python™

Programming Course

Aegis

SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Day - One

Getting Started with Python

Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Getting Started with Python

Day - 1 Agenda

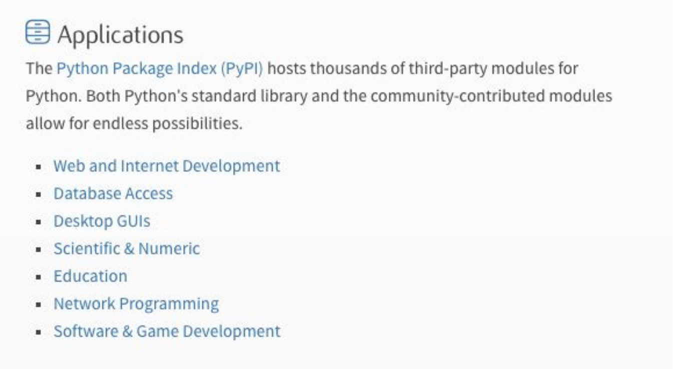
- Programming Fundamentals
- Thinking Programming
- Introduction to Python
- Why Python?
- Installation
- Developers Infrastructure
  - IDE vs Command Line vs Anaconda
- Hello Python
- Comments
- Inbuilt Functions
  - print, input, int, float
- Mathematics
  - Various Operations
- Variables
- Making Decisions - if, else, elif
- and, or operation
- Lab Assignment

Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Introduction to Python

First Appeared on 20 Feb 1991 - 26 years ago

- General Purpose, Open Source Programming Language
- Easy to Learn Programming Language
- Large Community - Many Free Libraries and Modules available
- Data Science, AI
  - With modules like numpy,scipy,Pandas Python dominates the data science and machine learning space and became one of the preferred language for computer science research

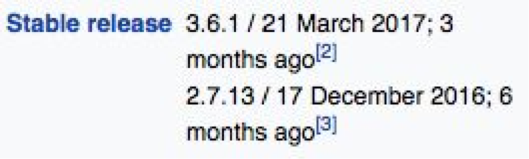


Applications

The Python Package Index (PyPI) hosts thousands of third-party modules for Python. Both Python's standard library and the community-contributed modules allow for endless possibilities.

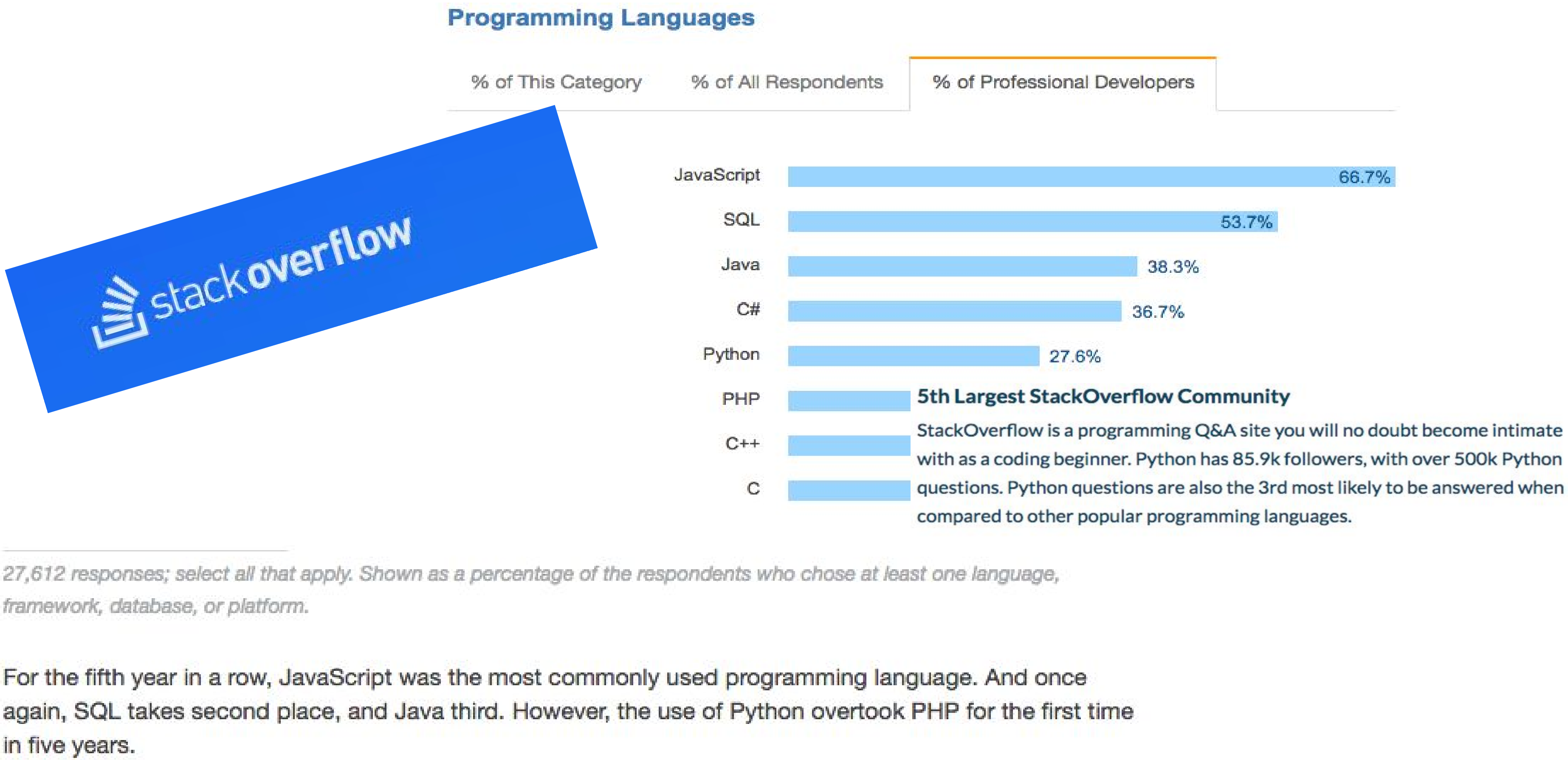
- Web and Internet Development
- Database Access
- Desktop GUIs
- Scientific & Numeric
- Education
- Network Programming
- Software & Game Development

Python 2.x vs Python 3.x



Stable release	3.6.1 / 21 March 2017; 3 months ago[2] 2.7.13 / 17 December 2016; 6 months ago[3]

Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION



Programming Languages

% of This Category | % of All Respondents | % of Professional Developers

5th Largest StackOverflow Community

StackOverflow is a programming Q&A site you will no doubt become intimate with as a coding beginner. Python has 85.9k followers, with over 500k Python questions. Python questions are also the 3rd most likely to be answered when compared to other popular programming languages.

27,612 responses; select all that apply. Shown as a percentage of the respondents who chose at least one language, framework, database, or platform.

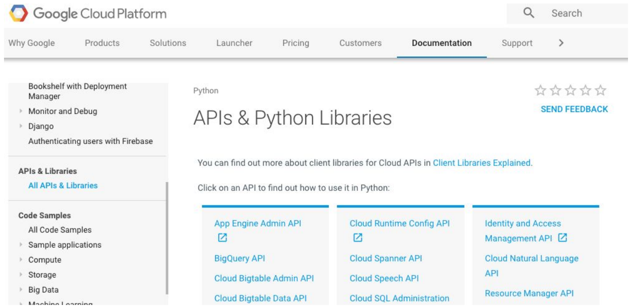
For the fifth year in a row, JavaScript was the most commonly used programming language. And once again, SQL takes second place, and Java third. However, the use of Python overtook PHP for the first time in five years.

Source: https://insights.stackoverflow.com/survey/2017

Why Python? (Cont …)

Almost Every Big Organization use Python:

- Google - Google Cloud has Python API Libraries
- Facebook
- Twitter
- AWS (Amazon)



Bookshelf with Deployment Manager
Monitor and Debug
Django
Authenticating users with Firebase

APIs & Libraries
All APIs & Libraries

Code Samples
All Code Samples
Sample applications
Compute
Storage
Big Data
Machine Learning

Python

APIs & Python Libraries

SEND FEEDBACK

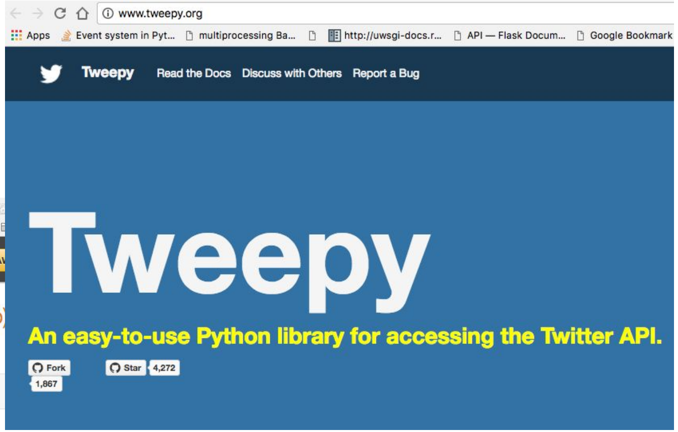
You can find out more about client libraries for Cloud APIs in Client Libraries Explained.

Click on an API to find out how to use it in Python:

App Engine Admin API
BigQuery API
Cloud Bigtable Admin API
Cloud Bigtable Data API

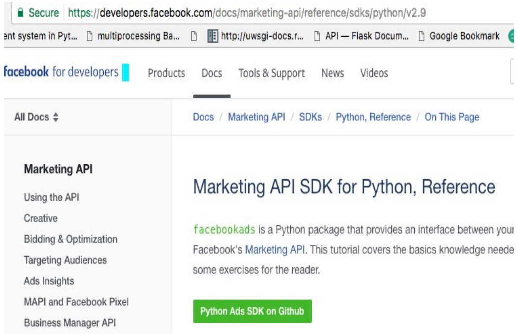
Cloud Runtime Config API
Cloud Spanner API
Cloud Speech API
Cloud SQL Administration

Identity and Access Management API
Cloud Natural Language API
Resource Manager API



Tweepy

An easy-to-use Python library for accessing the Twitter API.

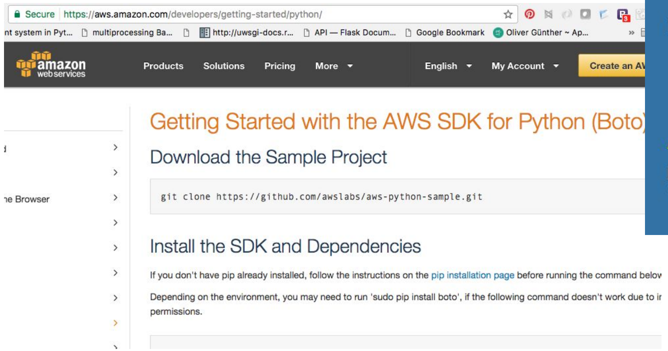


Marketing API
Using the API
Creative
Bidding & Optimization
Targeting Audiences
Ads Insights
MAPI and Facebook Pixel
Business Manager API

Marketing API SDK for Python, Reference

`facebookads` is a Python package that provides an interface between your Facebook's Marketing API. This tutorial covers the basics knowledge neede some exercises for the reader.

Python Ads SDK on Github



Getting Started with the AWS SDK for Python (Boto)

Download the Sample Project

```
git clone https://github.com/awslabs/aws-python-sample.git
```

Install the SDK and Dependencies

If you don't have pip already installed, follow the instructions on the pip installation page before running the command belov

Depending on the environment, you may need to run 'sudo pip install boto', if the following command doesn't work due to ir permissions.

Why Python? (Cont …)

- Used in almost every Domain
  - GUI Applications
  - Mobile Applications
  - Web Applications - based on many frameworks from Python
    - Django, Flask, Falcon etc

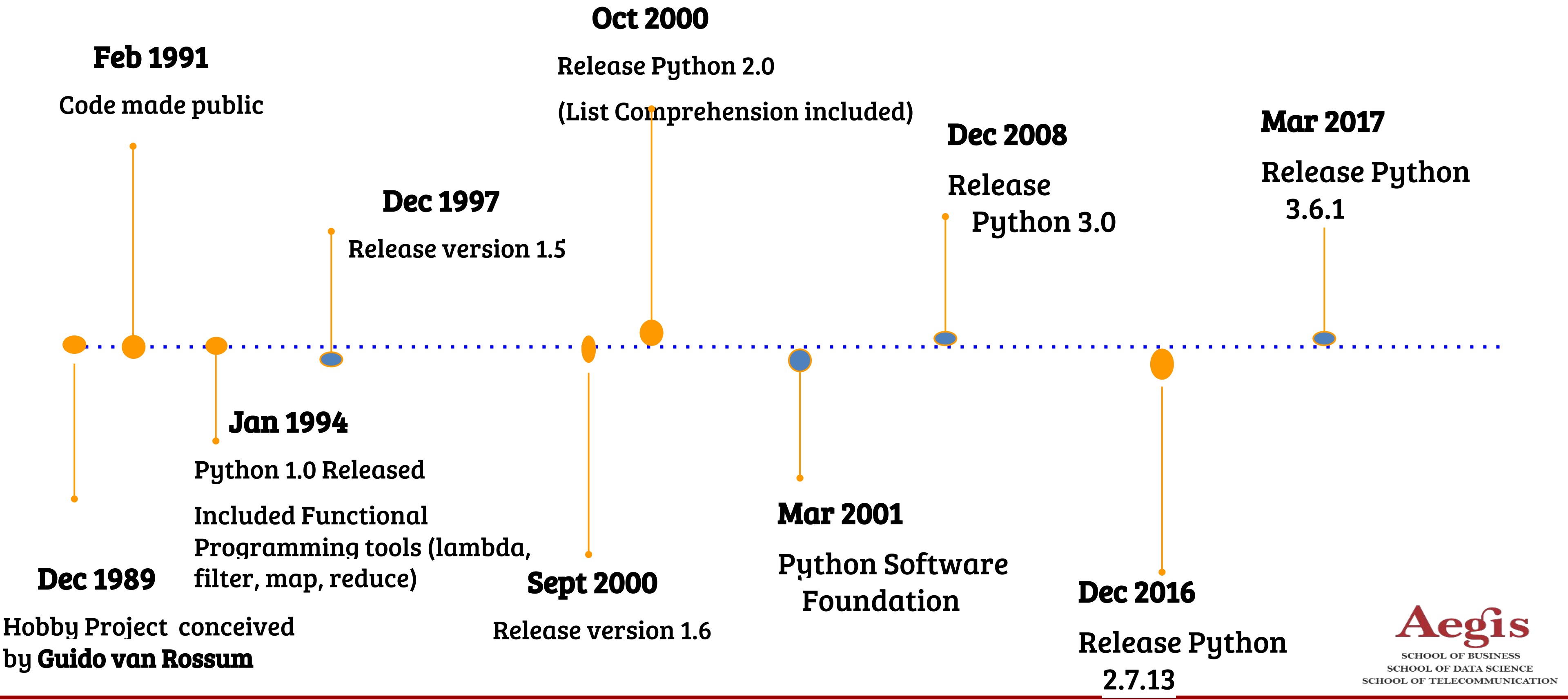
Why Python? (Cont …)

Take a look at

- https://www.splunk.com/view/SP-CAAABF9
- https://www.splunk.com/en_us/solutions/solution-areas/log-management.html

Python Programming History

Python Programming Foundation will provide support to version 2.7 till 2020



Dec 1989

Hobby Project conceived by **Guido van Rossum**

Feb 1991

Code made public

Jan 1994

Python 1.0 Released

Included Functional Programming tools (lambda, filter, map, reduce)

Dec 1997

Release version 1.5

Sept 2000

Release version 1.6

Oct 2000

Release Python 2.0

(List Comprehension included)

Mar 2001

Python Software Foundation

Dec 2008

Release Python 3.0

Dec 2016

Release Python 2.7.13

Mar 2017

Release Python 3.6.1

Aegis

SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Python 2 vs Python 3

Python 2.x is legacy, Python 3.x is the present and future of the language

- Let us stick to Python 3
  - Latest Version, Matured Releases
  - Current Development focus for “Python Foundation”

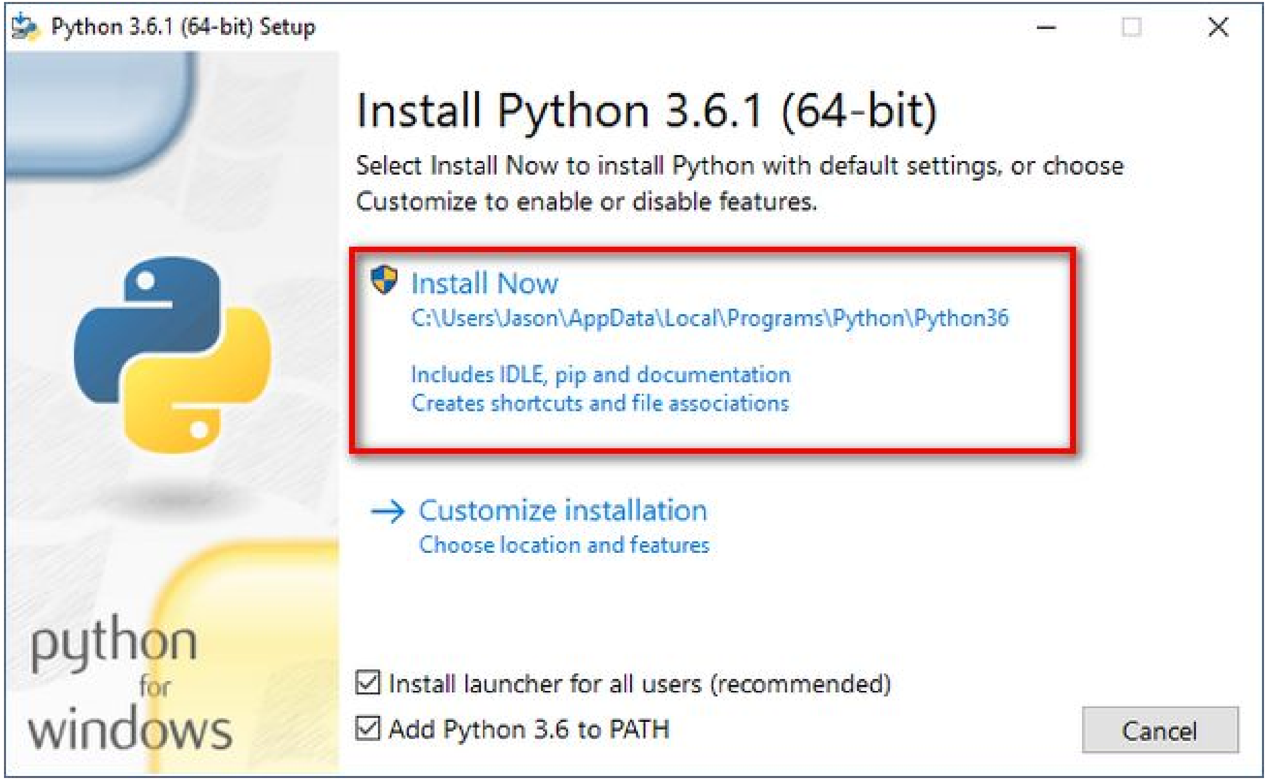
Python - 2
-No Major Releases, Only Patch Releases

- Python 2 and Python 3 are almost different Programming Languages
  - Syntax is reasonably similar, developed same set of developers, same set of communities

-print is method in Python 3
-By default text/strings are utf-8 encoded/decode support in Python 3
-Python 2 has long, plain integer. Python 3 does not differentiate

- Python 3.0 released: Year2008
- The final 2.7 release: Year 2010

Installation - Start Programming



Python 3.6.1 (64-bit) Setup

Install Python 3.6.1 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Jason\AppData\Local\Programs\Python\Python36

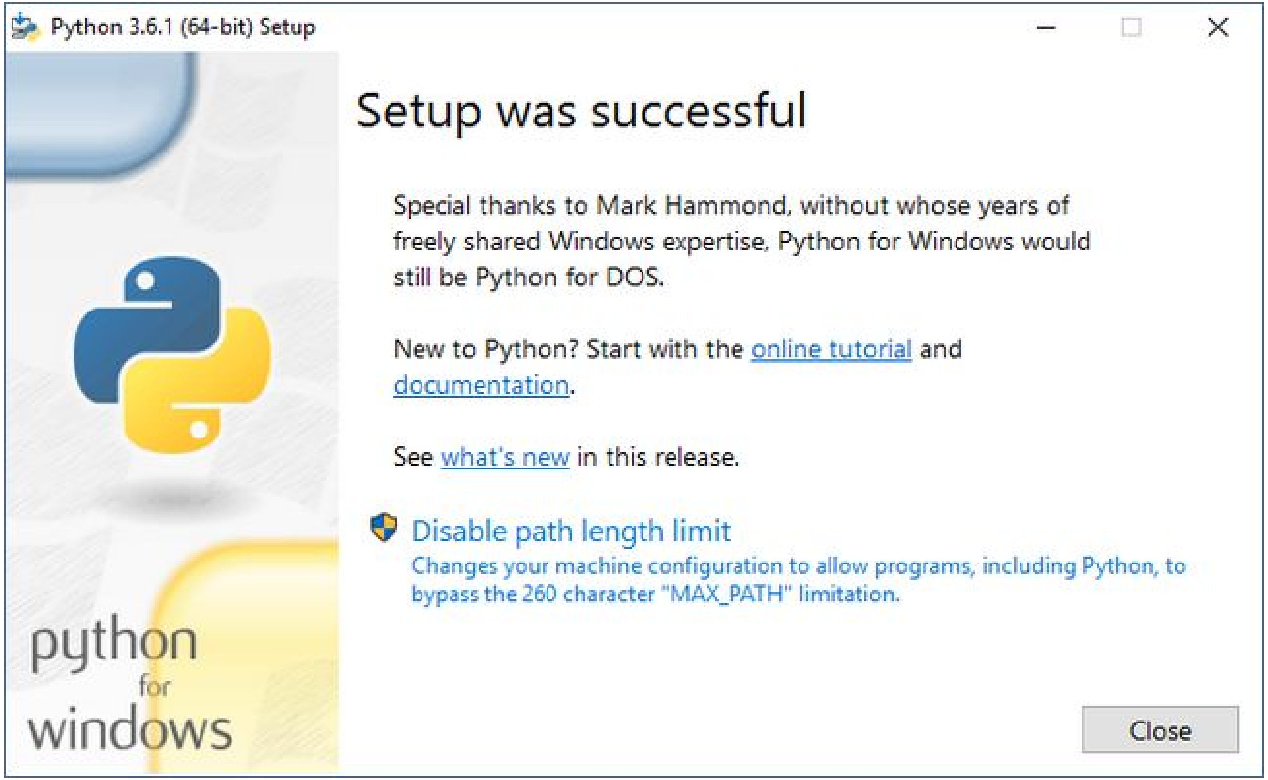
Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)

☑ Add Python 3.6 to PATH

Installation- Start Programming (Cont ...)



Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

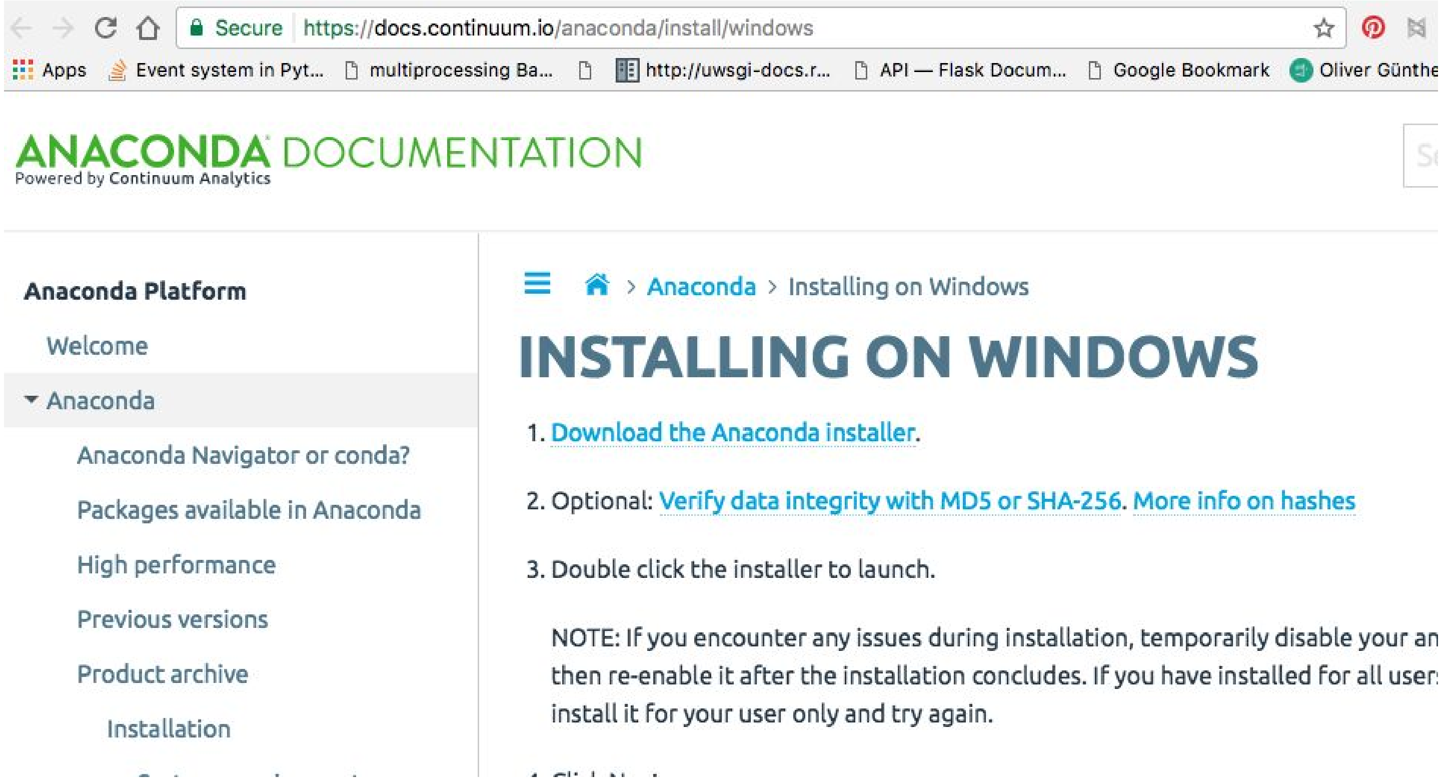
New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Disable path length limit

Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Installation - Anaconda



Secure https://docs.continuum.io/anaconda/install/windows

Apps Event system in Pyt... multiprocessing Ba... http://uwsgi-docs.r... API — Flask Docum... Google Bookmark Oliver Günthe

ANACONDA DOCUMENTATION
Powered by Continuum Analytics

Anaconda Platform

Welcome

Anaconda

Anaconda Navigator or conda?

Packages available in Anaconda

High performance

Previous versions

Product archive

Installation

Anaconda > Installing on Windows

INSTALLING ON WINDOWS

1. Download the Anaconda installer.

2. Optional: Verify data integrity with MD5 or SHA-256. More info on hashes

3. Double click the installer to launch.

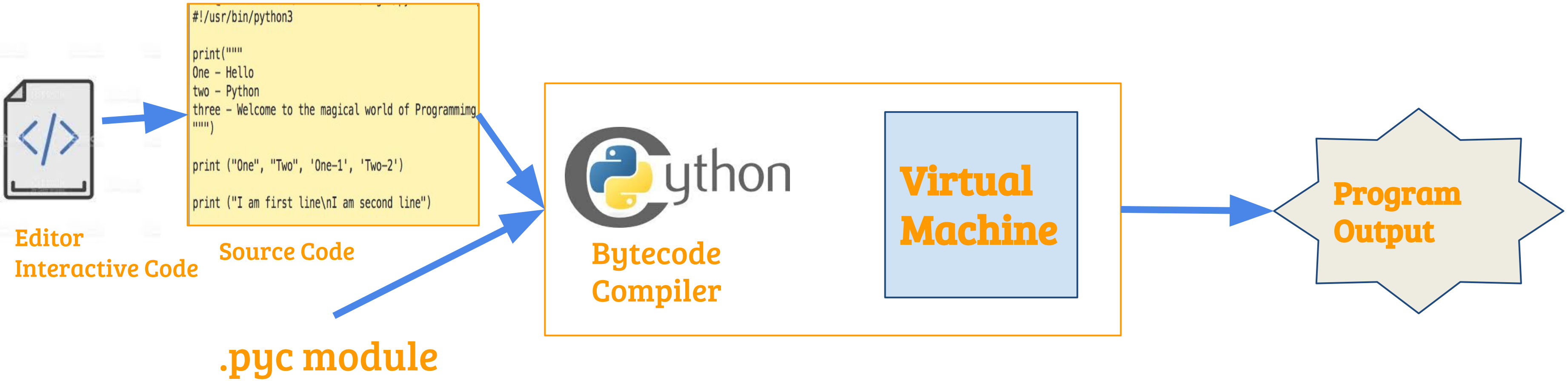
   NOTE: If you encounter any issues during installation, temporarily disable your an then re-enable it after the installation concludes. If you have installed for all user install it for your user only and try again.

Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Executing Python

- Python Script
  - Use your favourite Editor, IDE to develop scripts
  - Use Python Interpreter to execute Code
- Python Interactive Shell
  - Helpful to explore Python Syntax
  - Get interactive help on commands
  - Debug short programs

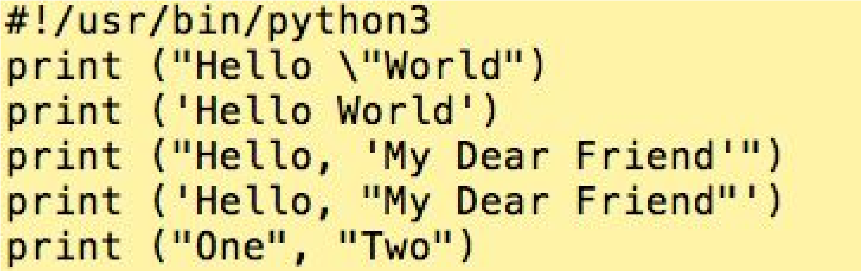
How Python Works?



```
#!/usr/bin/python3

print("""
One - Hello
two - Python
three - Welcome to the magical world of Programmimg
""")

print ("One", "Two", 'One-1', 'Two-2')

print ("I am first line\nI am second line")
```

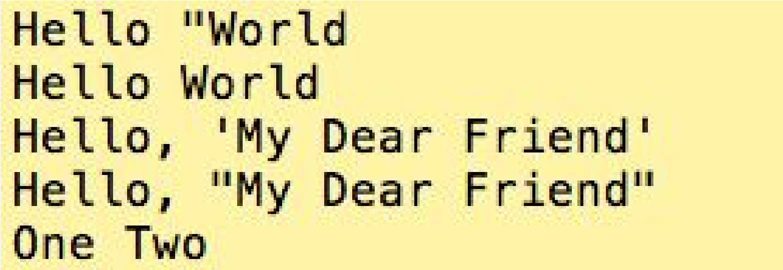
Hello to Python

hello.py



```
#!/usr/bin/python3
print ("Hello \"World")
print ('Hello World')
print ("Hello, 'My Dear Friend'")
print ('Hello, "My Dear Friend"')
print ("One", "Two")
```

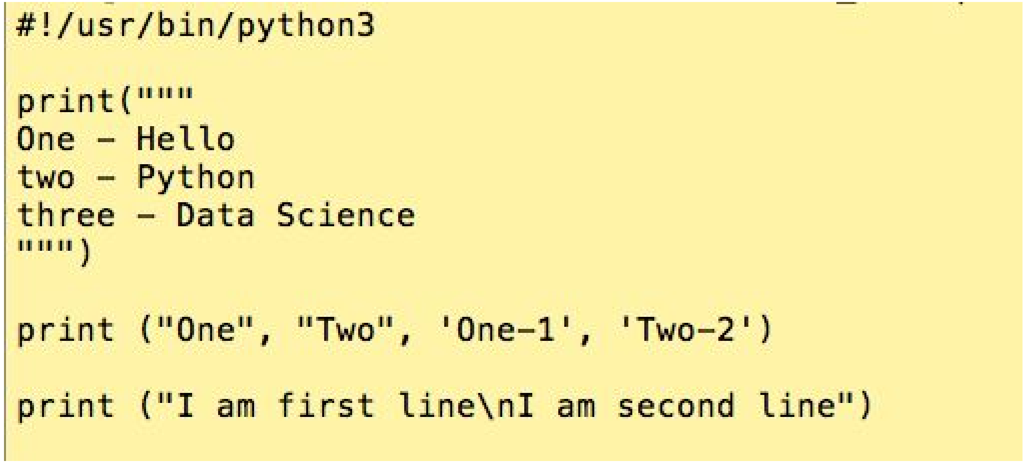
output



```
Hello "World
Hello World
Hello, 'My Dear Friend'
Hello, "My Dear Friend"
One Two
```

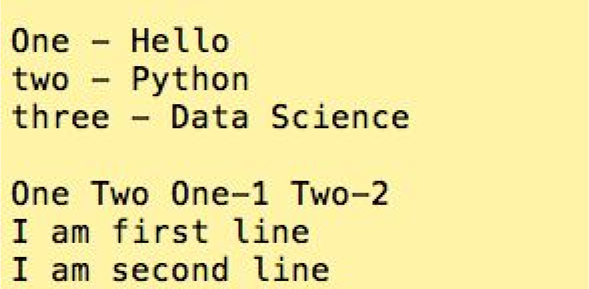
- print is built-in function
- Top to bottom approach for program execution
- Strings - escape, double, single quotes

Hello to Python (Cont…)



```
#!/usr/bin/python3

print("""
One - Hello
two - Python
three - Data Science
""")

print ("One", "Two", 'One-1', 'Two-2')

print ("I am first line\nI am second line")
```

output

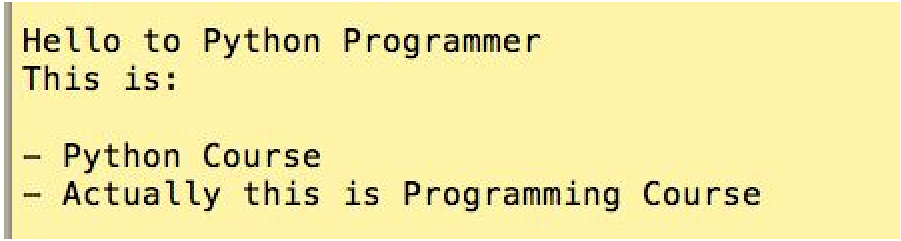


```
One - Hello
two - Python
three - Data Science

One Two One-1 Two-2
I am first line
I am second line
```

hello_multiple.py

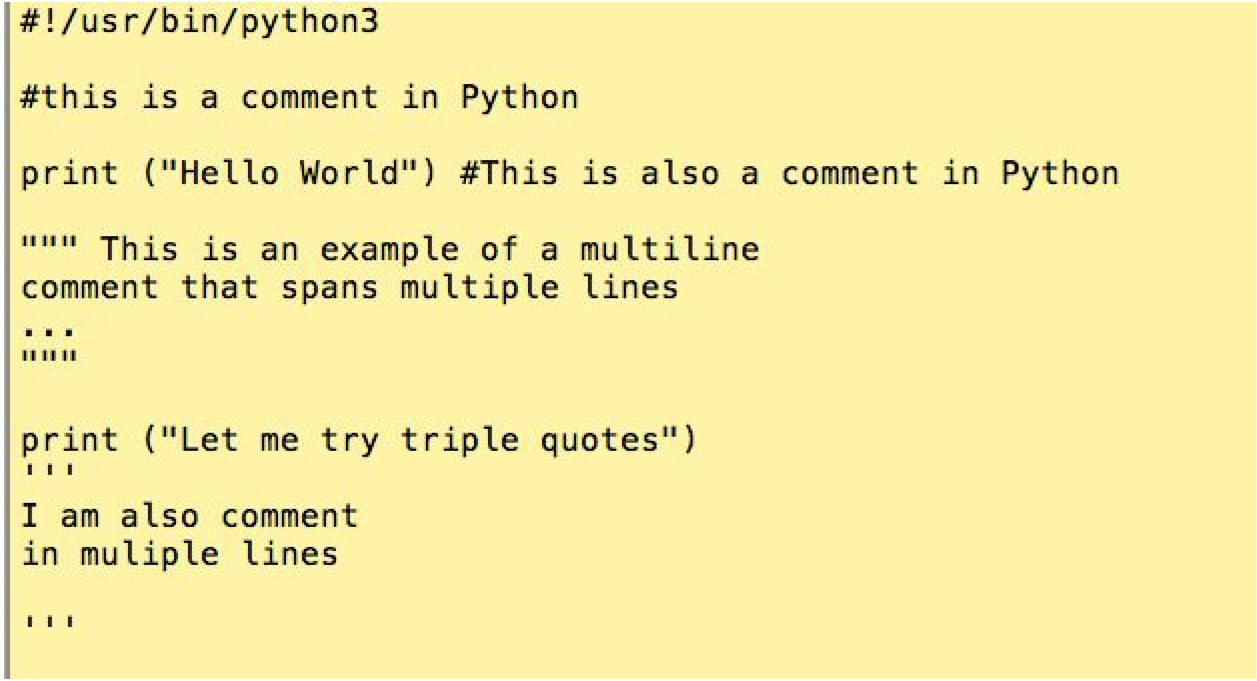
Task - 1

- Write “hello_python_programmer.py” Script
- It should print following output



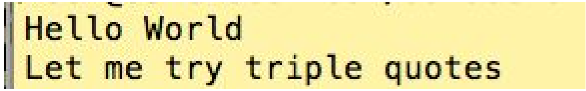
```
Hello to Python Programmer
This is:

- Python Course
- Actually this is Programming Course
```

Comments



```
#!/usr/bin/python3

#this is a comment in Python

print ("Hello World") #This is also a comment in Python

""" This is an example of a multiline
comment that spans multiple lines
...
"""


print ("Let me try triple quotes")
'''
I am also comment
in muliple lines

'''
```

output

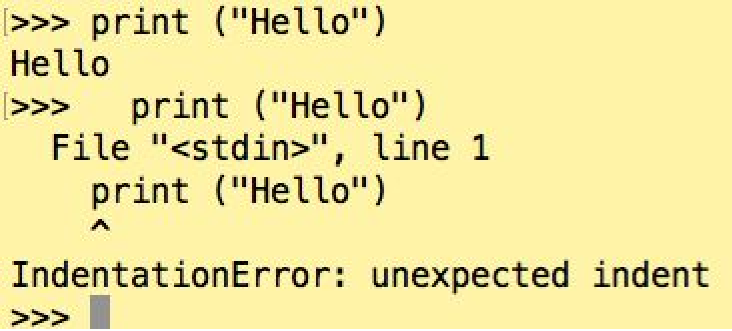


```
Hello World
Let me try triple quotes
```

comments.py

Coding Style

https://www.python.org/dev/peps/pep-0008/



```
>>> print ("Hello")
Hello
>>>   print ("Hello")
  File "<stdin>", line 1
    print ("Hello")
    ^
IndentationError: unexpected indent
>>> 
```

- Python uses indentation to indicate blocks, instead of {} or similar characters
- Makes the code simpler and more readable

Mathematics

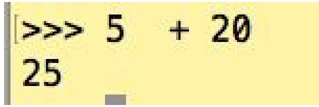
Python Supports - Integers, Float, Complex

Addition of Numbers	+
Subtraction of Numbers	-
Multiplication of Numbers	*
Division of Numbers	/
Floored Division of Numbers	//
Exponentiation, Power of Numbers	* *
Modulo, Remainder of Numbers	%

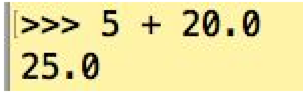
Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Mathematics (Cont ...)

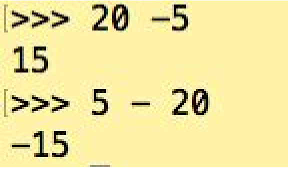
Sum of numbers



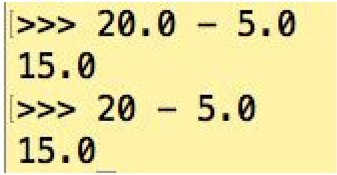
```
>>> 5  + 20
25
```



```
>>> 5 + 20.0
25.0
```



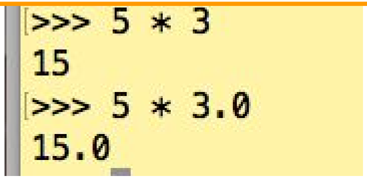
```
>>> 20 -5
15
>>> 5 - 20
-15
```



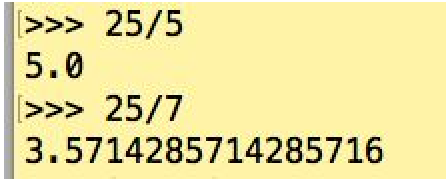
```
>>> 20.0 - 5.0
15.0
>>> 20 - 5.0
15.0
```

Difference of Numbers

Multiplication of Numbers



```
>>> 5 * 3
15
>>> 5 * 3.0
15.0
```

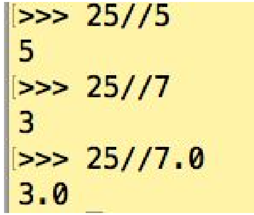


```
>>> 25/5
5.0
>>> 25/7
3.5714285714285716
```

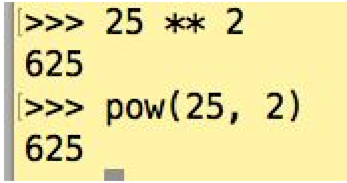
Division of Numbers

Mathematics (Cont …)

Floored Division



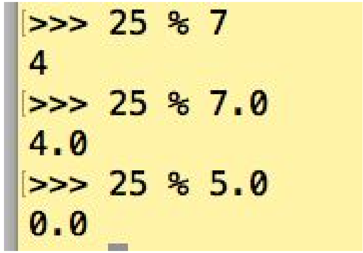
```
>>> 25//5
5
>>> 25//7
3
>>> 25//7.0
3.0
```



```
>>> 25 ** 2
625
>>> pow(25, 2)
625
```

Exponentiation, Power of Numbers

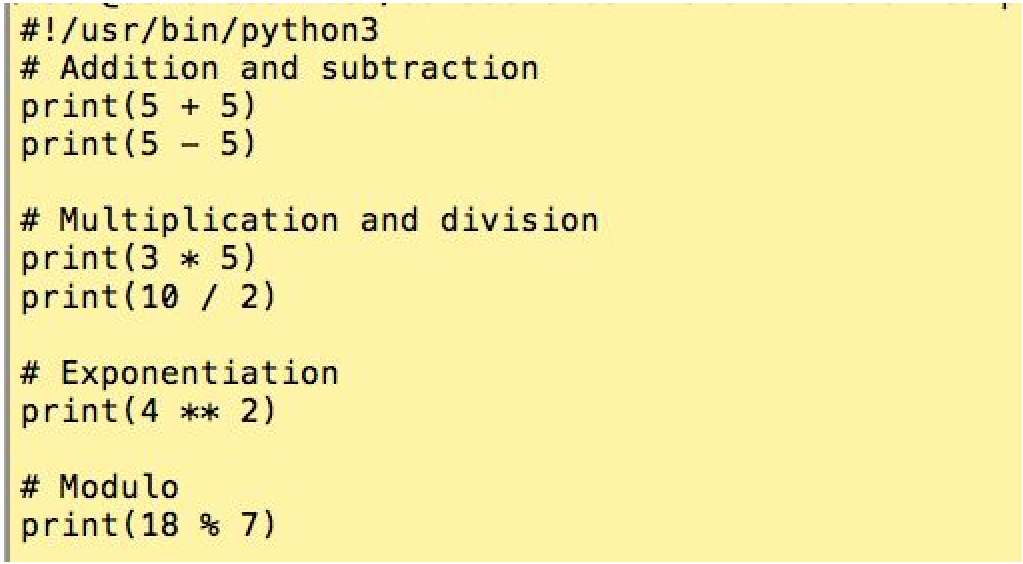
Modulo, Remainder Operation



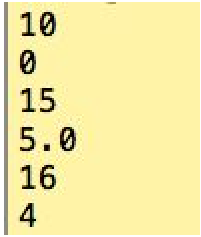
```
>>> 25 % 7
4
>>> 25 % 7.0
4.0
>>> 25 % 5.0
0.0
```

Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

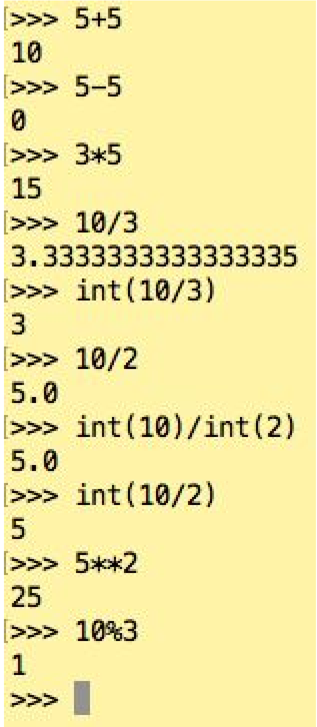
Mathematics (Cont…)



```
#!/usr/bin/python3
# Addition and subtraction
print(5 + 5)
print(5 - 5)

# Multiplication and division
print(3 * 5)
print(10 / 2)

# Exponentiation
print(4 ** 2)

# Modulo
print(18 % 7)
```



```
10
0
15
5.0
16
4
```



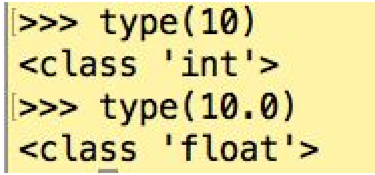
```
>>> 5+5
10
>>> 5-5
0
>>> 3*5
15
>>> 10/3
3.3333333333333335
>>> int(10/3)
3
>>> 10/2
5.0
>>> int(10)/int(2)
5.0
>>> int(10/2)
5
>>> 5**2
25
>>> 10%3
1
>>>
```

mathematics.py

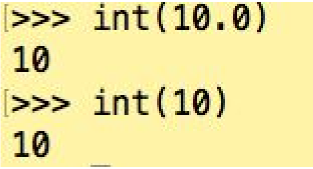
Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

A few Built-in Methods

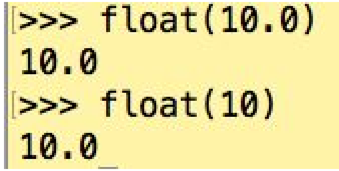
- print
- type
- int
- float



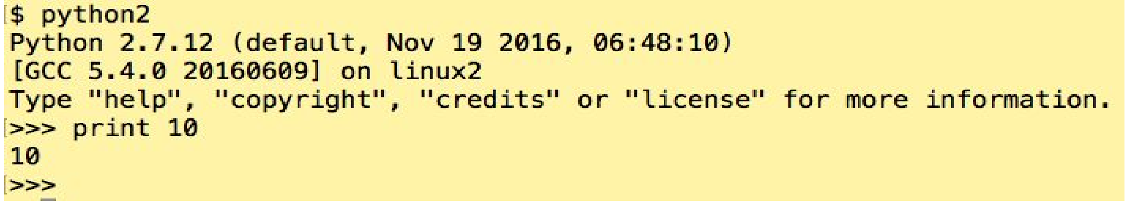
```
>>> type(10)
<class 'int'>
>>> type(10.0)
<class 'float'>
```



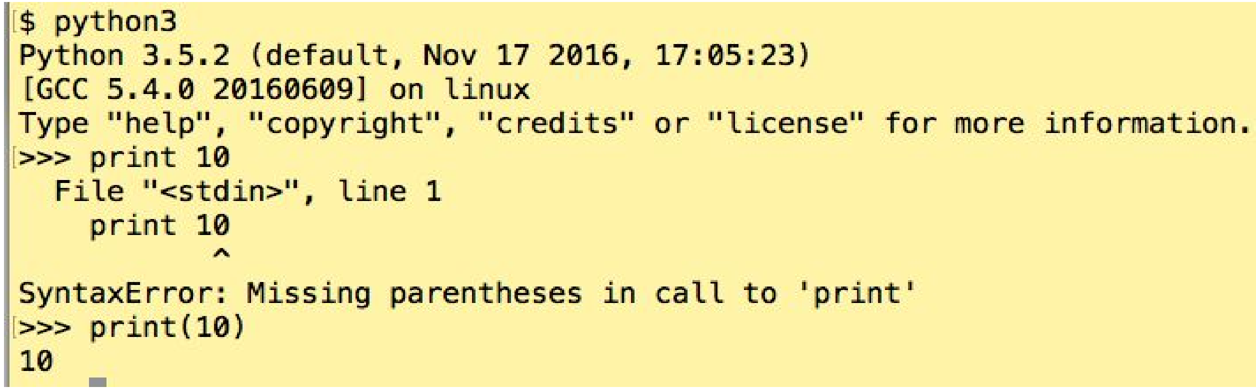
```
>>> int(10.0)
10
>>> int(10)
10
```



```
>>> float(10.0)
10.0
>>> float(10)
10.0
```



```
$ python2
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print 10
10
>>>
```



```
$ python3
Python 3.5.2 (default, Nov 17 2016, 17:05:23)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print 10
  File "<stdin>", line 1
    print 10
           ^
SyntaxError: Missing parentheses in call to 'print'
>>> print(10)
10
```

Task 2

Suppose you have ₹ 1000. You have invested that with a 10% return per year scheme

After one year, you earnings = 1000×1.1= ₹ 1100

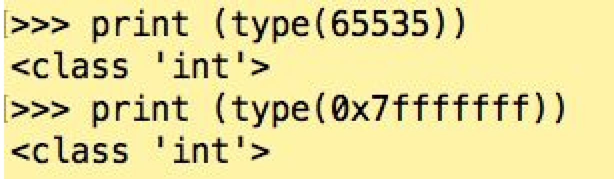
After two years, earnings = 1000×1.1×1.1= ₹ 1210

- How much money you end up earning after 5 years?
- How much money you end up earning after 10 years?

Develop a script called “my_earnings.py” to handle above questions

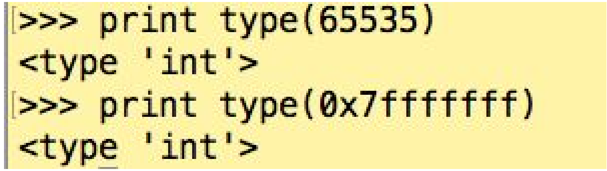
Quiz

1. Which Version of Python is being used in following Code?



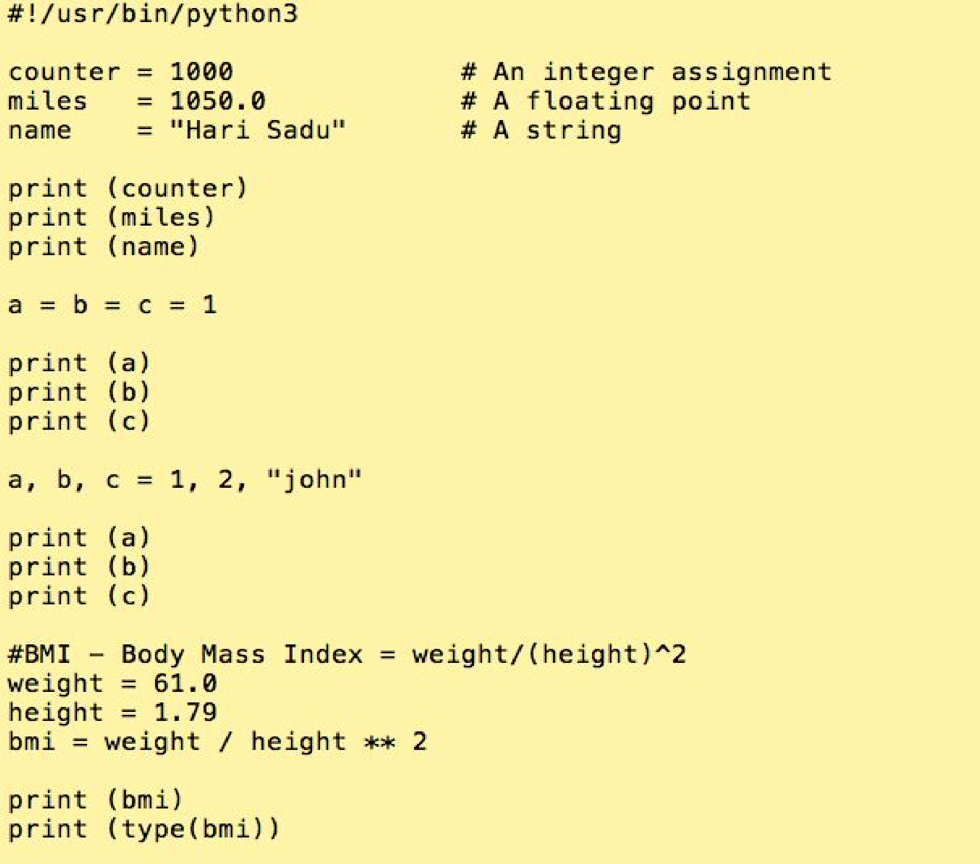
```
>>> print (type(65535))
<class 'int'>
>>> print (type(0x7fffffff))
<class 'int'>
```

2. Which Version of Python is being used in following Code?

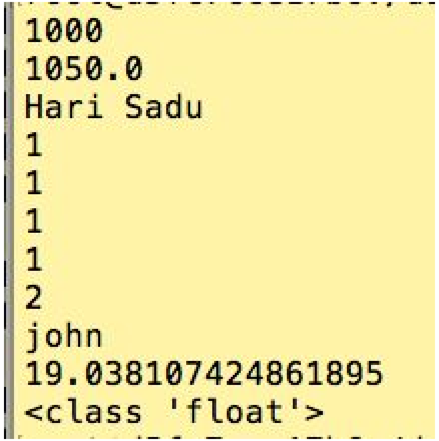


```
>>> print type(65535)
<type 'int'>
>>> print type(0x7fffffff)
<type 'int'>
```

variables



```
#!/usr/bin/python3

counter = 1000                 # An integer assignment
miles   = 1050.0               # A floating point
name    = "Hari Sadu"          # A string

print (counter)
print (miles)
print (name)

a = b = c = 1

print (a)
print (b)
print (c)

a, b, c = 1, 2, "john"

print (a)
print (b)
print (c)

#BMI - Body Mass Index = weight/(height)^2
weight = 61.0
height = 1.79
bmi = weight / height ** 2

print (bmi)
print (type(bmi))
```



```
1000
1050.0
Hari Sadu
1
1
1
1
2
john
19.03810742486189 5
<class 'float'>
```

variables.py

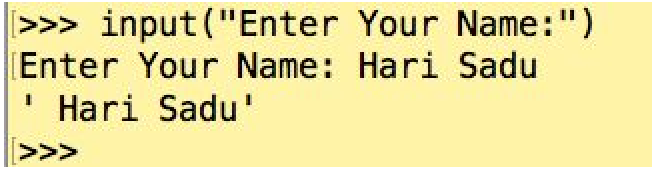
Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Task - 3

- Develop a script called “my_variable earnings.py” to handle questions below
  a. Create a variable *savings* equal to 1000
  b. Create a variable *interest* equal to 1.10.
  c. Use *savings* and *interest* variables to calculate the amount of money you end up earning after 8 years.
- Store the result in a new variable, *earnings*
- print out the value of *earnings*

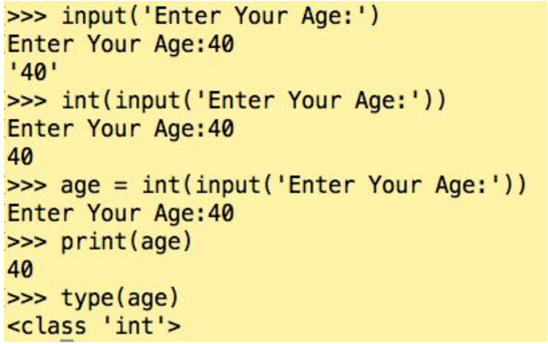
Interact with Script

Python has many built-in functions - input



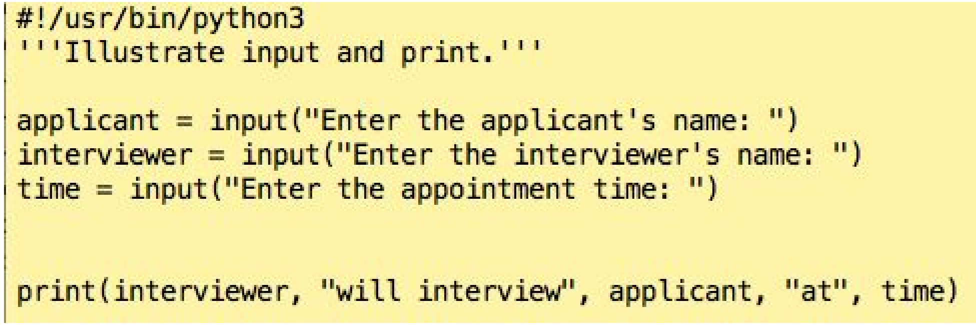
```
>>> input("Enter Your Name:")
Enter Your Name: Hari Sadu
' Hari Sadu'
>>>
```

To deal with numbers - you need to apply method called “int”

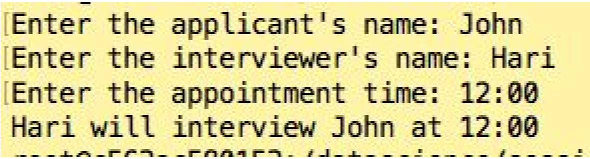


```
>>> input('Enter Your Age:')
Enter Your Age:40
'40'
>>> int(input('Enter Your Age:'))
Enter Your Age:40
40
>>> age = int(input('Enter Your Age:'))
Enter Your Age:40
>>> print(age)
40
>>> type(age)
<class 'int'>
```

Interact with Script (Cont …)

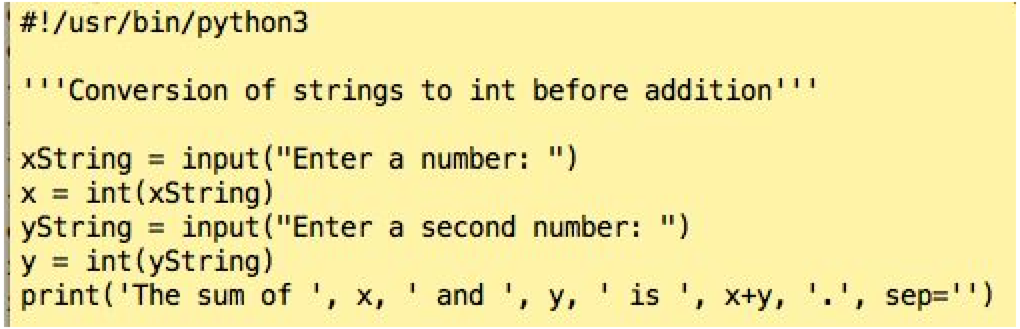


```
#!/usr/bin/python3
'''Illustrate input and print.'''

applicant = input("Enter the applicant's name: ")
interviewer = input("Enter the interviewer's name: ")
time = input("Enter the appointment time: ")


print(interviewer, "will interview", applicant, "at", time)
```

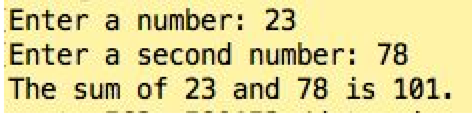


```
Enter the applicant's name: John
Enter the interviewer's name: Hari
Enter the appointment time: 12:00
Hari will interview John at 12:00
```

interview.py



```
#!/usr/bin/python3

'''Conversion of strings to int before addition'''

xString = input("Enter a number: ")
x = int(xString)
yString = input("Enter a second number: ")
y = int(yString)
print('The sum of ', x, ' and ', y, ' is ', x+y, '.', sep='')
```



```
Enter a number: 23
Enter a second number: 78
The sum of 23 and 78 is 101.
```

addition.py

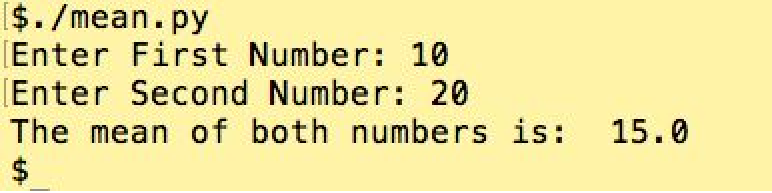
Task - 4

Develop a script “mean.py” which takes two numbers as input from user.

The script creates two variables (number_one & number_two).

It calculates mean, let us store the value in number_mean variable.

At the end, print number_mean.



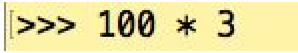
```
[$./mean.py
[Enter First Number: 10
[Enter Second Number: 20
 The mean of both numbers is:  15.0
 $_
```

Aegis

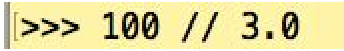
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Quiz

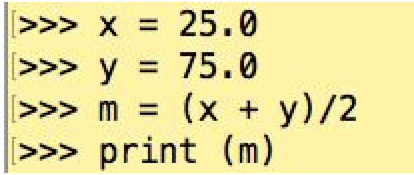
What’s the output of following code snippets?

1. 

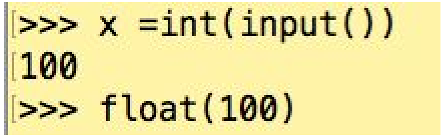
```
>>> 100 * 3
```

2. 

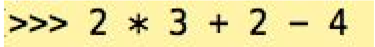
```
>>> 100 // 3.0
```

3. 

```
>>> x = 25.0
>>> y = 75.0
>>> m = (x + y)/2
>>> print (m)
```

4. 

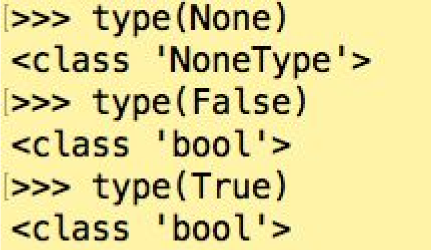
```
>>> x =int(input())
100
>>> float(100)
```

5. 

```
>>> 2 * 3 + 2 - 4
```

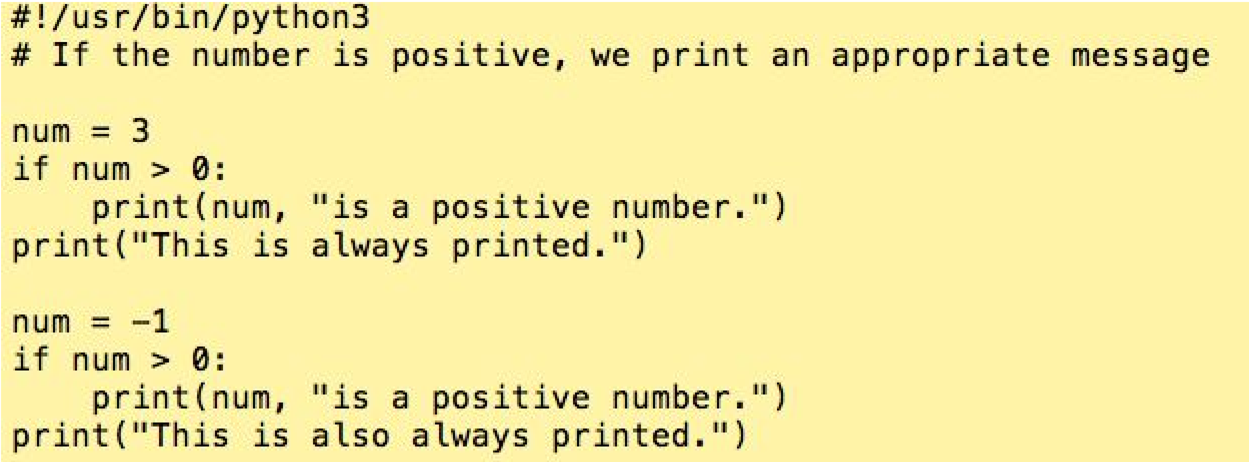
Built-in Constants

- **False**
  - The false value of the bool type
- **True**
  - The true value of the bool type
- **None**
  - The sole value of the type NoneType

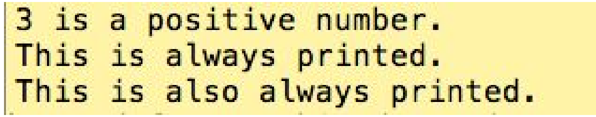


```
>>> type(None)
<class 'NoneType'>
>>> type(False)
<class 'bool'>
>>> type(True)
<class 'bool'>
```

Making Decisions



```
#!/usr/bin/python3
# If the number is positive, we print an appropriate message

num = 3
if num > 0:
    print(num, "is a positive number.")
print("This is always printed.")

num = -1
if num > 0:
    print(num, "is a positive number.")
print("This is also always printed.")
```

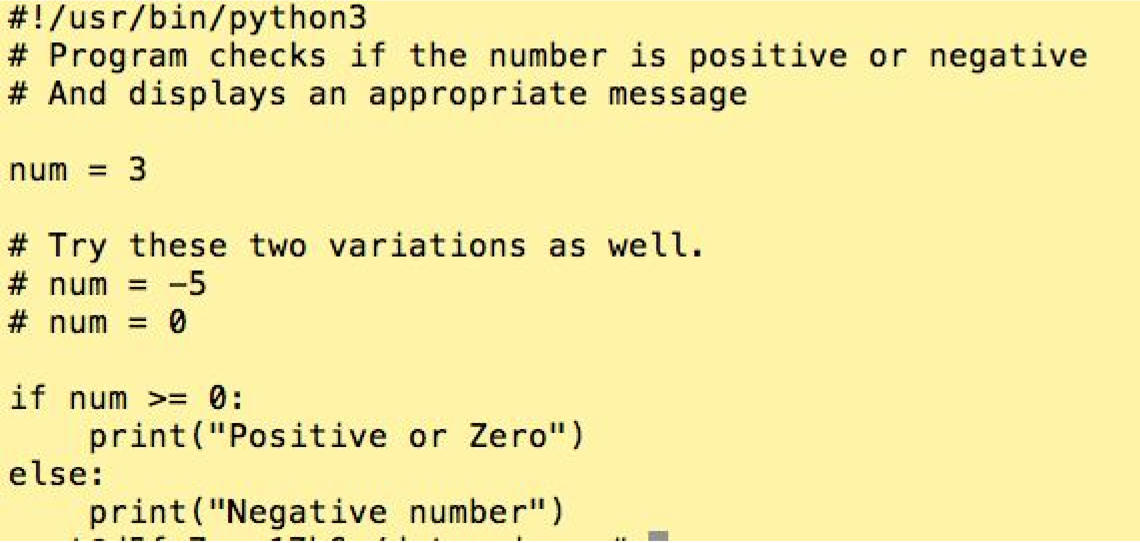


```
3 is a positive number.
This is always printed.
This is also always printed.
```

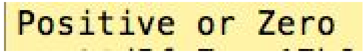
if_demo.py

Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Making Decisions (Cont…)



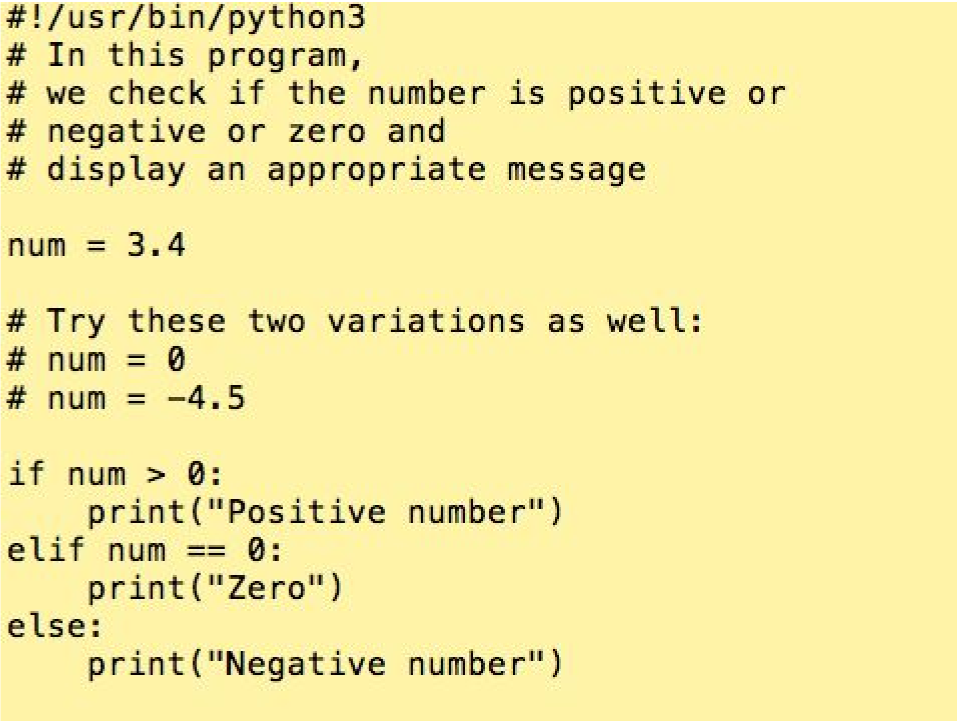
```
#!/usr/bin/python3
# Program checks if the number is positive or negative
# And displays an appropriate message

num = 3

# Try these two variations as well.
# num = -5
# num = 0

if num >= 0:
    print("Positive or Zero")
else:
    print("Negative number")
```



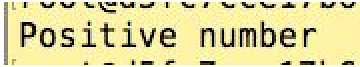
```
Positive or Zero
```

if_else_demo.py

Making Decisions (Cont...)



```
#!/usr/bin/python3
# In this program,
# we check if the number is positive or
# negative or zero and
# display an appropriate message

num = 3.4

# Try these two variations as well:
# num = 0
# num = -4.5

if num > 0:
    print("Positive number")
elif num == 0:
    print("Zero")
else:
    print("Negative number")
```



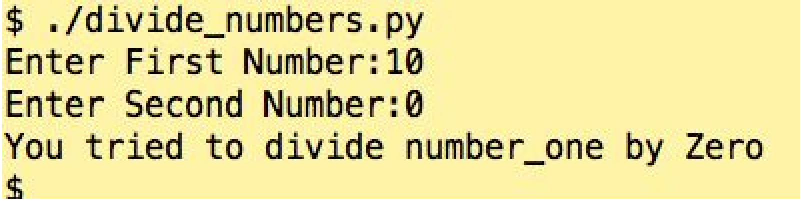
```
Positive number
```

if_elif_demo.py

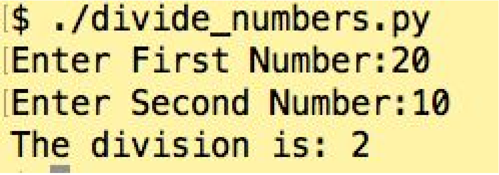
Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Task - 5

Develop a script “divide_numbers.py” which takes two numbers as input from user. The script creates two variables (number_one & number_two). The script prints “You tried to divide number_one by Zero” if number_two is Zero. If number_two is not zero then it prints integer part of quotient.



```
$ ./divide_numbers.py
Enter First Number:10
Enter Second Number:0
You tried to divide number_one by Zero
$
```

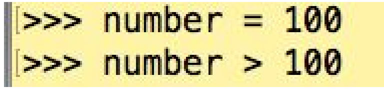


```
$ ./divide_numbers.py
Enter First Number:20
Enter Second Number:10
The division is: 2
```

Quiz

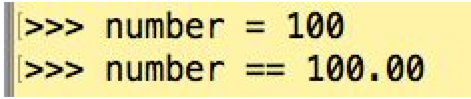
What’s the output of following code snippets?

1.



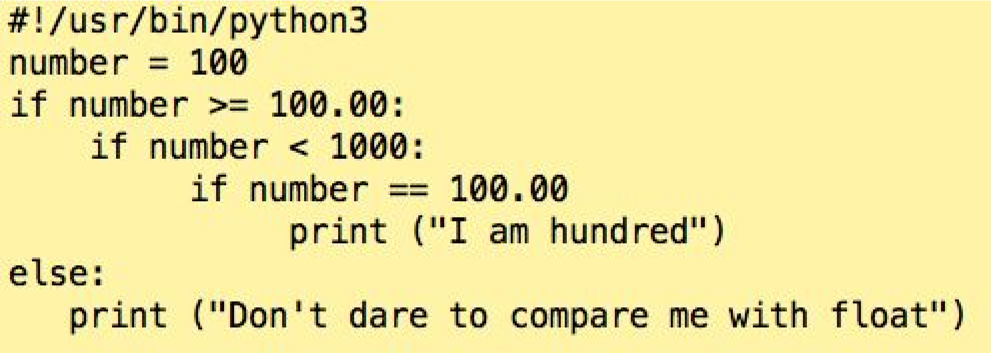
```
[>>> number = 100
[>>> number > 100
```

2.



```
[>>> number = 100
[>>> number == 100.00
```

3.



```
#!/usr/bin/python3
number = 100
if number >= 100.00:
     if number < 1000:
          if number == 100.00
               print ("I am hundred")
else:
   print ("Don't dare to compare me with float")
```

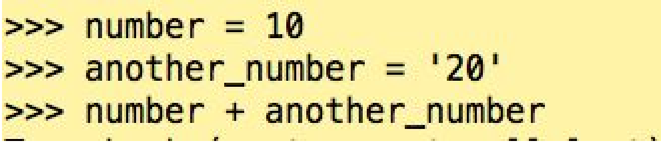
4. Is Python Scripting Language?

5. Is Python High Level Language?

Quiz

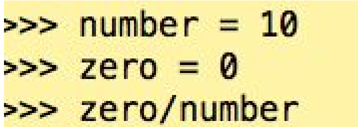
What is the output of following code?

1.



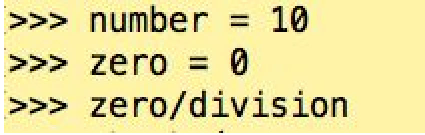
```
>>> number = 10
>>> another_number = '20'
>>> number + another_number
```

2



```
>>> number = 10
>>> zero = 0
>>> zero/number
```

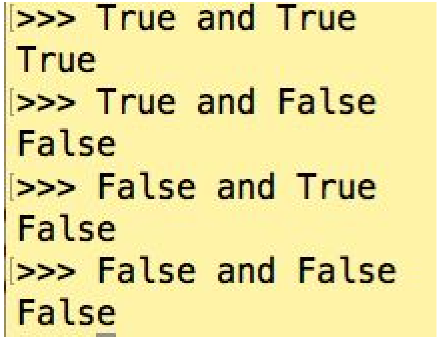
3



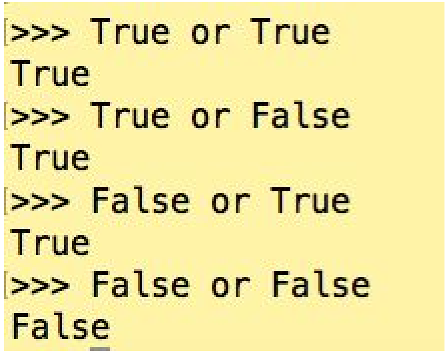
```
>>> number = 10
>>> zero = 0
>>> zero/division
```

Basic Operations

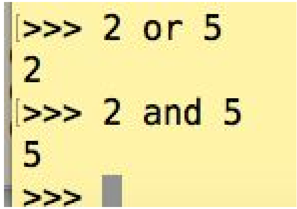
“and” and “or” are very frequently used operators



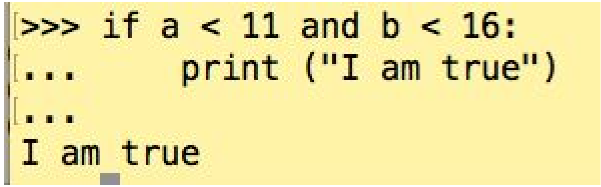
```
>>> True and True
True
>>> True and False
False
>>> False and True
False
>>> False and False
False
```



```
>>> True or True
True
>>> True or False
True
>>> False or True
True
>>> False or False
False
```



```
>>> 2 or 5
2
>>> 2 and 5
5
>>>
```



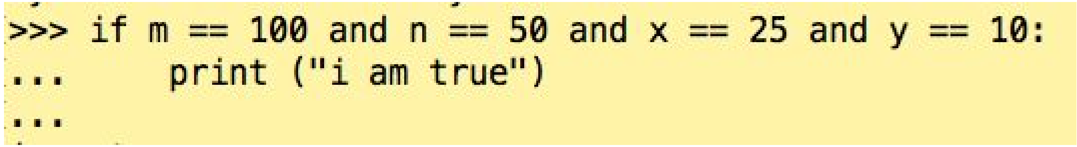
```
>>> if a < 11 and b < 16:
...     print ("I am true")
...
I am true
```

Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

Quiz

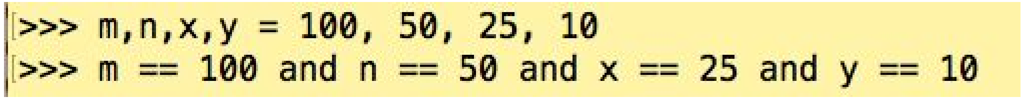
What is the output of the following code?

1.



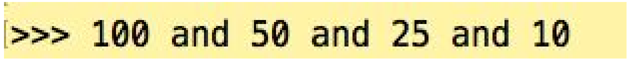
```
>>> if m == 100 and n == 50 and x == 25 and y == 10:
...     print ("i am true")
...
```

2



```
>>> m,n,x,y = 100, 50, 25, 10
>>> m == 100 and n == 50 and x == 25 and y == 10
```

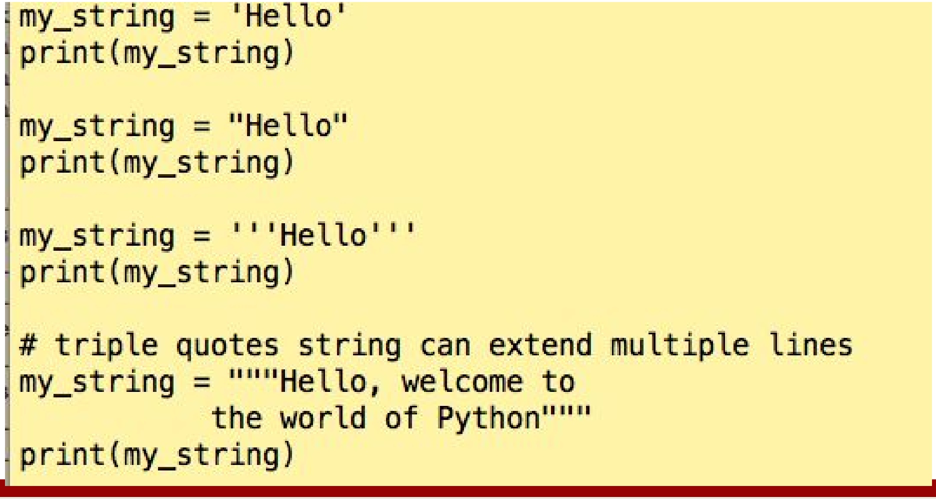
3



```
>>> 100 and 50 and 25 and 10
```

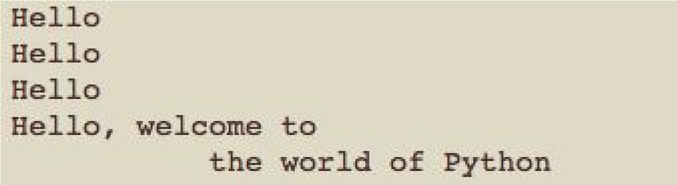
Strings

A string is a sequence of characters. A character is simply a symbol. For example, the English language has 26 characters. The same symbols - computer stores in numbers internally (0's and 1's) (encoding/decoding process - ASCII, Unicode)



```
my_string = 'Hello'
print(my_string)

my_string = "Hello"
print(my_string)

my_string = '''Hello'''
print(my_string)

# triple quotes string can extend multiple lines
my_string = """Hello, welcome to
           the world of Python"""
print(my_string)
```

strings_one.py



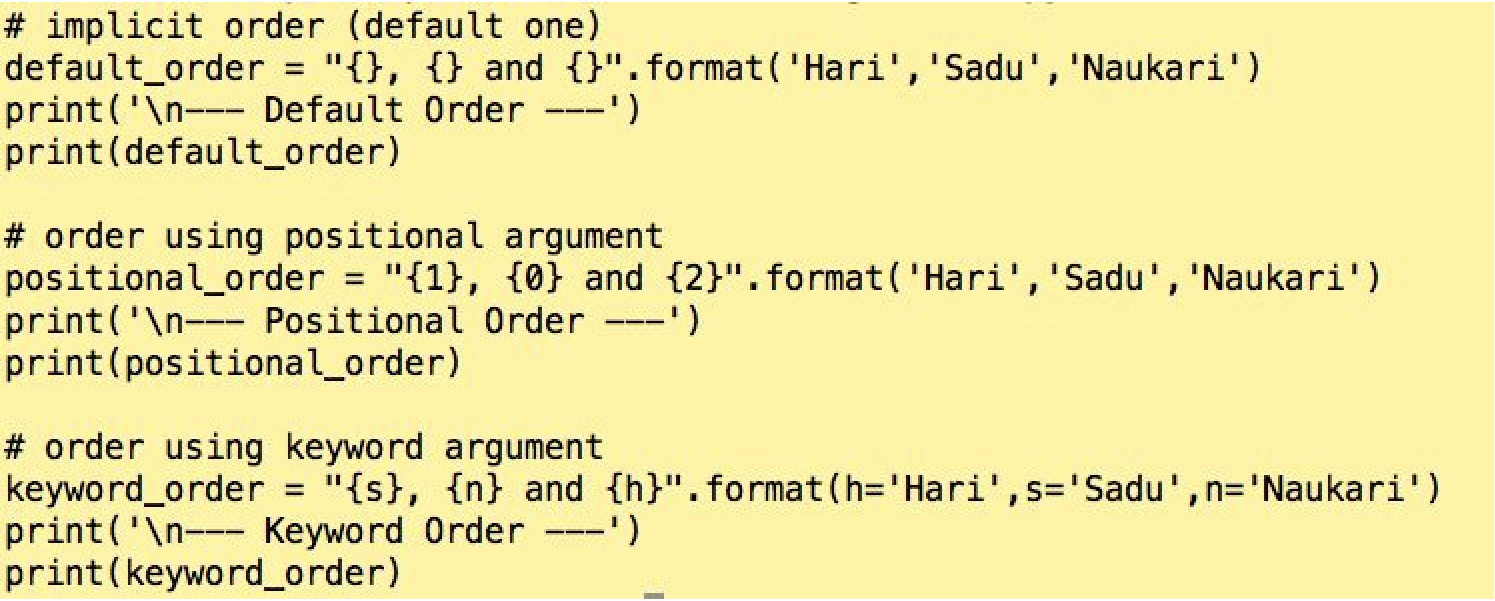
```
Hello
Hello
Hello
Hello, welcome to
           the world of Python
```

Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

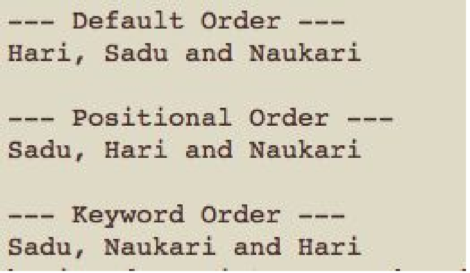
Strings - Important Methods

- s.lower(), s.upper() -- returns the lowercase or uppercase version of the string
- s.strip() -- returns a string with whitespace removed from the start and end
- s.isalpha()/s.isdigit()/s.isspace()... -- tests if all the string chars are in the various character classes
- s.startswith('other'), s.endswith('other') -- tests if the string starts or ends with the given other string
- s.find('other') -- searches for the given other string (not a regular expression) within s, and returns the first index where it begins or -1 if not found
- s.replace('old', 'new') -- returns a string where all occurrences of 'old' have been replaced by 'new'
- s.split('delim') -- returns a list of substrings separated by the given delimiter. T
- s.join(list) -- opposite of split()

String Formatting - Text Displaying



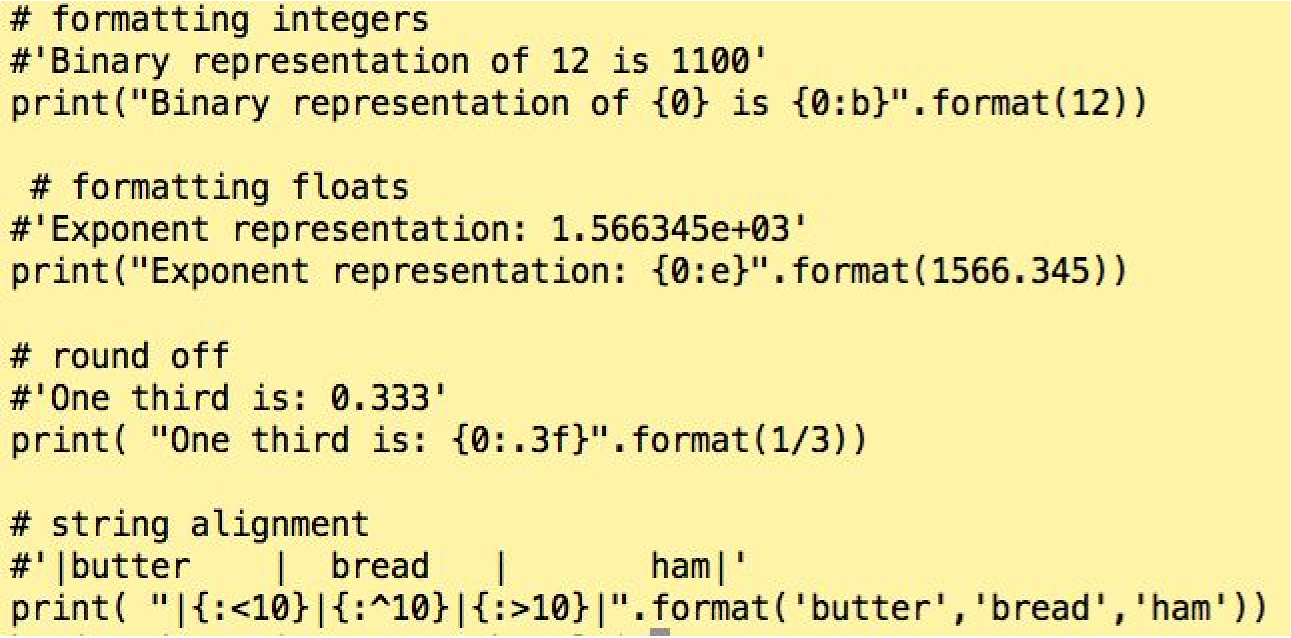
```
# implicit order (default one)
default_order = "{}, {} and {}".format('Hari','Sadu','Naukari')
print('\n--- Default Order ---')
print(default_order)

# order using positional argument
positional_order = "{1}, {0} and {2}".format('Hari','Sadu','Naukari')
print('\n--- Positional Order ---')
print(positional_order)

# order using keyword argument
keyword_order = "{s}, {n} and {h}".format(h='Hari',s='Sadu',n='Naukari')
print('\n--- Keyword Order ---')
print(keyword_order)
```



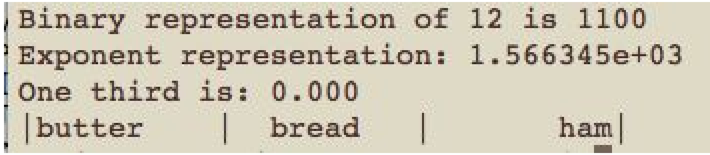
```
--- Default Order ---
Hari, Sadu and Naukari

--- Positional Order ---
Sadu, Hari and Naukari

--- Keyword Order ---
Sadu, Naukari and Hari
```

Aegis
SCHOOL OF BUSINESS
SCHOOL OF DATA SCIENCE
SCHOOL OF TELECOMMUNICATION

String Format (Cont ...)



```
# formatting integers
#'Binary representation of 12 is 1100'
print("Binary representation of {0} is {0:b}".format(12))

 # formatting floats
#'Exponent representation: 1.566345e+03'
print("Exponent representation: {0:e}".format(1566.345))

# round off
#'One third is: 0.333'
print( "One third is: {0:.3f}".format(1/3))

# string alignment
#'|butter    |  bread   |       ham|'
print( "|{:<10}|{:^10}|{:>10}|".format('butter','bread','ham'))
```



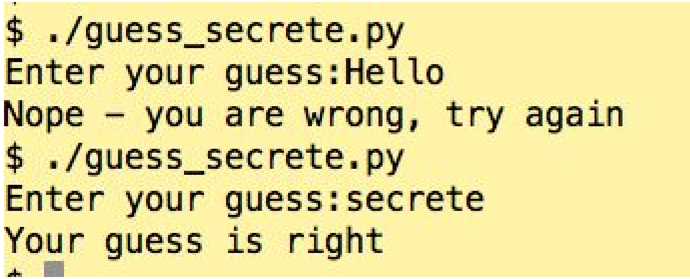
```
Binary representation of 12 is 1100
Exponent representation: 1.566345e+03
One third is: 0.000
|butter    |  bread   |       ham|
```

LAB Assignment

Q. 1 - Develop script “generic_maximum.py” to find maximum of any two numbers. The numbers are entered interactively by the user.

Q. 2 - Develop script “generic_minimum.py” to find minimum of any two numbers. The numbers are entered interactively by the user.

Q. 3 - Develop script “guess_secrete.py” - similar to following screen. It asks to guess secrete word.



```
$ ./guess_secrete.py
Enter your guess:Hello
Nope - you are wrong, try again
$ ./guess_secrete.py
Enter your guess:secrete
Your guess is right
$ 
```

Q. 4 Correct “lab_one.py” script from shared Repository.

LAB Assignment (Cont …)

PE ratio is one of the most widely used tools for **stock** market

It is calculated by dividing the current market price of the **stock** by its earning per share (EPS).

EPS is the portion of a company's profit allocated to each outstanding share of common stock. That is,

EPS = net income ÷ average outstanding common shares

A company had ₹ 40 million of net income, paid out ₹ 2 million in preferred dividends and had on average 10 million outstanding shares for that year. What is its EPS? The stock is traded at ₹ 38.00, what's the PE Ratio?

Q. 5 Develop a script "calculate_pe.py" to take input "net income", "preferred dividend", "average common stocks" and "stock price". It should calculate and print P/E

Thank you