



INTRODUCTION

Python



Programming Languages

- Computers are built for one purpose - to do things for us
- But we need to speak their language to describe what we want do
- How do we communicate with the computer?
 - *Programming languages*

Python Programming Language

- How did the language Python get its name?
- Named for the British comedy group Monty Python (really!)
- Guido van Rossum invented the Python programming language in early 1990s

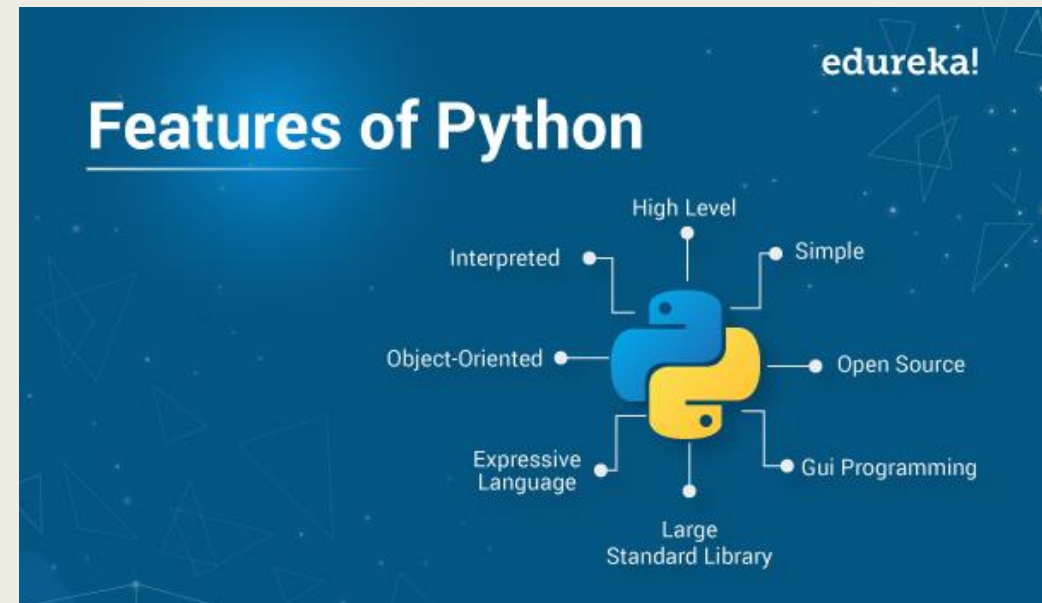


Download Python 3.X

- <https://www.python.org/downloads/>
- There are different versions for Windows, Mac and Linux
- Please download appropriate version for your own computer

Why Python

- Interpretive and object oriented programming language
- Shallow learning curve
- Great support for many different area
- Large community



Interpreted VS Compiled

- Python uses an interpreter, translates high-level language one statement at a time into machine language and then runs
- Check the video → <https://youtu.be/I1f45REi3k4>
- Check the web → <https://www.freecodecamp.org/news/compiled-versus-interpreted-languages/>

Machine code	Assembly code	Description
001 1 000010	LOAD #2	Load the value 2 into the Accumulator
010 0 001101	STORE 13	Store the value of the Accumulator in memory location 13
001 1 000101	LOAD #5	Load the value 5 into the Accumulator
010 0 001110	STORE 14	Store the value of the Accumulator in memory location 14
001 0 001101	LOAD 13	Load the value of memory location 13 into the Accumulator
011 0 001110	ADD 14	Add the value of memory location 14 to the Accumulator
010 0 001111	STORE 15	Store the value of the Accumulator in memory location 15
111 0 000000	HALT	Stop execution

Address	Machine Language				Assembly Language	
0000 0000	0000	0000	0000	0000	TOTAL	.BLOCK 1
0000 0001	0000	0000	0000	0010	ABC	.WORD 2
0000 0010	0000	0000	0000	0011	XYZ	.WORD 3
0000 0011	0001	1101	0000	0001		LOAD REGD, ABC
0000 0100	0001	1110	0000	0010		LOAD REGE, XYZ
0000 0101	0101	1111	1101	1110		ADD REGF, REGD, REGE
0000 0110	0010	1111	0000	0000		STORE REGF, TOTAL
0000 0111	1111	0000	0000	0000		HALT

Object Oriented Programming Language

- Use the concept of "objects", which can contain data and code: data in the form of fields (often known as attributes or *properties*) code in the form of procedures (often known as *methods*)
- Check the video → https://youtu.be/m_MQYyJpljg
- Check the web → <https://www.indeed.com/career-advice/career-development/what-is-object-oriented-programming>

Python: Object-Oriented Programming Language

LOOPS!

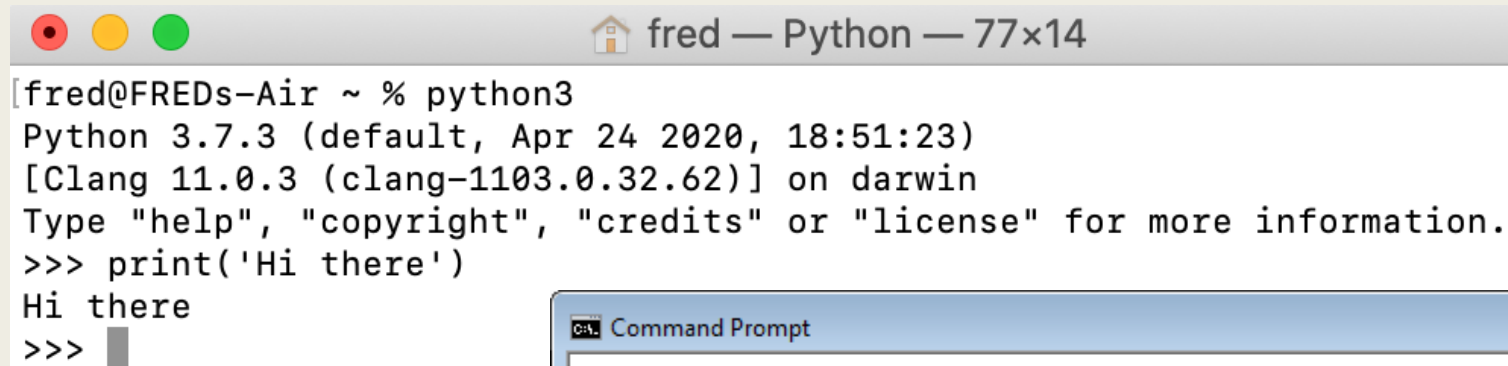
www.tekkiehead.com

Interactive VS Script

- There are two ways to use the interpreter: **interactive** mode and **script** mode
- **Interactive**
You type directly to Python one line at a time and it responds
- **Script**
You enter a sequence of statements (lines) into a file using a text editor and tell Python to execute the statements in the file

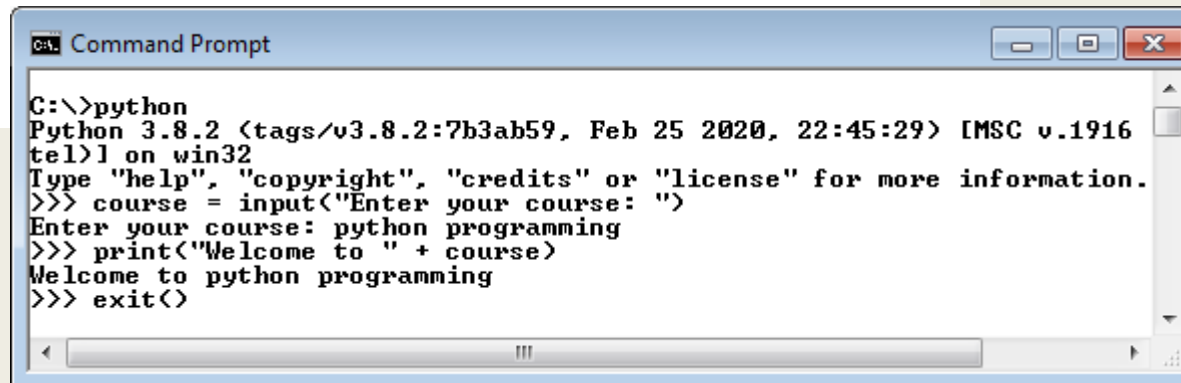
Running Code in the Interactive Mode

- Interactive Python is good for experimenting with short expressions or statements



A macOS terminal window titled "fred — Python — 77x14". The prompt is "[fred@FREDs-Air ~ % python3]". The output shows Python 3.7.3 (default, Apr 24 2020, 18:51:23) [Clang 11.0.3 (clang-1103.0.32.62)] on darwin. The user enters "Type 'help', 'copyright', 'credits' or 'license' for more information." followed by ">>> print('Hi there')". The output is "Hi there". The prompt ">>>" is followed by a cursor.

```
[fred@FREDs-Air ~ % python3
Python 3.7.3 (default, Apr 24 2020, 18:51:23)
[Clang 11.0.3 (clang-1103.0.32.62)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hi there')
Hi there
>>> █
```

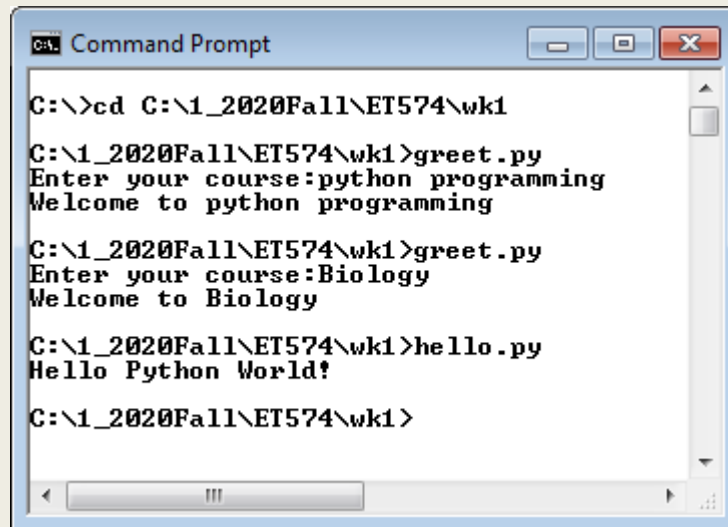


A Windows Command Prompt window titled "Command Prompt". The prompt is "C:\>python". The output shows Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916] on win32. The user enters "Type 'help', 'copyright', 'credits' or 'license' for more information." followed by ">>> course = input('Enter your course: ')". The input is "python programming". The user then enters ">>> print('Welcome to ' + course)". The output is "Welcome to python programming". The user enters ">>> exit()".

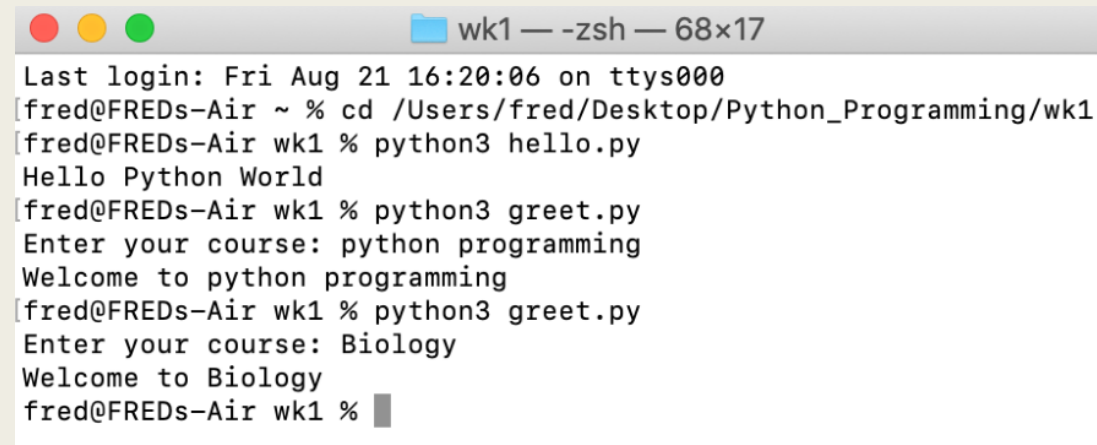
```
C:\>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> course = input("Enter your course: ")
Enter your course: python programming
>>> print("Welcome to " + course)
Welcome to python programming
>>> exit()
```

Running Code in the Script Mode

- Most programs are much longer, so we type them into a file and tell Python to run the commands in the file
- As a convention, we add “.py” as the suffix on the end of these files to indicate they contain Python



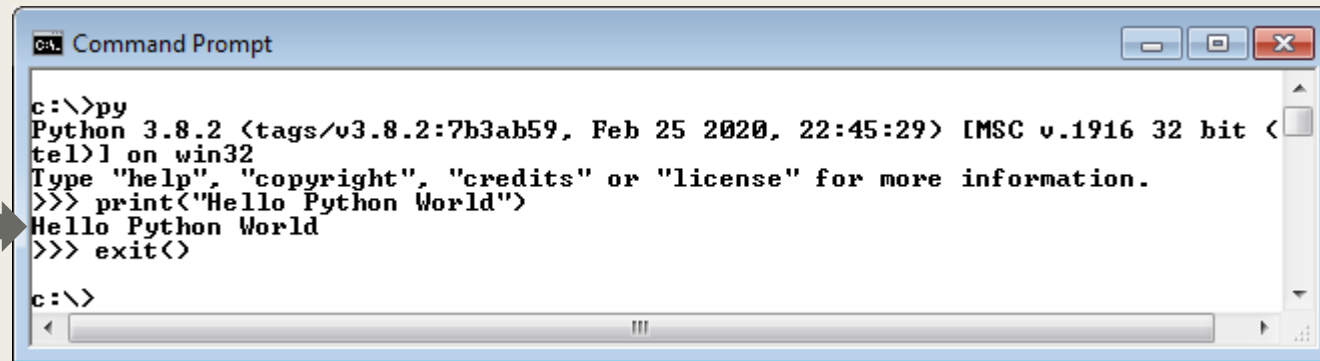
```
C:\>cd C:\1_2020Fall\ET574\wk1
C:\1_2020Fall\ET574\wk1>greet.py
Enter your course:python programming
Welcome to python programming
C:\1_2020Fall\ET574\wk1>greet.py
Enter your course: Biology
Welcome to Biology
C:\1_2020Fall\ET574\wk1>hello.py
Hello Python World!
C:\1_2020Fall\ET574\wk1>
```



```
wk1 — -zsh — 68x17
Last login: Fri Aug 21 16:20:06 on ttys000
fred@FREDs-Air ~ % cd /Users/fred/Desktop/Python_Programming/wk1
fred@FREDs-Air wk1 % python3 hello.py
Hello Python World
fred@FREDs-Air wk1 % python3 greet.py
Enter your course: python programming
Welcome to python programming
fred@FREDs-Air wk1 % python3 greet.py
Enter your course: Biology
Welcome to Biology
fred@FREDs-Air wk1 %
```

An Example in Command Prompt (Windows)

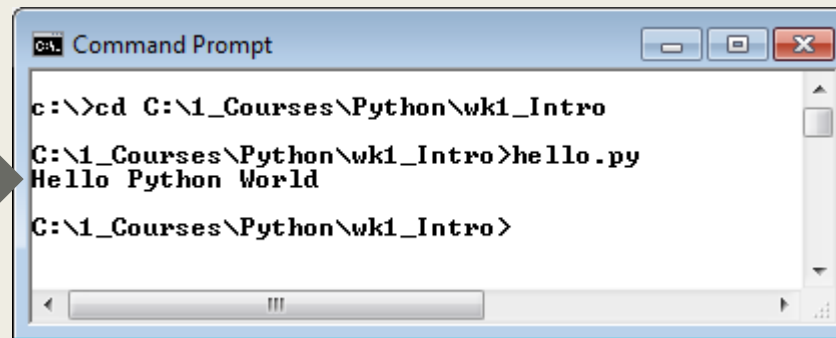
Interactive



```
c:\>py
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit <
tel>] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello Python World")
Hello Python World
>>> exit()

c:\>
```

Script



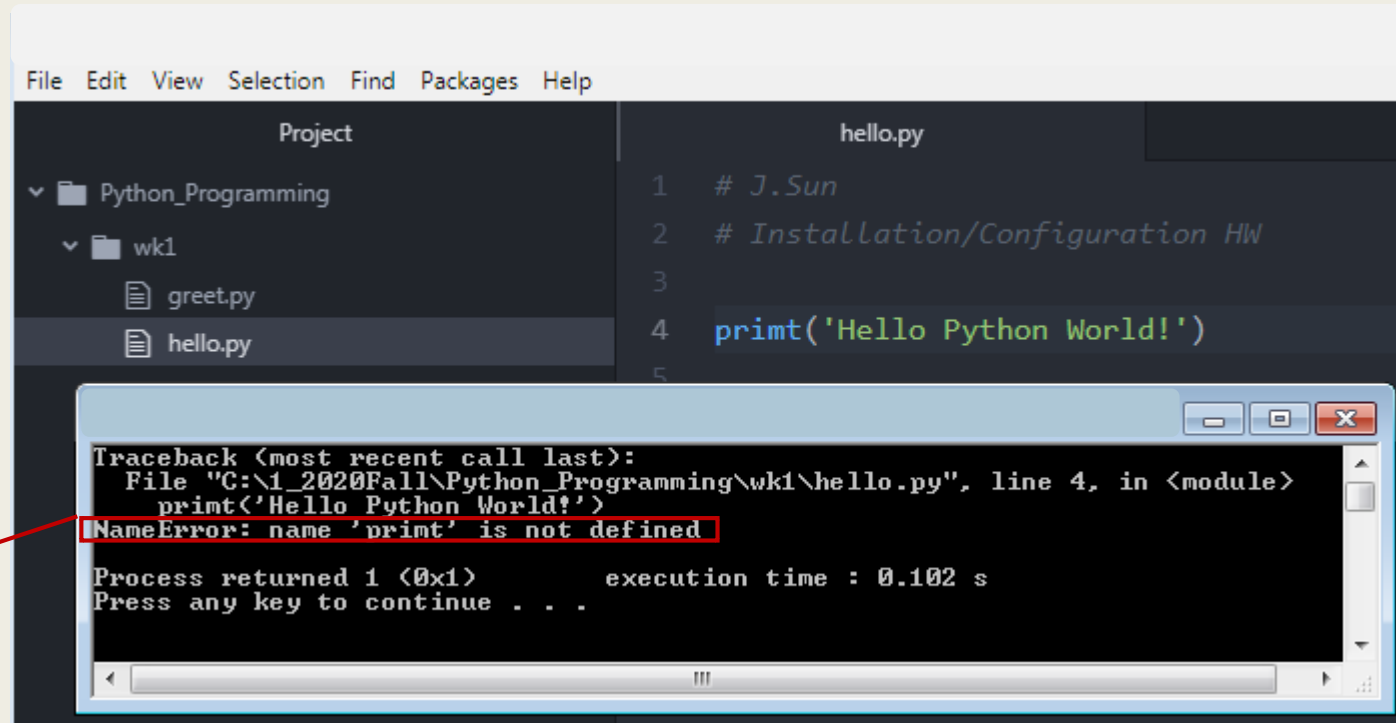
```
c:\>cd C:\1_Courses\Python\wk1_Intro
C:\1_Courses\Python\wk1_Intro>hello.py
Hello Python World
C:\1_Courses\Python\wk1_Intro>
```

Early Learner: Syntax Errors

- Programmers inevitably make typographical errors when editing programs, called syntax errors
- Syntax: rules for forming sentences in a language
- When Python encounters a syntax error in a program, it halts execution with an error message
- Most programs are much longer, so we type them into a file and tell Python to run the commands in the file

Traceback

print



The screenshot shows a Python IDE with a menu bar (File, Edit, View, Selection, Find, Packages, Help) and a project explorer on the left. The project explorer shows a folder 'Python_Programming' containing a subfolder 'wk1' with files 'greet.py' and 'hello.py'. The 'hello.py' file is open in the editor, showing the following code:

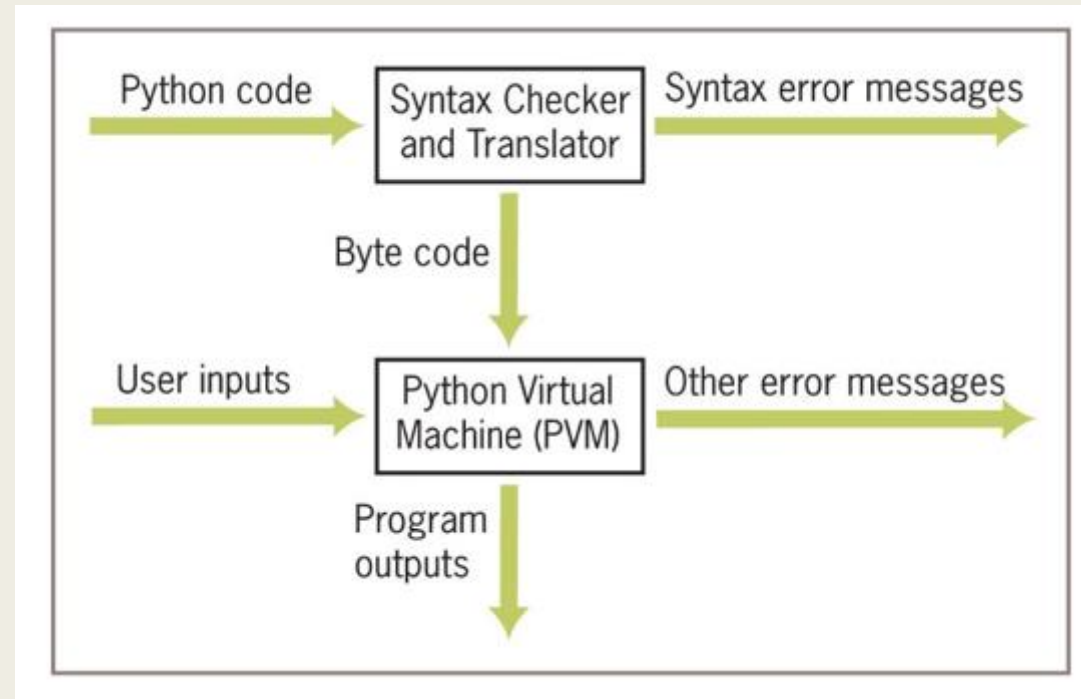
```
1 # J.Sun
2 # Installation/Configuration HW
3
4 print('Hello Python World!')
5
```

A traceback window is open in the foreground, displaying the following error message:

```
Traceback (most recent call last):
  File "C:\1_2020Fall\Python_Programming\wk1\hello.py", line 4, in <module>
    print('Hello Python World!')
NameError: name 'print' is not defined
```

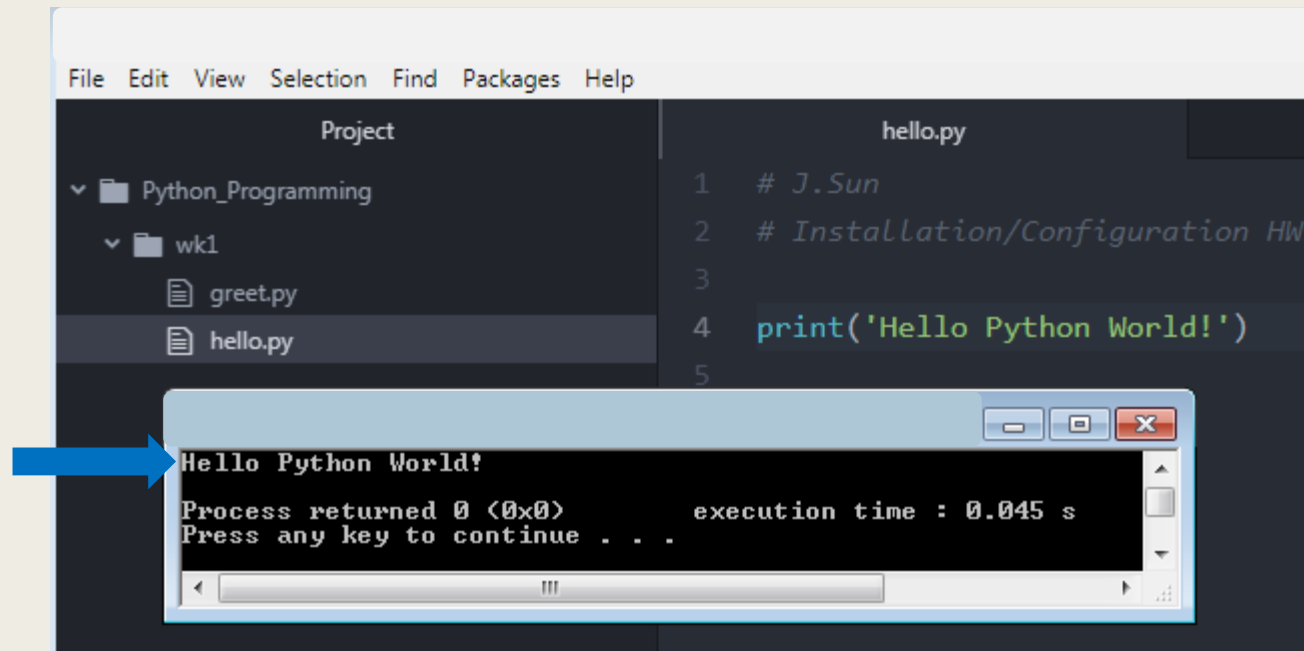
The error message is highlighted with a red box. Below the error message, it says "Process returned 1 (0x1) execution time : 0.102 s" and "Press any key to continue . . .". A red arrow points from the word "print" to the error message.

Behind the Scenes: How Python Works

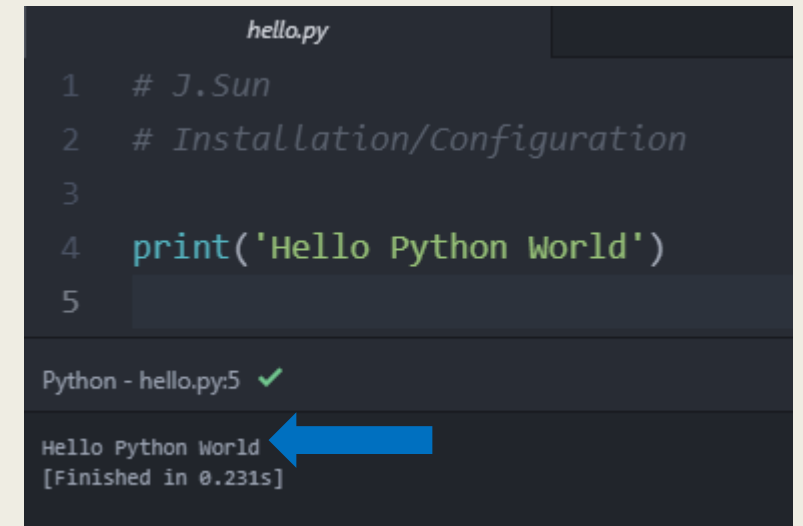


Installation/Configuration

- Use Atom, the text editor, to create the source file (hello.py)
- Run the source code

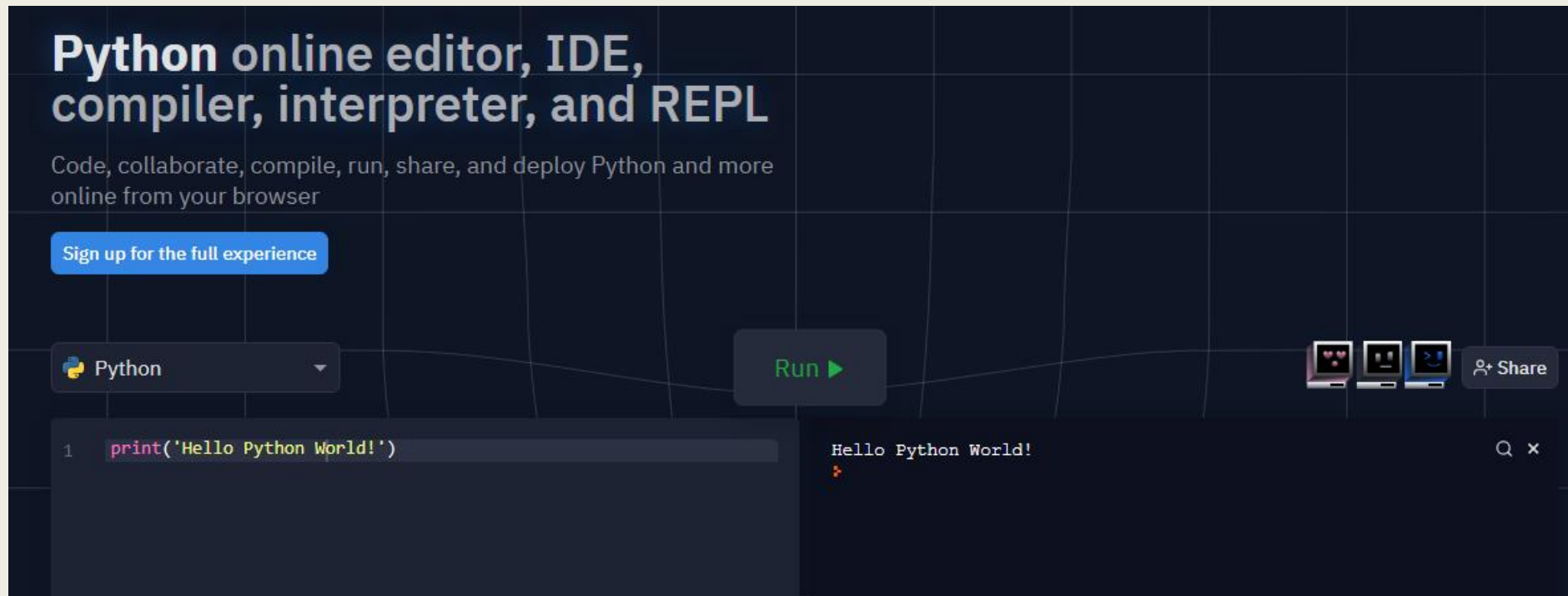


atom-python-run package



script package

Python Online Editor



Summary

- Programming languages such as Python are high-level
- Interpreter translates a Python program to a lower-level form that can be executed on a real computer
- Interactive Python shell provides a command prompt for evaluating and viewing the results of Python expressions and statements
- IDE/IDLE is an integrated development environment that allows the programmer to save programs in files and load them into a shell for testing
- Python scripts are programs that are saved in files and run from a terminal command prompt
- When a Python program is executed, it is translated into byte code
 - *Sent to PVM for further interpretation and execution*

Summary

- Quick overview of Python
- Revisit these concepts throughout the course
- Get your Python Integrated Development Environment (IDE) ready to code

Practice Quiz Instructions

Begin: Practice Quiz

INSTRUCTIONS


Timed Test	This test has a time limit of 20 minutes.
Timer Setting	This test will save and submit automatically when the time expires.
Force Completion	Once started, this test must be completed in one sitting. Do not leave the test before clicking Save and Submit .
Due Date	This Test is due on August 31, 2022 11:59:00 PM EDT. Test cannot be started past this date.

Click **Begin** to start: Practice Quiz. Click **Cancel** to go back.
You'll be previewing this assessment and your results won't be recorded.

A Quizzes

Begin: Quiz 1A - Variables

INSTRUCTIONS

Timed Test	This test has a time limit of 15 minutes.
Timer Setting	This test will save and submit automatically when the time expires.
Force Completion	Once started, this test must be completed in one sitting. Do not leave the test before clicking Save and Submit .
Multiple Attempts	This test allows multiple attempts.  Only quizzes A have multiple attempts
Due Date	This Test is due on September 7, 2022 11:59:00 PM EDT. Test cannot be started past this date.

Click **Begin** to start: Quiz 1A - Variables. Click **Cancel** to go back.
You'll be previewing this assessment and your results won't be recorded.

Preview Test

Preview Test: Practice Quiz

⤴ Test Information

Description

Instructions

Timed Test This test has a time limit of 20 minutes. This test will save and submit automatically when the time expires.
Warnings appear when **half the time, 5 minutes, 1 minute, and 30 seconds** remain. *[The timer does not appear when previewing this test]*

Multiple Attempts Not allowed. This test can only be taken once.

Force Completion Once started, this test must be completed in one sitting. Do not leave the test before clicking **Save and Submit**.
Your answers are saved automatically.

⌵ Question Completion Status:

Taking Test

QUESTION 1

0.25 points ✓ Saved

Python _____.

- ☐ was created by Guido van Rossum, and released in 1991
- ☐ works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- ☐ is a high-level programming language.
- ☒ all of the above

QUESTION 2

0.25 points Save Answer

Code can be copied from Atom/IDE and pasted in the answer area in your quiz.

- ☐ True
- ☐ False

Save and Submit

bbhosted.cuny.edu says

The following questions may be incomplete:

1,2,3,4

Click cancel to return to the test. Click Ok to submit assessment.

OK

Cancel

bbhosted.cuny.edu says

Test submission confirmation: Click Cancel to return to the test. Click OK to submit assessment.

OK

Cancel

Test Submitted: Practice Quiz

Test saved and submitted.

Student:

Test: Practice Quiz

Course:

Started:

Submitted:

Due Date:

Time Used: 9 minutes out of 20 minutes

Click **OK** to review results.

Review Test Submission (1 of 2)


Question 1

0.25 out of 0.25 points

Python _____.

Selected Answer: ☒ all of the above

Correct Answer: ☒ all of the above

Response Feedback: 


Question 2

0.25 out of 0.25 points


Code can be copied from Atom/IDE and pasted in the answer area in your quiz.

Selected Answer: ☒ False

Correct Answer: ☒ False

Response Feedback: 



Review Test Submission (2 of 2)


Status	Needs Grading 
Attempt Score	Grade not available.
Time Elapsed	9 minutes out of 20 minutes
Results Displayed	All Answers, Submitted Answers, Feedback

Question 3

 0 out of 0.25 points

The missing quiz can be made-up.

Selected Answer:  True 

Correct Answer:  False


Response Feedback: Quizzes must be completed once begin.
Quizzes cannot be made up if missed or an internet/Blackboard error occurs.
Check the syllabus and ET574 Start Here/00 Start Here/Course Documents for more details.


Question 4

 Needs Grading

Write **one line** of code to display "Hello World" to the output console.

Selected Answer: `print("Hello World")`

Correct Answer: Use `print()` function and put the displayed text in the quotation marks.
 See Ex1_Output.py for more details.

Response Feedback: [None Given] 

Assignments



Email & Announcements Confirmation

- Two questions in 10 minutes, 1 pt.
 - Make sure the email address associated with your Blackboard account is correct
 - Check the announcements weekly
 - Submit it **before** the due date



Practice Quiz

- Four questions in 20 minutes, 1 pt.
 - 0.25 pt. for each question
 - Used as demo purpose
 - Practice Quiz has **one** attempt
 - Submit it **before** the due date

Discussion Board:

DB 0: Start Here

[DB 0: Start Here](#)

Instruction:

- 1) State your desktop/computer **software package** you've used/installed such as Atom. Or share the web link if you are using Python online (0.4 pt).
 - 2) State the difference between **compiler** and **interpreter** (0.4 pt).
 - 3) Define **OOP** (0.4 pt).
 - 4) After posting your solution, check each other's answers. Please ask questions and/or make at least one **comment** to one of the other responses (0.3 pt).
- Submit your posts before the due date.

Forum: DB 0: Start Here

In a thread, you can view the post and information

[Create Thread](#)

[Grading Information](#)

MESSAGE

* Subject

John Smith Starts Here

Message

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

B	<i>I</i>	<u>U</u>	S	Paragraph	▼	Arial
↶↷	↶	—	⊕	ABC ▼	✓	¶ " Ω ☺

Look Ahead (1 of 2):

Quiz 1

- Quiz 1A has 10 questions in 15 minutes, 10 pts, due 11:59 PM on 9/7
 - 10 multiple choice/true or false questions, 1 pt. for each question
 - Quiz 1A has *two* attempt, the *higher* grade will be selected
 - Submit Quiz 1A (at least 1-minute) **before** the due time to Blackboard
- Quiz 1B has 2 code questions, 15 pts
 - Write the Python code based on the given question
 - Each question will be given during the first 10-minute of each session of week 1
 - Quiz 1B-1 on session A, 9/1 and Quiz 1B-2 on session B, 9/6
 - Quiz 1B has *one* attempt

Look Ahead (2 of 2): DB 1

PE1_6

PE1 Variables**Page 2 of 2**

6. 1 - 12 evaluate the numeric expression without the computer, and then use Python to check your answer.

1) $2+3*4$	2) $1-7**2$	3) $1//2**3$
4) $(3+4)*5$	5) $(5\%3)*4$	6) $(-2)**(-2)$
7) $7//3$	8) $14\%4$	9) $1+7\%4$
10) $14//4*4$	11) $5//2+2$	12) $5\%5*5$

13 - 18 determine whether the name is a valid variable name. Explain your answer.

13) NewYear.sales	14) room&color	15) TOrF_1040
16) 311HotLine	17) expense#	18) INCOME 101

19 - 24 rewrite the statements using augmented assignment operators.

19) $cost = cost + 5$	20) $sum = sum * rate$	21) $product = product / 10$
22) $cost = cost // num$	23) $total = total - cost$	24) $quotient = quotient \% rate$

25 - 27 find the value of the function where $a = 5$ and $b = 3$.

25) $int(-a / b)$	26) $round(a / b, 2)$	27) $abs(b - a)$
-------------------	-----------------------	------------------

Make your post

- Instruction:
 - 1) Choose any **three** question (1 - 27) to complete from PE1_6. Please **avoid** answering the same questions. Make sure to indicate the **question #** you're working on in the thread title as soon as you open your thread. Then you can **write your answer and explain your solution**. Check the class recording for more details (1.2 pt).
 - 2) After posting your solution, check each other's answers. Please ask questions or make comments (0.3 pt).
 - 3) Submit your posts before the due date. Let's learn from each other.