
>>>
>>> myFriendList = ["Ram", "Radha", "Nivedita"]
>>> for friend in myFriendList:
...         print(friend)
...
Ram
Radha
Nivedita
>>>
>>>
```

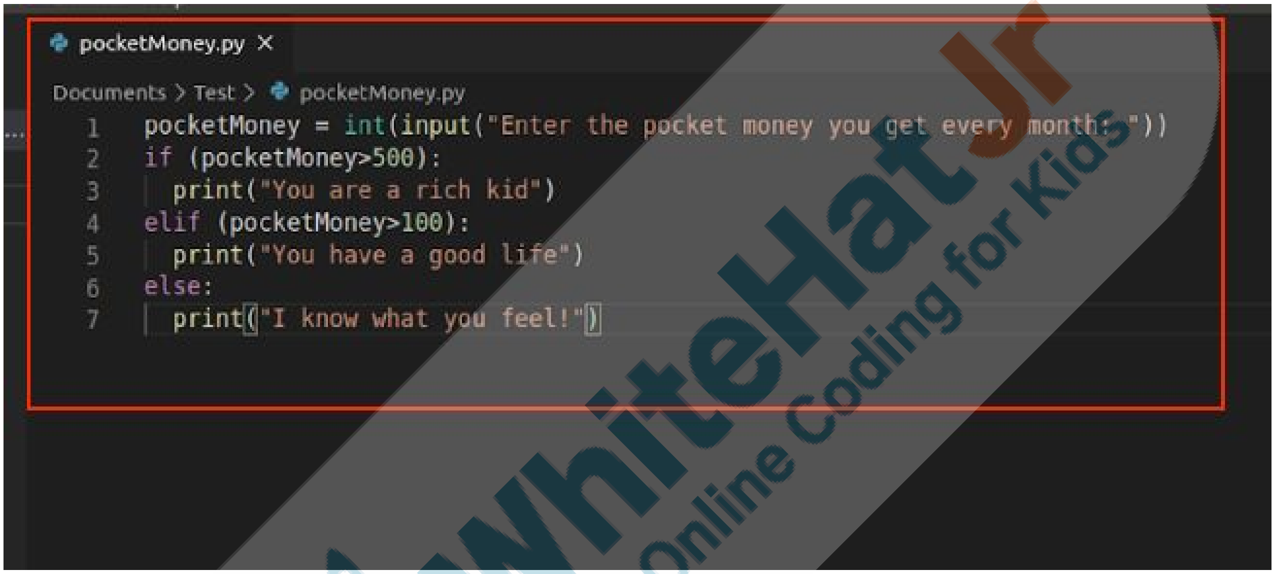


```
>>>
>>> count = 5
>>> while (count>=0):
...     print(count)
...     count = count - 1
...
5
4
3
2
1
0
>>>
```

	Great! we are introduced to all the simple python constructs and we are ready to write python programs!
	We have been writing python programs in python shell. We will continue to use python shell to test small code snippets we write.
	
	But we can also write our code in visual code studio and run them! Let's see how.
	
	- Teacher opens up visual code studio
	- Creates a new file
	- writes a simple python program below:
	pocketMoney = int(input("Enter the pocket money you get every month: "))
	if (pocketMoney>500):
	  print("You are a rich kid")
	elif (pocketMoney>100):
	  print("You have a good life")
	else:
		Student sees the output of the code on the terminal.

<table>
<tr>
<td></td>
<td>print("I know what you feel!")

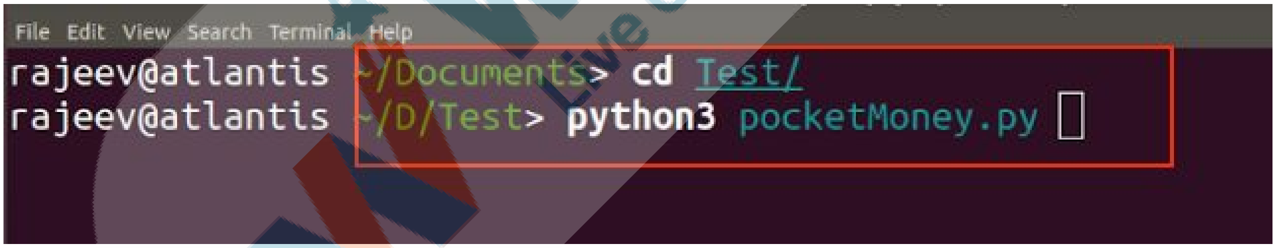
- Save the file as pocketMoney.py
- Open up the terminal
- Run the command
 <path where the file is> python3 pocketMoney.py</td>
<td></td>
</tr>
<tr>
<td colspan="3">



pocketMoney.py ×

Documents > Test > pocketMoney.py

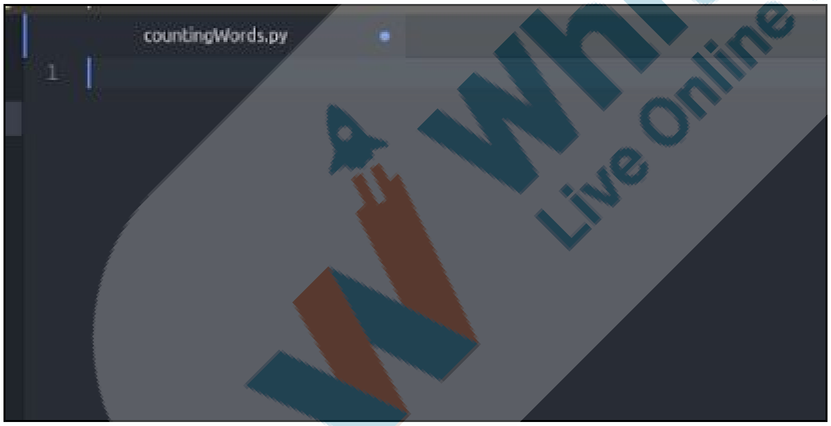
```python
pocketMoney = int(input("Enter the pocket money you get every month: "))
if (pocketMoney>500):
  print("You are a rich kid")
elif (pocketMoney>100):
  print("You have a good life")
else:
  print("I know what you feel!")
```



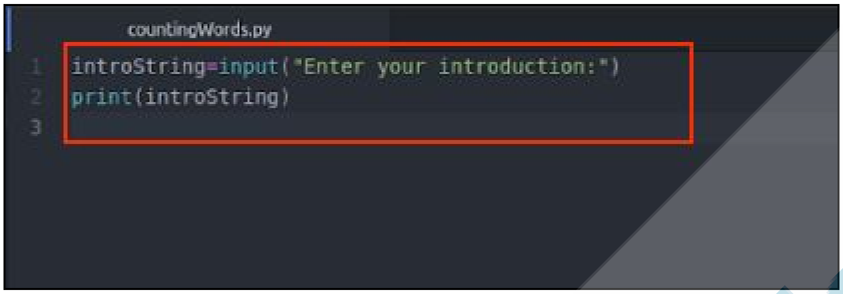
```
rajeev@atlantis ~/Documents> cd Test/
rajeev@atlantis ~/D/Test> python3 pocketMoney.py
```

</td>
</tr>
<tr>
<td></td>
<td>Ok. Now here's a challenge for you. Can you write a simple program which asks the user to introduce themselves? The program should then count and display the number of words used by the user to give their introduction.

Do you think you can do that?</td>
<td><b>ESR:</b>
Yes!</td>
</tr>
</table>

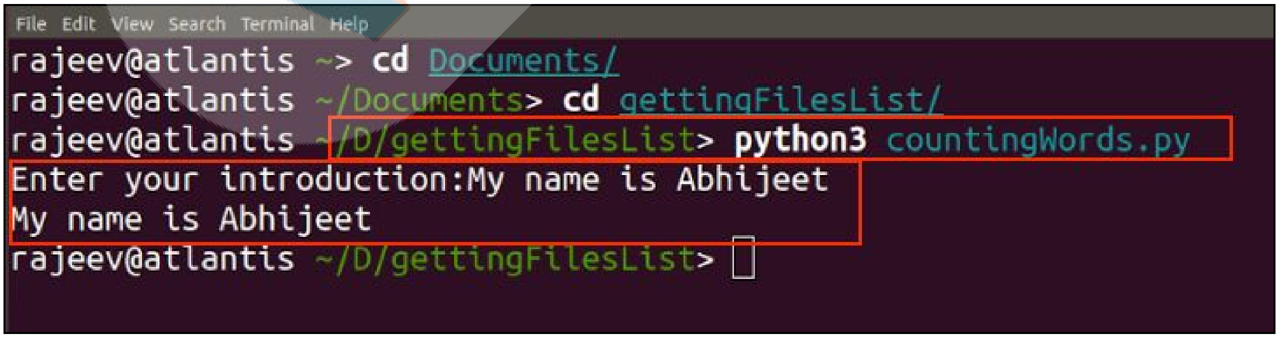
	Alright...let's do it then. I can guide you wherever you need help.	
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	
● **Ask Student to press ESC key to come back to panel** ● **Guide Student to start Screen Share** ● **Teacher gets into Fullscreen**		
Step 3: **Student-Led Activity** **(5 min)**	Let's open our editor and create a new file and name it as countWords.py. We are using .py as the file extension to tell our system that it's a python file. *<Teacher helps student to open the editor and create a new file countWords.py >*	*<Student opens the editor and creates a new file countWords.py>*
 countingWords.py		
	So first let's start by taking input and store it in a variable called string. introString=input("Enter your introduction:") *<Teacher helps student take an input and store it in a variable >*	*<Student writes code to take input and store it in a variable>*

	Very good now let's print the string that we stored in the variable. To do so in python we use a statement called print. So we'll write print(intorString) and save the code. *<Teacher helps student write print statement in the code>*	*<Student adds the print statement to the code>*



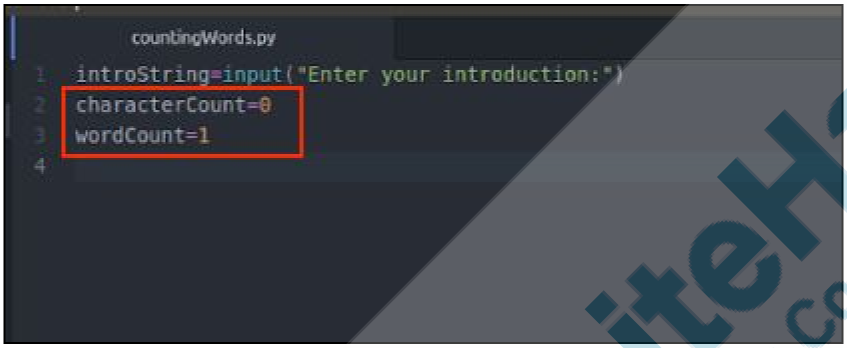
```
introString=input("Enter your introduction:")
print(introString)
```

	Now let's check for the output. To do so we will use a terminal. - Open the terminal - Using cd command go to the folder which contains your countWords.py file - To run the file we use python3 filename command. Here we'll write python3 countWords.py We'll see output on the terminal. *<Teacher helps student to run the file to check for output.>*	*<Student runs the file and see's the output>*



```
rajeev@atlantis ~> cd Documents/
rajeev@atlantis ~/Documents> cd gettingFilesList/
rajeev@atlantis ~/D/gettingFilesList> python3 countingWords.py
Enter your introduction:My name is Abhijeet
My name is Abhijeet
rajeev@atlantis ~/D/gettingFilesList>
```

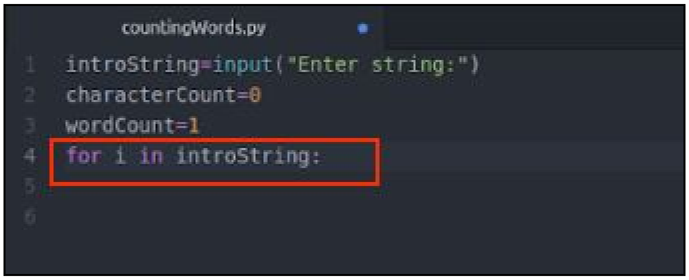
	Now, let's declare two variables wordCount and characterCount in our program. The 'characterCount' variable is initialized to zero and the wordCount variable is initialized to 1. characterCount=0 wordCount=1 *<Teacher helps student to declare the two variables and set their values to 0 and 1>*	*<Student codes to declare two variables and set their values to 0 >*



countingWords.py

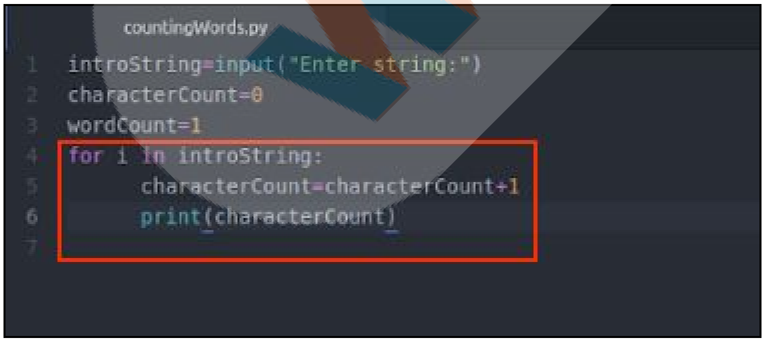
```
introString=input("Enter your introduction:")
characterCount=0
wordCount=1
```

	We will use a **for loop** to iterate on the string. To do that we'll write **for character in introString:** here "character" is the loop variable which will be assigned list elements from introString one after the other (including space). The for loop is used to traverse through the characters in the string. *<Teacher helps the student to write the for loop statement.>*	*<Student writes the* ***for loop*** *statement>*



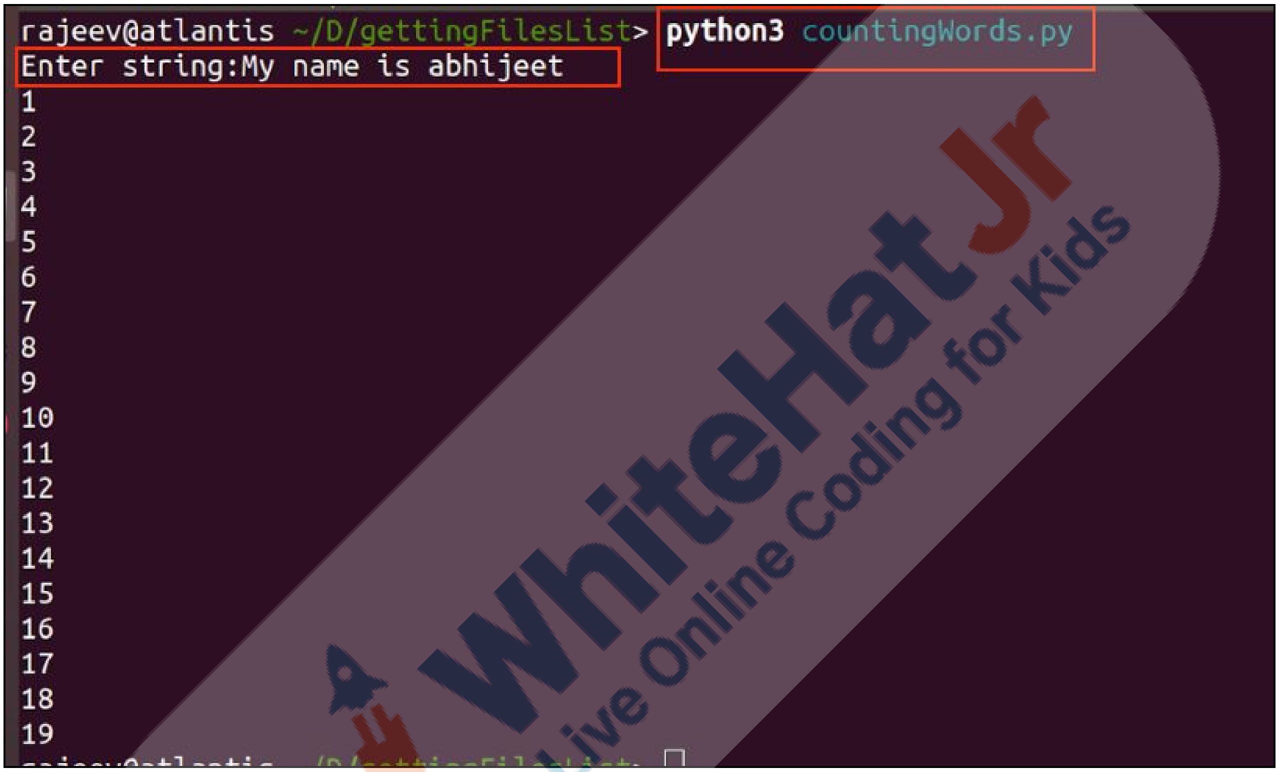
```
introString=input("Enter string:")
characterCount=0
wordCount=1
for i in introString:
```

	Inside the for loop to count the character we'll write :- (To write the statement or code inside a function we will add an indentation at the beginning) Code- **for character in introString:** **characterCount=characterCount + 1** **print(characterCount)** (The character count is incremented each time a character is encountered). *<Teacher helps the student with the code to count characters>*	*<Student writes the code to count the characters>*



```
introString=input("Enter string:")
characterCount=0
wordCount=1
for i in introString:
        characterCount=characterCount+1
        print(characterCount)
```

	Let's run the code to see our output.	*Student runs the code to check the output.*

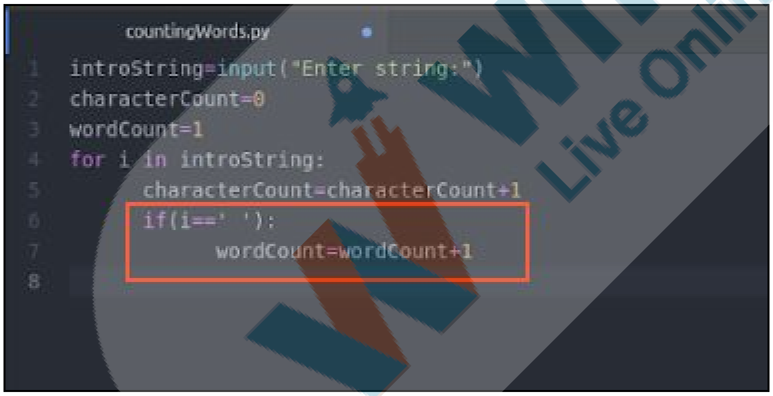


```
rajeev@atlantis ~/D/gettingFilesList> python3 countingWords.py
Enter string:My name is abhijeet
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
rajeev@atlantis ~/D/gettingFilesList> □
```

	Can you tell me how to check for words in our program.	**ESR:** We can check for the blank space in between the two words that way we can know if a word is completed.

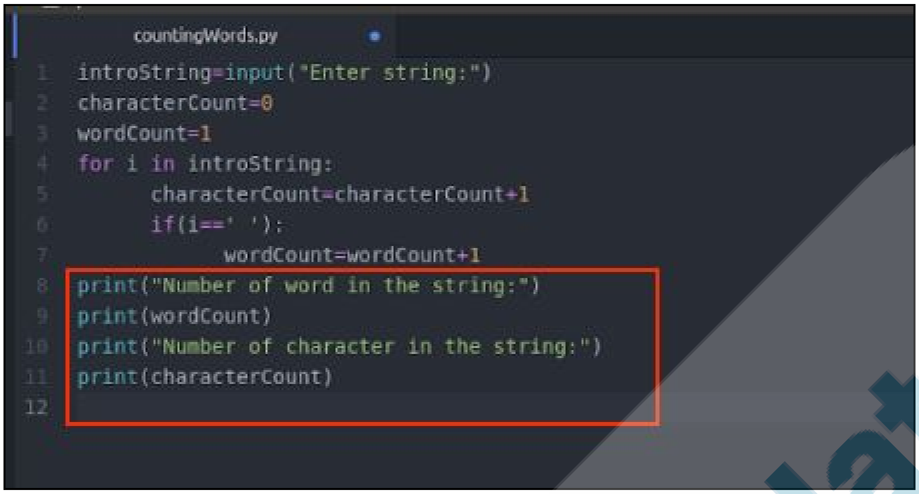
<table>
<tr>
<td></td>
<td>Very good. We'll check for the blank space in between the words.
While iterating on the introString "character" also takes all the blank spaces it contains.
So whenever "character" encounters a blank space we'll mark it as a word and increase the word count.
The code for this would be:
<b>if(character == ' '):</b>
        <b>wordCount=wordCount+1</b>

Note the user does not press space after the last word. This needs to be kept in account when printing the final wordCount, the final wordCount has to be increased by 1.

<i>&lt;Teacher helps the student with the code for counting the word&gt;</i></td>
<td><i>&lt;Student writes the code for counting the word&gt;</i></td>
</tr>
<tr>
<td colspan="3">
 <pre>
introString=input("Enter string:")
characterCount=0
wordCount=1
for i in introString:
    characterCount=characterCount+1
    if(i==' '):
        wordCount=wordCount+1
</pre></td>
</tr>
<tr>
<td></td>
<td>Now all that is left is to print the word count and the characters count.
We'll use the print statement to print the word count and character count.

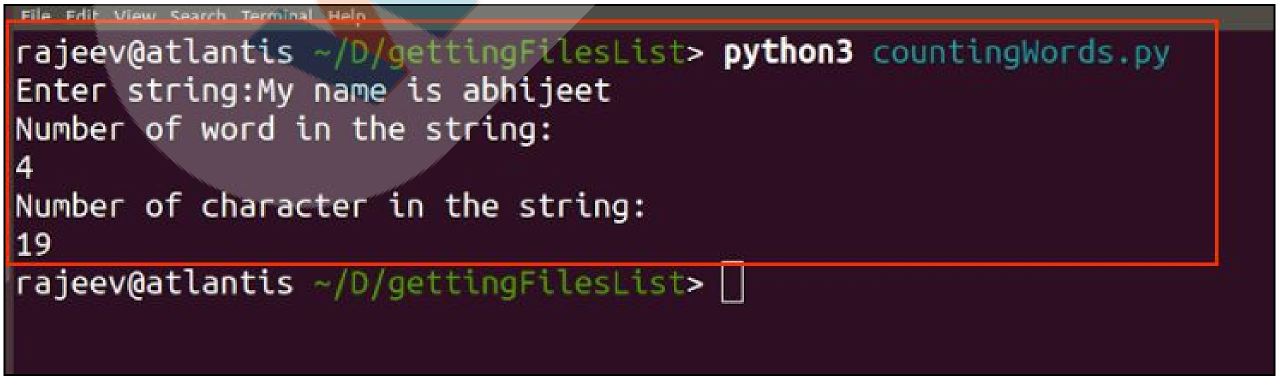
<b>print("Number of words in the string:")</b>
<b>print(wordCount + 1)</b></td>
<td><i>&lt;Student writes the code to print the number of words and characters in the string&gt;</i></td>
</tr>
</table>

print("Number of characters in the string:")
print(characterCount)



```
introString=input("Enter string:")
characterCount=0
wordCount=1
for i in introString:
        characterCount=characterCount+1
        if(i==' '):
                wordCount=wordCount+1
print("Number of word in the string:")
print(wordCount)
print("Number of character in the string:")
print(characterCount)
```

	Now let's run and test our code. To run the code we'll do the same thing again that we did before . - We'll open our terminal - Then we'll go to the folder which has our countWords.py file. - Write python3 filename command and press enter. *<Teacher helps the student to run the file and test the code>*	*<Student runs the file and sees the output>*



```
rajeev@atlantis ~/D/gettingFilesList> python3 countingWords.py
Enter string:My name is abhijeet
Number of word in the string:
4
Number of character in the string:
19
rajeev@atlantis ~/D/gettingFilesList>
```

Teacher Guides Student to Stop Screen Share

Step 4: **Wrap-Up**	Amazing. Next class, we will learn how to read input from different files in python. We can manipulate files (add, edit, delete, etc.) using a python program. We can also move files from one location to another! I hope you are waiting for the next class.	-
Project Overview	**Introduction to python** **Goal of the Project:** Today you have learned the basic syntax of python language. **Story:** Due to the pandemic Rahul is stuck at home and is very bored. Create a guessing game of numbers using the knowledge of python that you have now. I am very excited to see your project solution and I know you will do really well. Bye Bye!	

Teacher Clicks		
Additional Activities	*Encourage the student to write reflection notes in their reflection journal using markdown.* Use these as guiding questions: ● What happened today? - Describe what happened - Code I wrote ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? What did I find difficult?	*The student uses the markdown editor to write her/his reflection in a reflection journal.*

Activity	Activity Name	Links
Student Activity 1	Python installation	https://www.python.org/downloads/
Student Activity 2	Basic Python Syntax	https://www.tutorialspoint.com/python/python_basic_syntax.htm
Student Activity 3	Installation of Python for Windows, Mac and Linux	https://realpython.com/installing-python/
Teacher Activity 1	Python installation	https://www.python.org/downloads/
Teacher Activity 2	Basic Python Syntax	https://www.tutorialspoint.com/python/python_basic_syntax.htm
Teacher Activity 3	solution	https://github.com/whitehatjr/countingWords

Teacher Activity 4	Installation of python for windows, mac and linux	https://realpython.com/installing-python/