Computer Programming in Python

Chapter 2

The Python Shell and IDLE

Chapter 2 Python

- The Python Language
  - Python is a high-level, general-purpose language
  - Created by Guido van Rossum and introduced in 1991
  - Emphasizes code readability and is similar in many respects to pseudocode
  - The name Python comes from the famous British comedy Monty Python's Flying Circus

Chapter 2 Python

- The Python Language
  - An ***interpretive language***
    - Uses an interpreter to translate the code one line at a time and execute it
    - Interpretive languages tend to be slower than direct native machine code, and can be reverse engineered more easily, so their use should be restricted to areas where this is not an issue

Chapter 2 Python

- The Python Language
  - Supports procedural programming and object oriented programming
    - More on this later
  - Simple yet strong language with many supporting ***libraries*** (code written by others that we can use) as well as a standard library containing extensive capability

Chapter 2 Python

- The Python Language
  - Open Source Software
    - Software with source code that anyone can inspect, modify, and enhance
  - Developed under an OSI-approved open source license
    - Makes it free to use and distribute including for commercial use

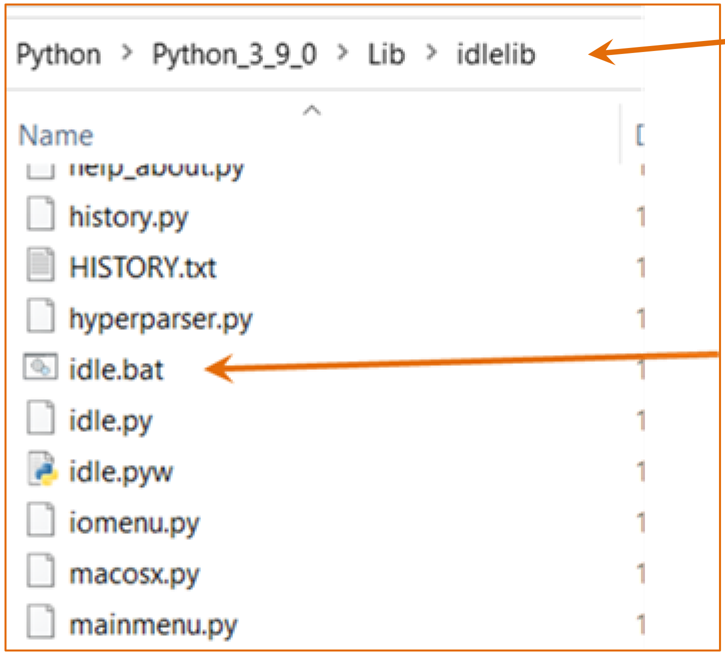
Available free to download and use from Python.org

Chapter 2 Python

- Getting and Installing Python
  - The website URL and steps to obtain and install Python are provided in Appendix B
    - Python should be installed before continuing in this chapter
  - The installation for Python includes:
    - The interpreter for executing Python code
    - The IDLE Integrated Development Environment for developing programs
    - Many libraries that provide functionality without having to write the code

Chapter 2 Python

- Getting and Installing Python
  - Once Python is installed, IDLE is launched using the idle.bat located in the Lib/idlelib directory



Python > Python_3_9_0 > Lib > idlelib

Name

help_about.py
history.py
HISTORY.txt
hyperparser.py
idle.bat
idle.py
idle.pyw
iomenu.py
macosx.py
mainmenu.py

Pat
h

Batch File

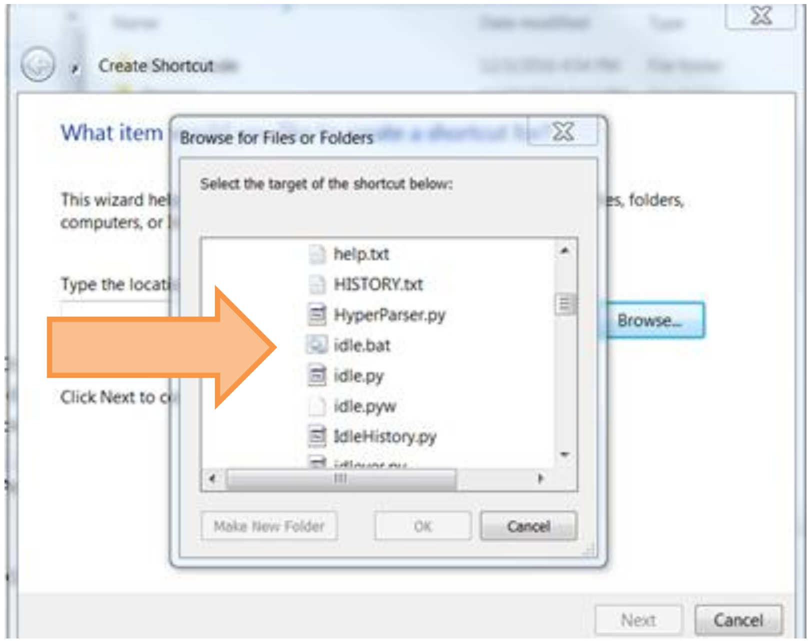
Chapter 2 Python

- Getting and Installing Python
  - A short-cut can be created at a higher level to eliminate drilling down into the sub-directory each time IDLE is launched
    - Right mouse click in the directory location for the short-cut
    - Select -> NEW -> Shortcut
    - Click on the “Browse” button and browse to the idle.bat file
    - Select idle.bat and click “Next”
    - Name the shortcut and click “Finish”

To see file extensions like “bat”, click the View tab on the window, and check the box for “File Name Extensions”

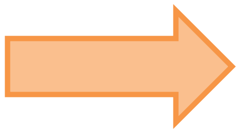
Chapter 2 Python

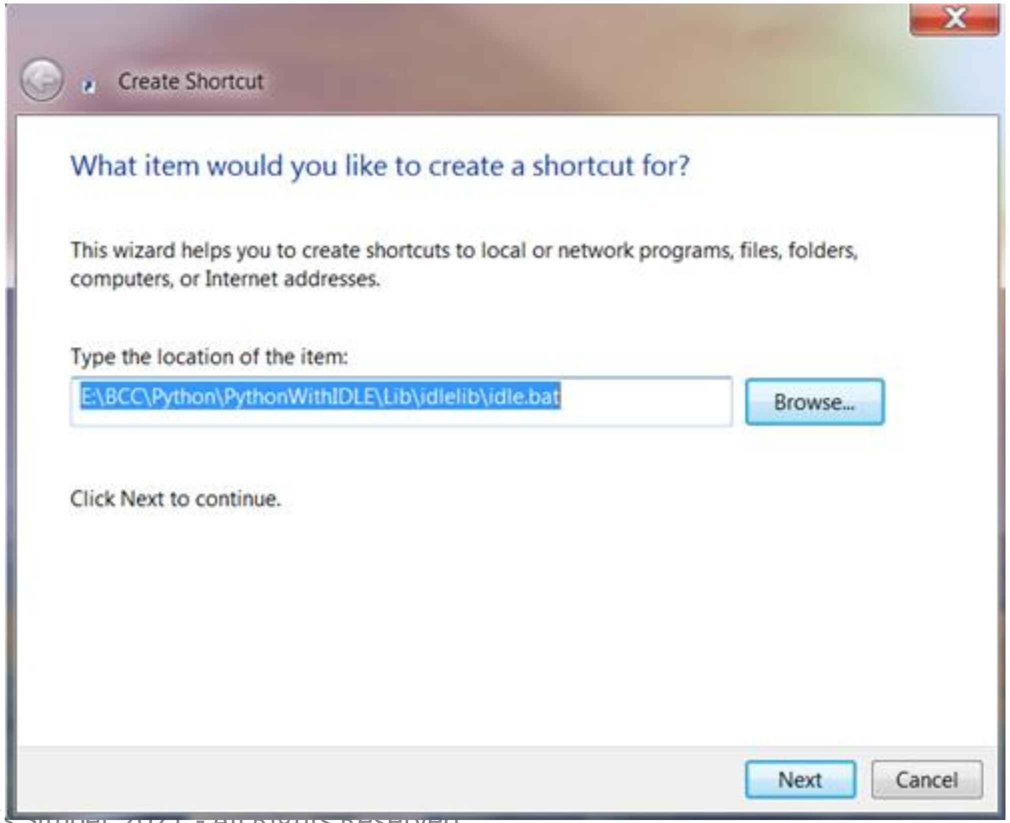
- Right mouse click in the directory location for the short-cut
- Select -> NEW -> Shortcut
- Click on the “Browse” button and browse to the idle.bat file
- Select idle.bat and click “Next”
- Name the shortcut and click “Finish”



Chapter 2 Python

- The complete path to the idle.bat should be in the create shortcut location
- Click “Next”





Create Shortcut

What item would you like to create a shortcut for?

This wizard helps you to create shortcuts to local or network programs, files, folders, computers, or Internet addresses.

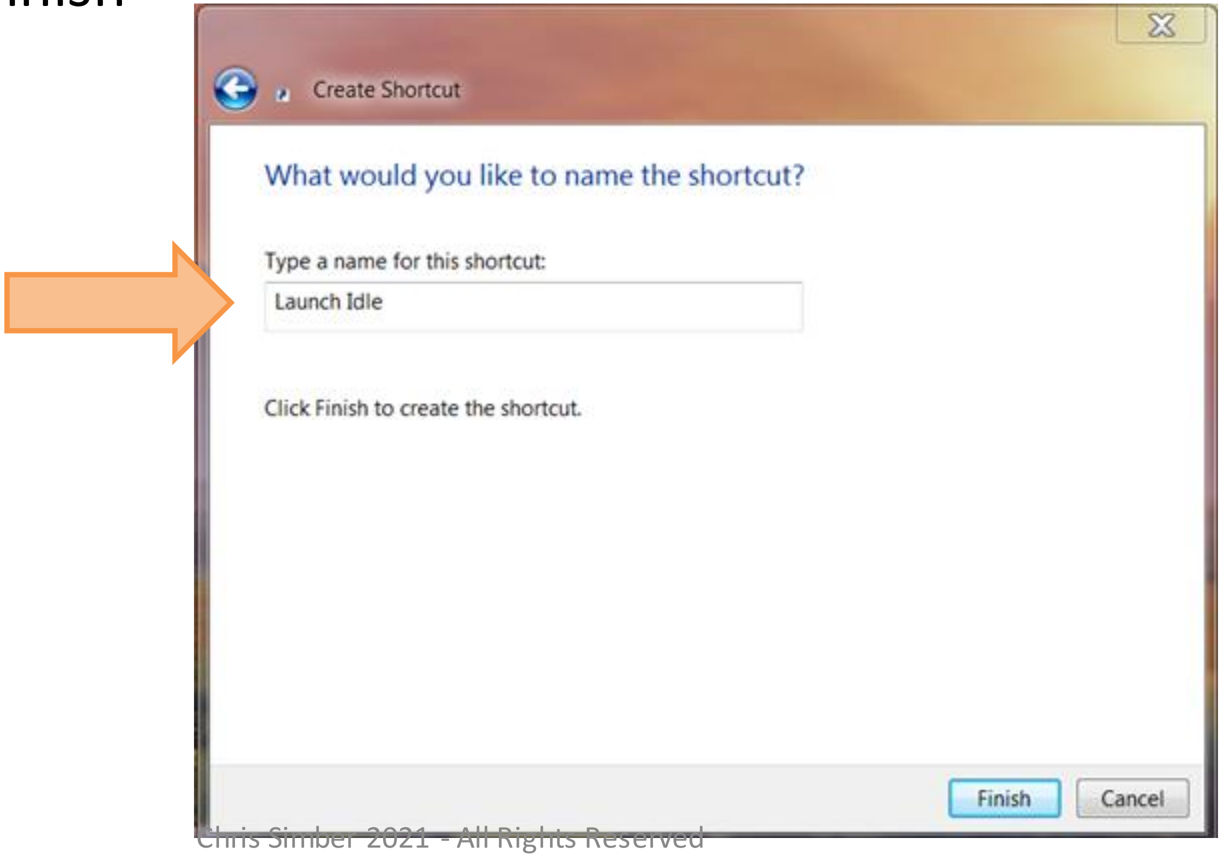
Type the location of the item:

```
E:\BCC\Python\PythonWithIDLE\Lib\idlelib\idle.bat
```

Click Next to continue.

Chapter 2 Python

- Now give the shortcut a **name** – shown here as *Launch Idle*
- Then click “Finish”



What would you like to name the shortcut?

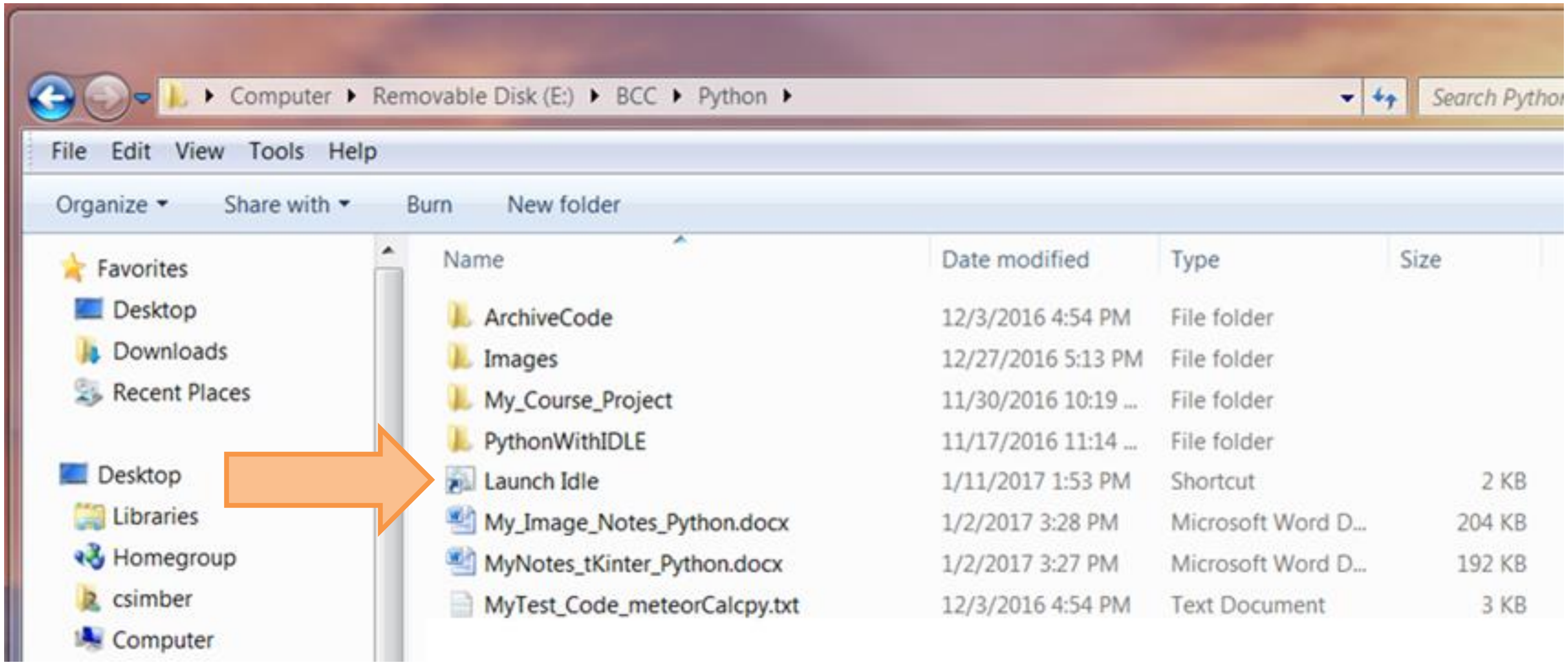
Type a name for this shortcut:

Launch Idle

Click Finish to create the shortcut.

Chapter 2 Python

- The shortcut to the idle.bat file should be in your working directory (shown here as "Launch Idle")

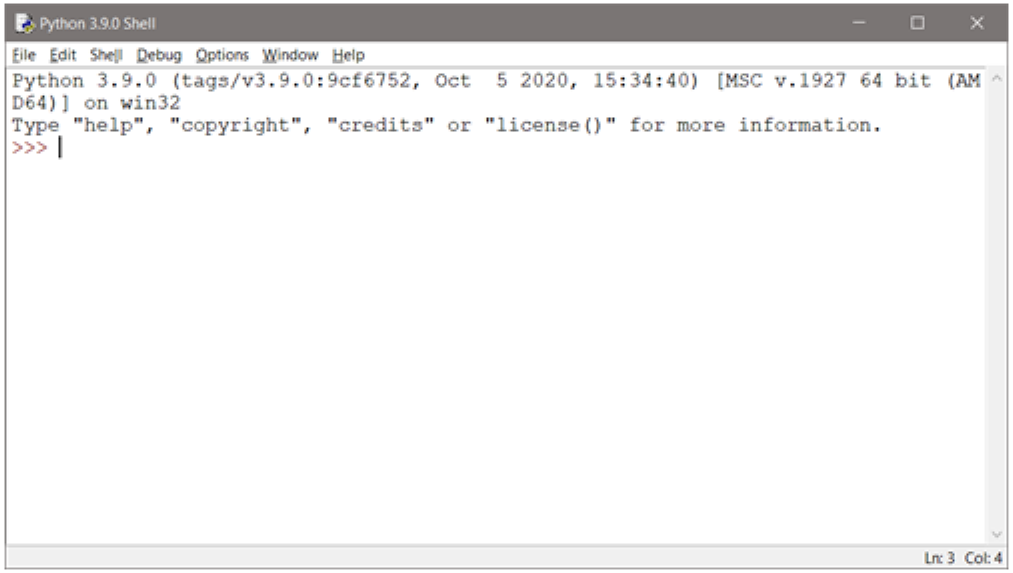


Computer ▸ Removable Disk (E:) ▸ BCC ▸ Python ▸
Search Python
File Edit View Tools Help
Organize ▾ Share with ▾ Burn New folder
Favorites
Desktop
Downloads
Recent Places
Desktop
Libraries
Homegroup
csimber
Computer

Name	Date modified	Type	Size
ArchiveCode	12/3/2016 4:54 PM	File folder	
Images	12/27/2016 5:13 PM	File folder	
My_Course_Project	11/30/2016 10:19 ...	File folder	
PythonWithIDLE	11/17/2016 11:14 ...	File folder	
Launch Idle	1/11/2017 1:53 PM	Shortcut	2 KB
My_Image_Notes_Python.docx	1/2/2017 3:28 PM	Microsoft Word D...	204 KB
MyNotes_tKinter_Python.docx	1/2/2017 3:27 PM	Microsoft Word D...	192 KB
MyTest_Code_meteorCalcpy.txt	12/3/2016 4:54 PM	Text Document	3 KB

Chapter 2 Python

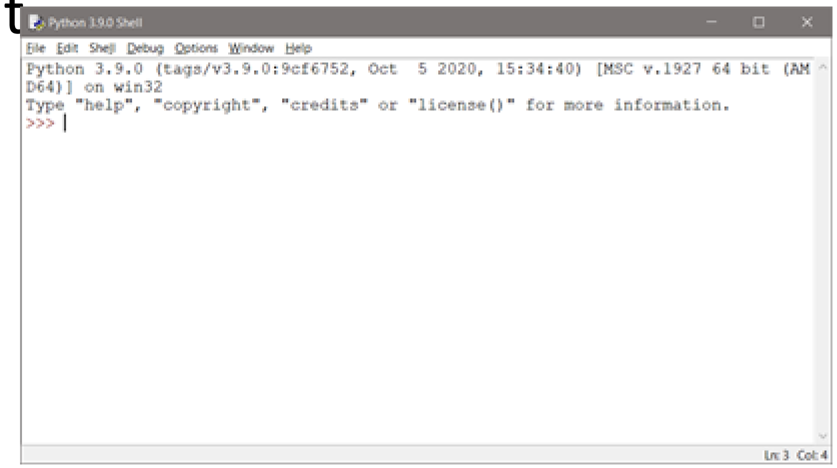
- The Python Shell
  - When double-clicked or accessed from the short-cut, idle.bat will open the Python ***Shell*** shown below
  - This is the Python interpreter in interactive mode



```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

Chapter 2 Python

- The Python Shell
  - Python statements can be run directly in the shell
  - Typing a single line in the shell and pressing *Enter* will execute the line using the interpreter
  - An easy way to demonstrate this is with a print statement



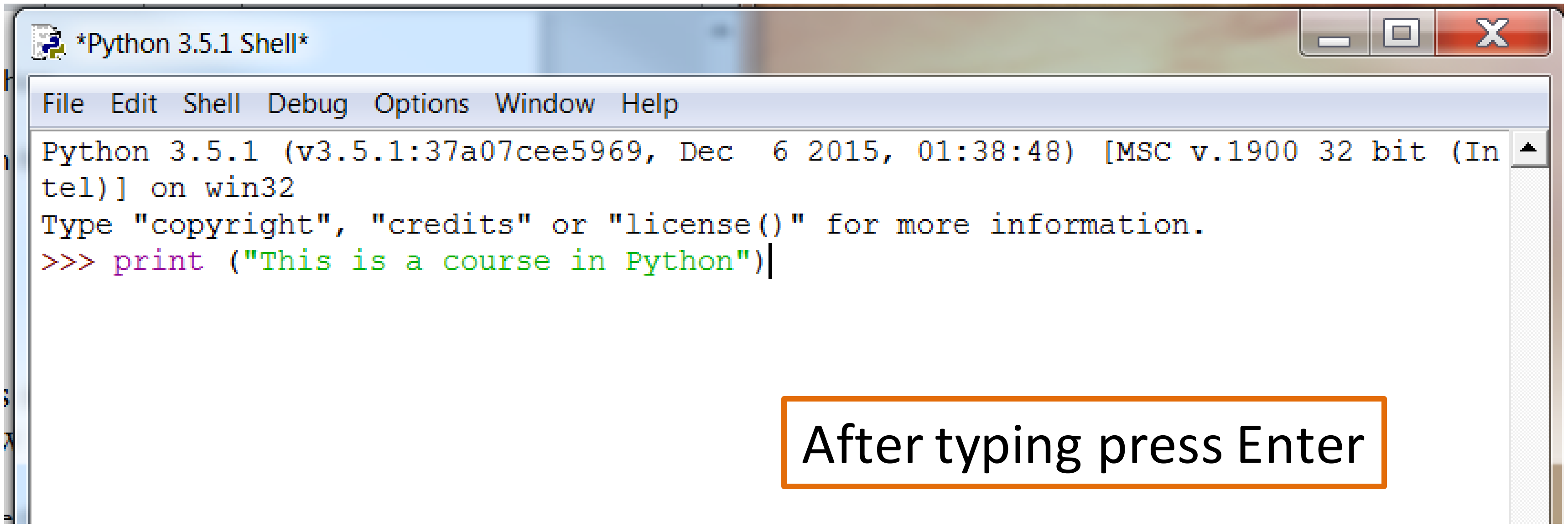
```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

Chapter 2 Python

- In the editor, after the “>>>” type:

  print (“This is a course in Python”)

- Notice the text color coding (more on this later)

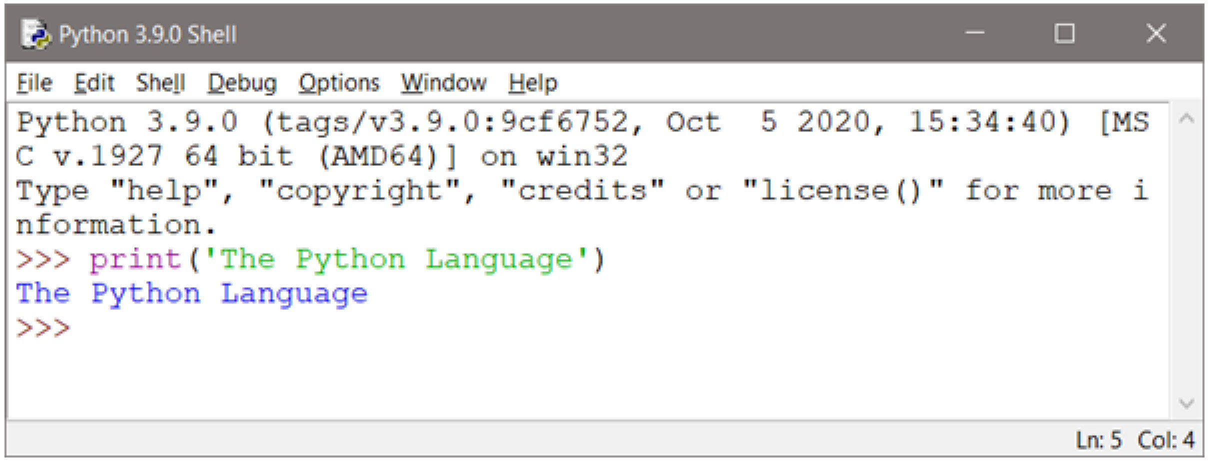


```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  6 2015, 01:38:48) [MSC v.1900 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print ("This is a course in Python")
```

After typing press Enter

Chapter 2 Python

- The Python Shell

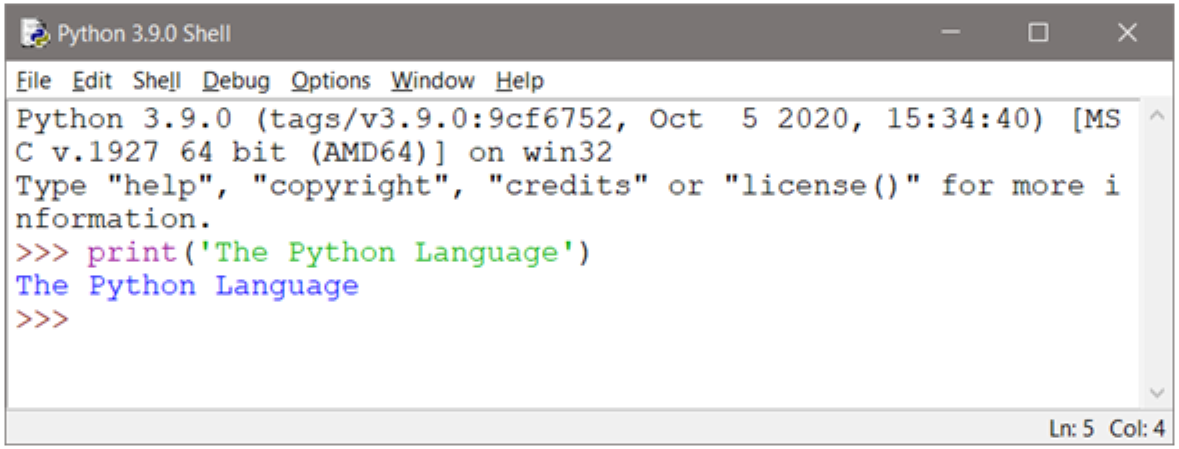


```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MS
C v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more i
nformation.
>>> print('The Python Language')
The Python Language
>>>
```

  – A line of code entered into the shell will execute when *Enter* is pressed
  – The prompt waiting for input in the shell is '>>>'

Chapter 2 Python

- The Python Shell

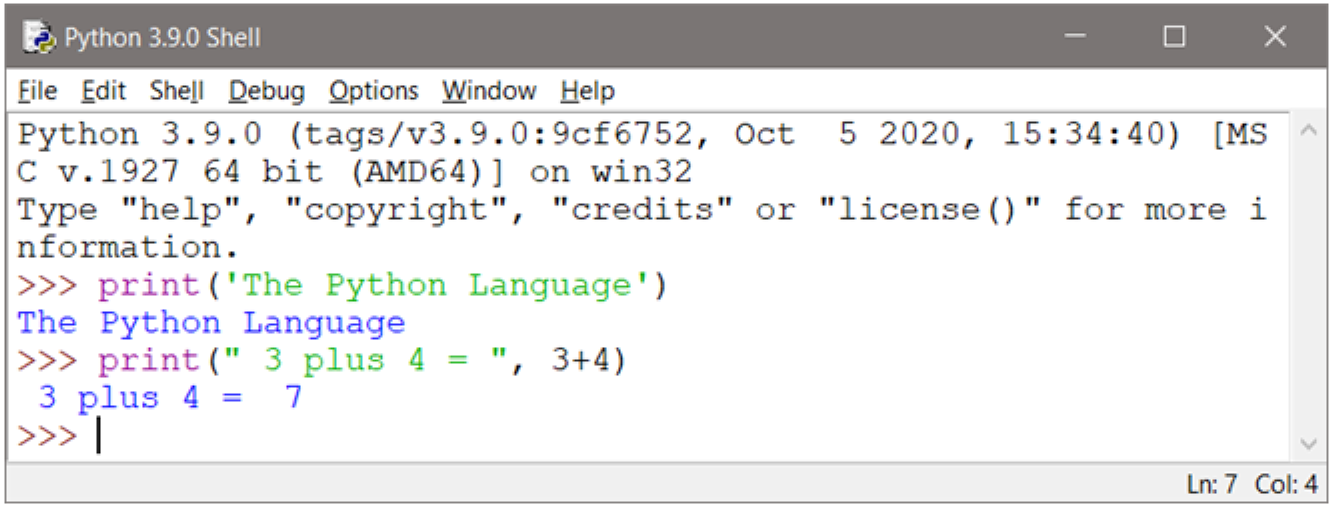


```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MS
C v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more i
nformation.
>>> print('The Python Language')
The Python Language
>>>
```

  - The words in the statements are color coded to highlight that they are different items
  - Also notice that the '>>>' prompt appears again after the output waiting for an additional statement to execute

Chapter 2 Python

- The Python Shell

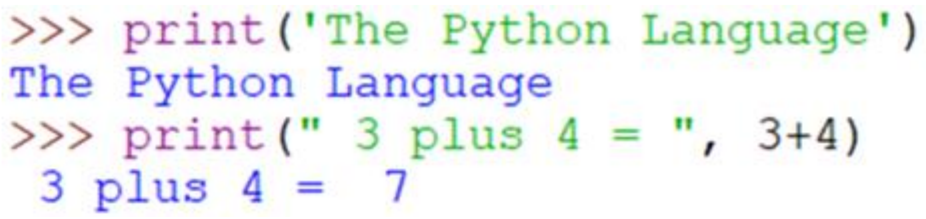


```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MS
C v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more i
nformation.
>>> print('The Python Language')
The Python Language
>>> print(" 3 plus 4 = ", 3+4)
 3 plus 4 =  7
>>> 
```

  – The line of code added here includes an equation
  – When *Enter* is pressed, the text in quotes and the result of the equation are displayed in the output

Chapter 2 Python

- The Python Shell



```
>>> print('The Python Language')
The Python Language
>>> print(" 3 plus 4 = ", 3+4)
 3 plus 4 =  7
```

  - The first print statement surrounds the text to output with single quotes
  - The second print statement uses double quotes
  - Python doesn’t care which is used, but it is important to be consistent
    - More on this later

Chapter 2 Python

- The Python Shell
  - Works well for one line of code, snippets, or examples
  - The goal is to write complete Python programs
  - The shell simply uses the interpreter to execute the line that was typed when *Enter* is pressed
  - Files will be used to write and execute more complex programs using the interpreter in ***script mode***
    - The interpreter will read the contents of a file to execute multiple lines of code (a program)

Chapter 2 Python

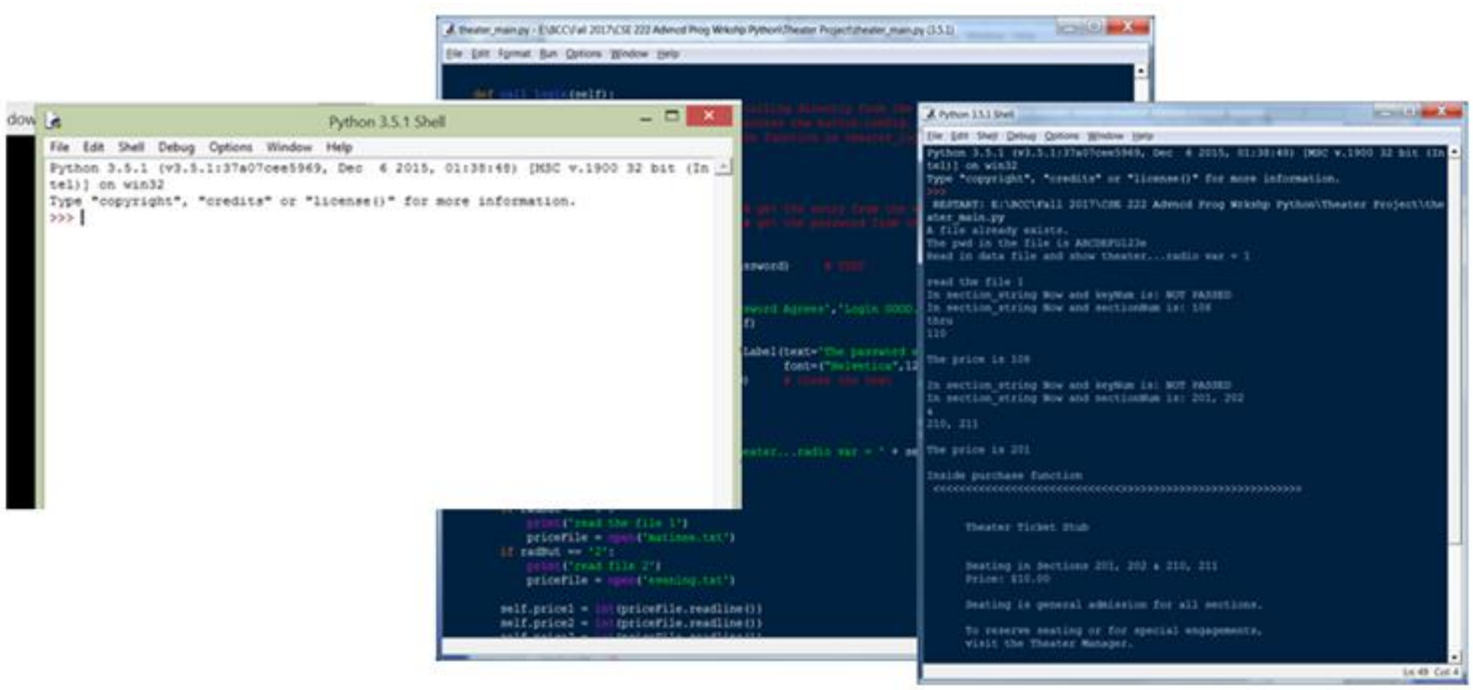
- The IDLE Development Environment
  - IDLE stands for "Integrated Development and Learning Environment"
  - Intended to be a simple ***Integrated Development Environment*** (IDE):
    - It is cross-platform (runs on multiple operating systems and computers)
    - It is suitable for starting out in Python especially in an educational setting

Chapter 2 Python

- The IDLE Development Environment
  - Provides a multi-window text editor and Python shell with syntax highlighting and smart indent
  - Features an integrated debugger with breakpoint capability and call stack visibility
  - It is automatically installed with Python, is free to use, and does not have the host of features that tend to clutter many IDEs with limited benefit

Chapter 2 Python

- The IDLE Development Environment

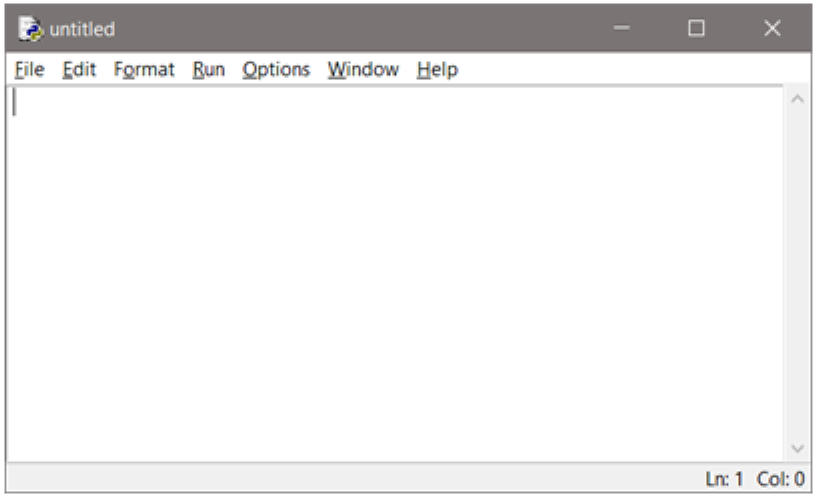

Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
theater_main.py - E:\BCC\Fall 2017\CSE 222 Advncd Prog Wrkshp Python\Theater Project\theater_main.py (3.5.1)
File Edit Format Run Options Window Help
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
Ln 49 Col 4

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  6 2015, 01:38:48) [MSC v.1900 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
```

```
    def call_login(self):
    ...
            print('read the file 1')
            priceFile = open('matinee.txt')
        if radBut == '2':
            print('read file 2')
            priceFile = open('evening.txt')

        self.price1 = int(priceFile.readline())
        self.price2 = int(priceFile.readline())
```

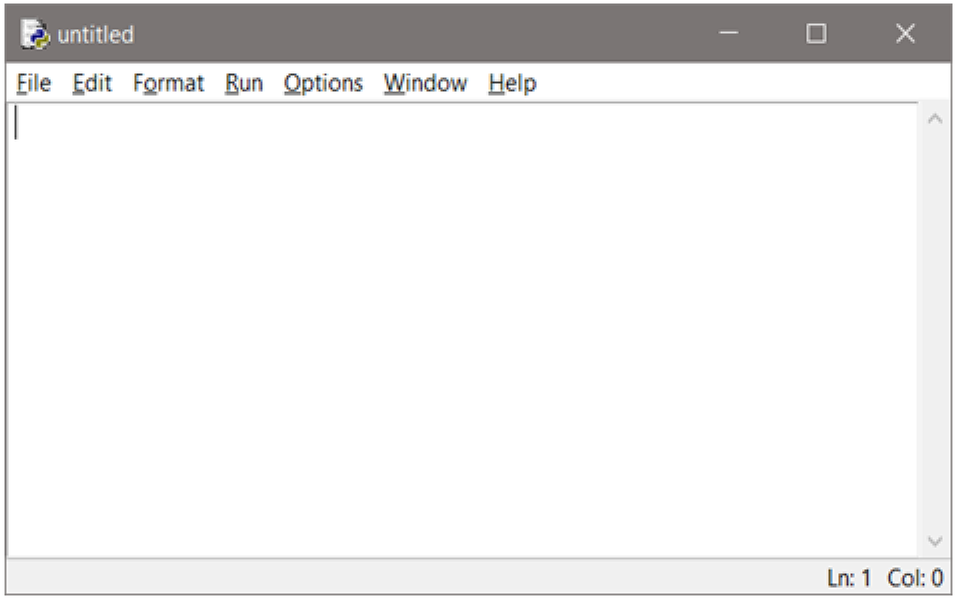
```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  6 2015, 01:38:48) [MSC v.1900 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: E:\BCC\Fall 2017\CSE 222 Advncd Prog Wrkshp Python\Theater Project\the
ater_main.py
A file already exists.
The pwd in the file is ABCDEFG123e
Read in data file and show theater...radio var = 1

read the file 1
In section_string Now and keyNum is: NOT PASSED
In section_string Now and sectionNum is: 108
thru
110

The price is 108

In section_string Now and keyNum is: NOT PASSED
In section_string Now and sectionNum is: 201, 202
&
210, 211

The price is 201

Inside purchase function
 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>


      Theater Ticket Stub


      Seating in Sections 201, 202 & 210, 211
      Price: $10.00

      Seating is general admission for all sections.

      To reserve seating or for special engagements,
      visit the Theater Manager.
```

- Starting the IDLE Text Editor
  - From the Python Shell menu, select File -> New File
  - The window that appears is the edit window where sequences of Python commands are written
  - The lines written in the editor will be executed as a group to form a program

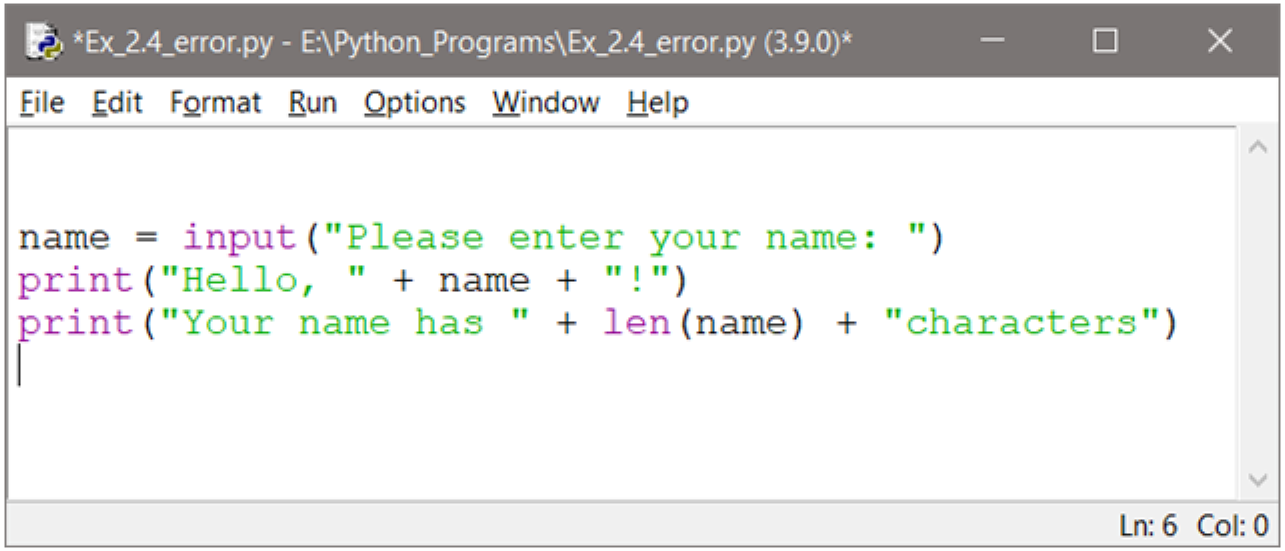


Chapter 2 Python

- Starting the IDLE Text Editor
  - Notice that the file name is currently "*Untitled*"
  - It will be named later before running the program

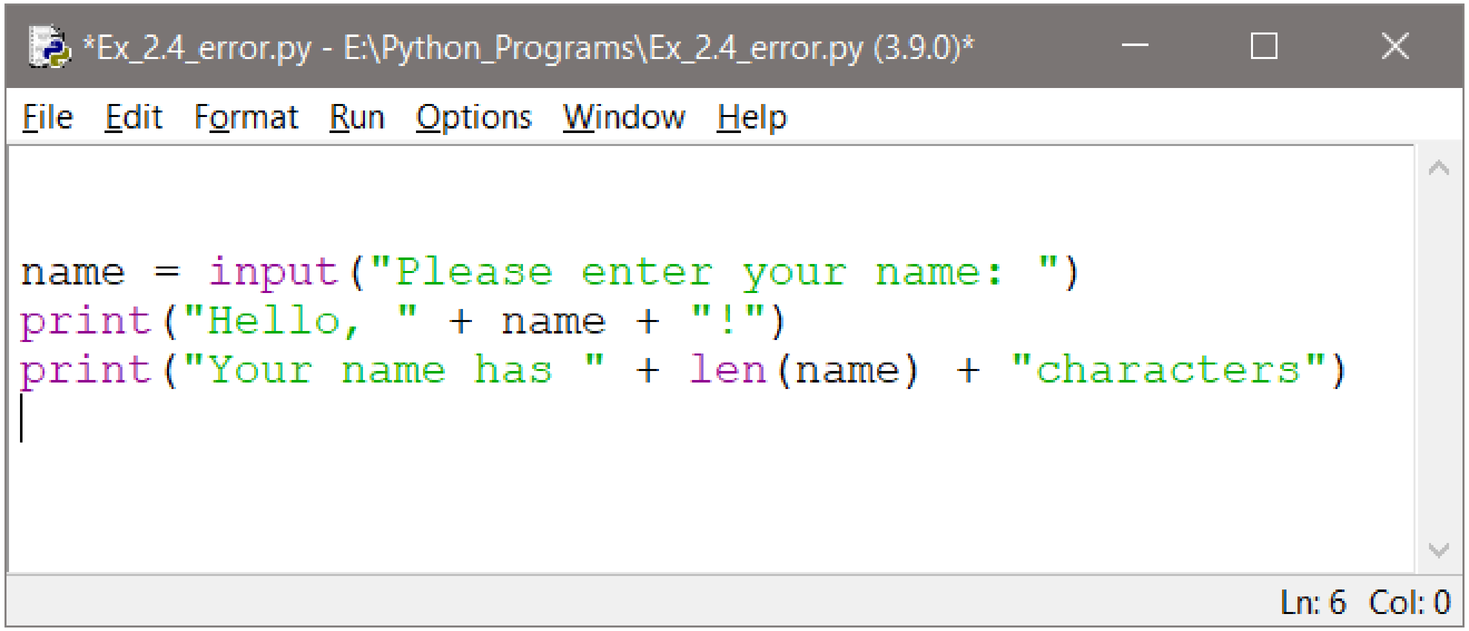


- Working in the IDLE Text Editor
  - The lines of code in the edit window below have an intentional error
  - Errors in programming are called ***bugs*** and are removed by "debugging" the program


Ex_2.4_error.py - E:\Python_Programs\Ex_2.4_error.py (3.9.0)
File Edit Format Run Options Window Help
Ln: 6 Col: 0

```python


name = input("Please enter your name: ")
print("Hello, " + name + "!")
print("Your name has " + len(name) + "characters")
```

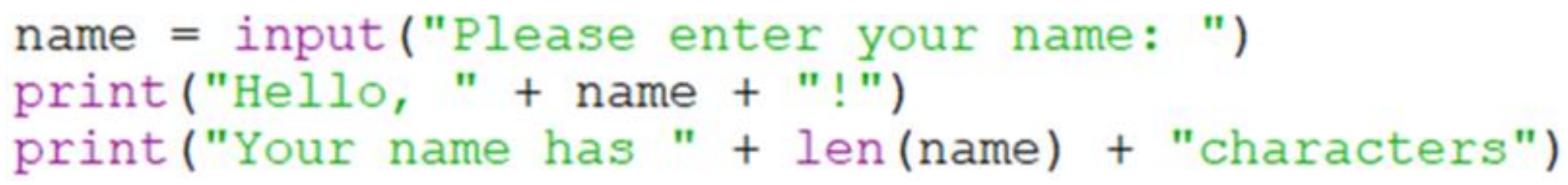
- Working in the IDLE Text Editor
  - Running this will show one way that IDLE indicates errors


Ex_2.4_error.py - E:\Python_Programs\Ex_2.4_error.py (3.9.0)
File Edit Format Run Options Window Help
Ln: 6 Col: 0

```python
name = input("Please enter your name: ")
print("Hello, " + name + "!")
print("Your name has " + len(name) + "characters")
```

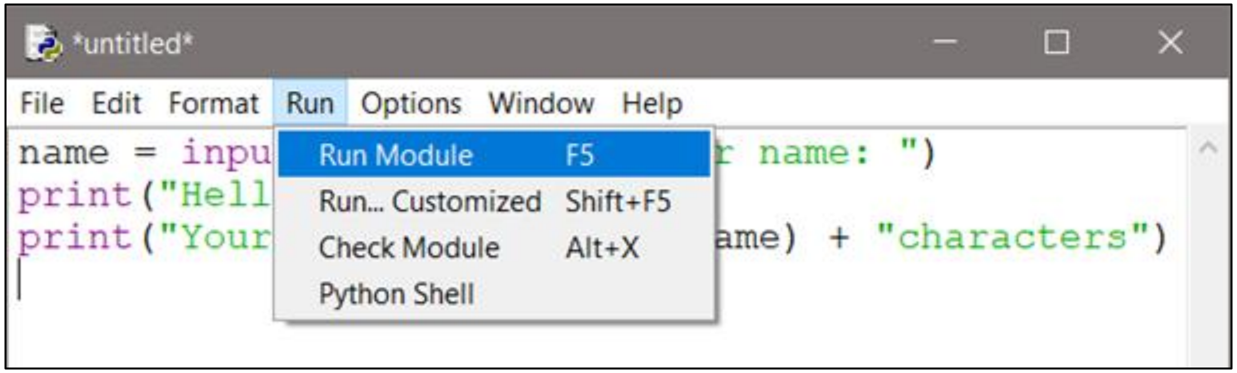
Chapter 2 Python

- Double-check that the three lines are exactly as shown
  - If something was mistyped, a second error may have been created



```
name = input("Please enter your name: ")
print("Hello, " + name + "!")
print("Your name has " + len(name) + "characters")
```

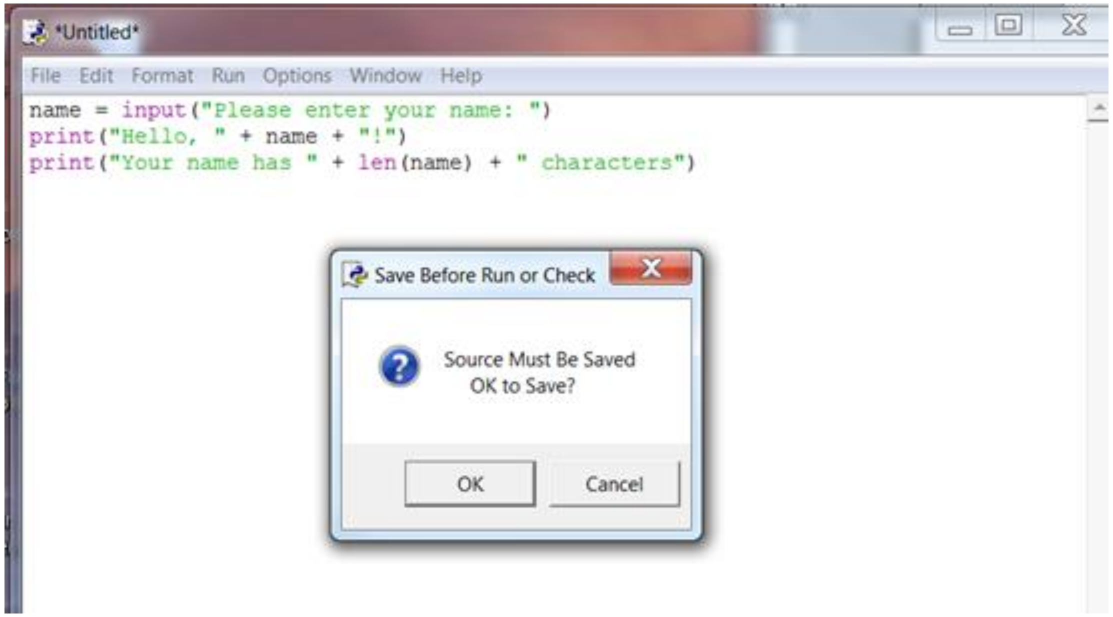
- Working in the IDLE Text Editor
  - To run the program using the text editor window
    - Press ‘**F5**’ on the keyboard, **OR**
    - Select ***Run*** and then ***Run Module*** from the menu


untitled
File Edit Format Run Options Window Help
Run Module F5
Run... Customized Shift+F5
Check Module Alt+X
Python Shell

```
name = inpu          r name: ")
print("Hell
print("Your          ame) + "characters")
```

Chapter 2 Python

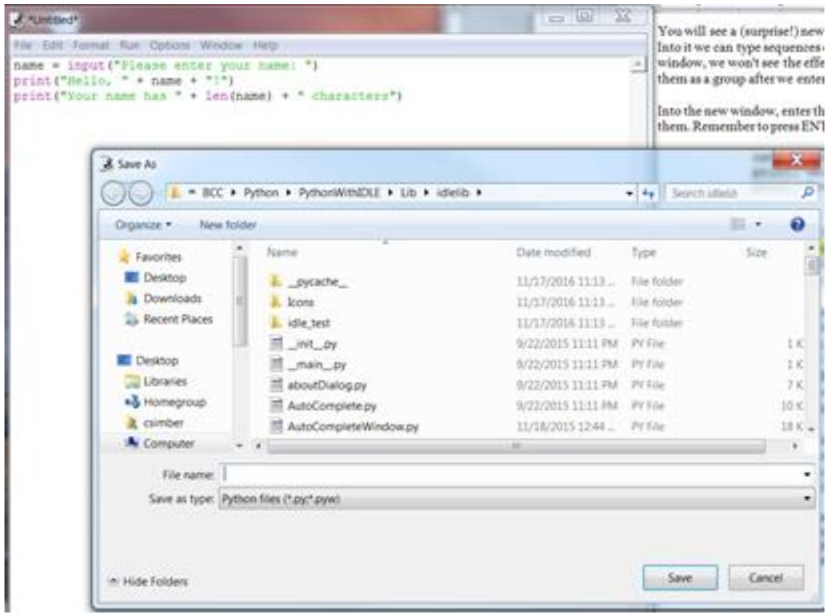
- Working in the IDLE Text Editor
  - Immediately, IDLE will pop up a little window that says, "Source Must Be Saved. OK to Save?“
    - Click the “OK” button



```
name = input("Please enter your name: ")
print("Hello, " + name + "!")
print("Your name has " + len(name) + " characters")
```

Source Must Be Saved
OK to Save?

- Working in the IDLE Text Editor
  - IDLE will force saving the file before running the program
    - Choose a name and the ".py" file extension will be added
    - After the file has been saved, Python will run the program



```
name = input("Please enter your name: ")
print("Hello, " + name + "!")
print("Your name has " + len(name) + " characters")
```

You will see a (surprise!) new
Into it we can type sequences
window, we won't see the effe
them as a group after we ente

Into the new window, enter th
them. Remember to press ENT

Save As

BCC ▸ Python ▸ PythonWithIDLE ▸ Lib ▸ idlelib ▸

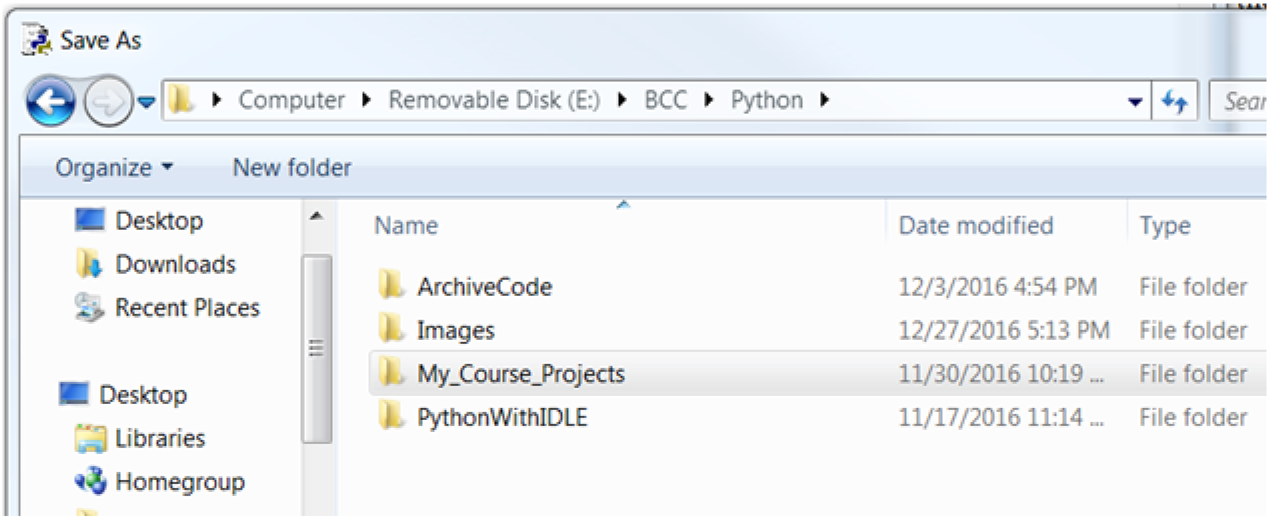
Name	Date modified	Type	Size
__pycache__	11/17/2016 11:13 ...	File folder	
Icons	11/17/2016 11:13 ...	File folder	
idle_test	11/17/2016 11:13 ...	File folder	
__init__.py	9/22/2015 11:11 PM	PY File	1 K
__main__.py	9/22/2015 11:11 PM	PY File	1 K
aboutDialog.py	9/22/2015 11:11 PM	PY File	7 K
AutoComplete.py	9/22/2015 11:11 PM	PY File	10 K
AutoCompleteWindow.py	11/18/2015 12:44 ...	PY File	18 K

File name:
Save as type: Python files (*.py;*.pyw)

Chapter 2 Python

- Working in the IDLE Text Editor
  - There is where a lot of students go down an ever more painful path
  - Developing the habit of using an organized file/folder structure saves a tremendous amount of time and frustration
  - Operating Systems fight this good habit through default folder locations that are file type dependent i.e. when we save a document, the OS defaults to “My Documents” directory
    - This couldn’t be more wrong!
    - Files should be organized

- Working in the IDLE Text Editor
  - Since we can save our *.py files anywhere and still run them, we are able to create a working folder for development and sub-folders for individual projects.
  - Create a folder for your work in this course, and then a sub-folder for this example.


Save As

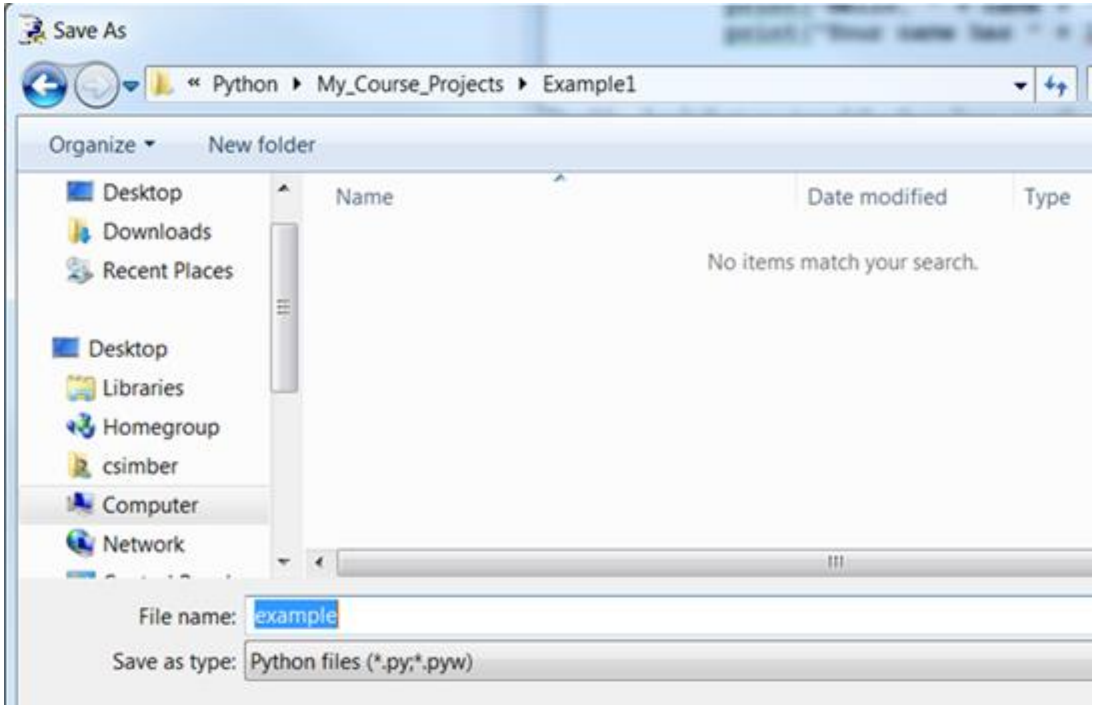
Computer ▸ Removable Disk (E:) ▸ BCC ▸ Python ▸

Organize ▾ New folder

Name	Date modified	Type
ArchiveCode	12/3/2016 4:54 PM	File folder
Images	12/27/2016 5:13 PM	File folder
My_Course_Projects	11/30/2016 10:19 ...	File folder
PythonWithIDLE	11/17/2016 11:14 ...	File folder

Chapter 2 Python

- Working in the IDLE Text Editor
  - Now to name the file (example)…**do not add the .py**
  - Notice the directory: Python\My_Course_Projects\Example1



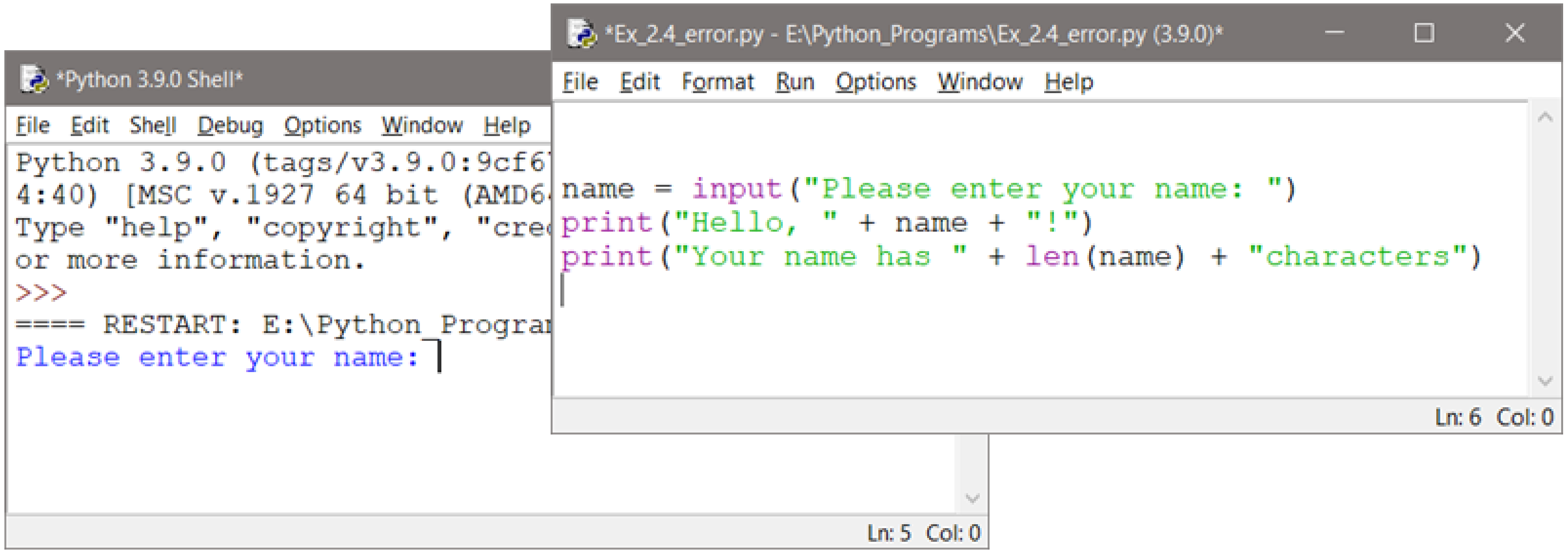
No items match your search.

File name: example

Save as type: Python files (*.py;*.pyw)

Chapter 2 Python

- Working in the IDLE Text Editor
  - Give it a name like “*example*” and click "Save“
  - Immediately, Python will start running your program
    - Look back at the Python Shell window to interact with it.

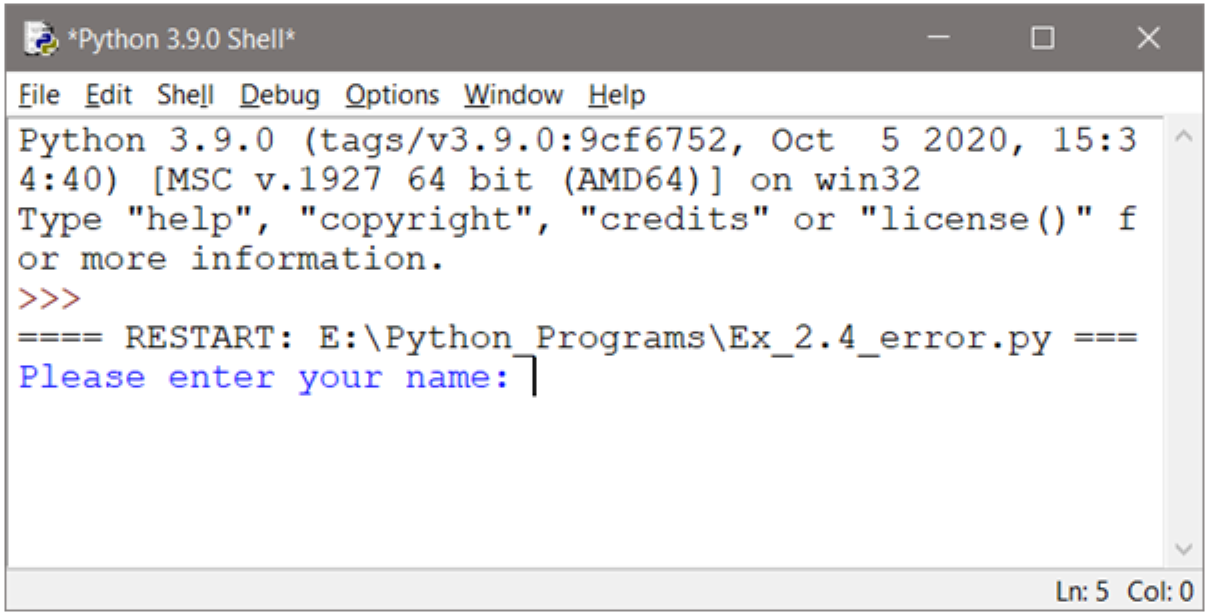

Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Ln: 5 Col: 0
Ex_2.4_error.py - E:\Python_Programs\Ex_2.4_error.py (3.9.0)
File Edit Format Run Options Window Help
Ln: 6 Col: 0

```
Python 3.9.0 (tags/v3.9.0:9cf6
4:40) [MSC v.1927 64 bit (AMD6
Type "help", "copyright", "cre
or more information.
>>>
==== RESTART: E:\Python_Prograr
Please enter your name:
```

```
name = input("Please enter your name: ")
print("Hello, " + name + "!")
print("Your name has " + len(name) + "characters")
```

Chapter 2 Python

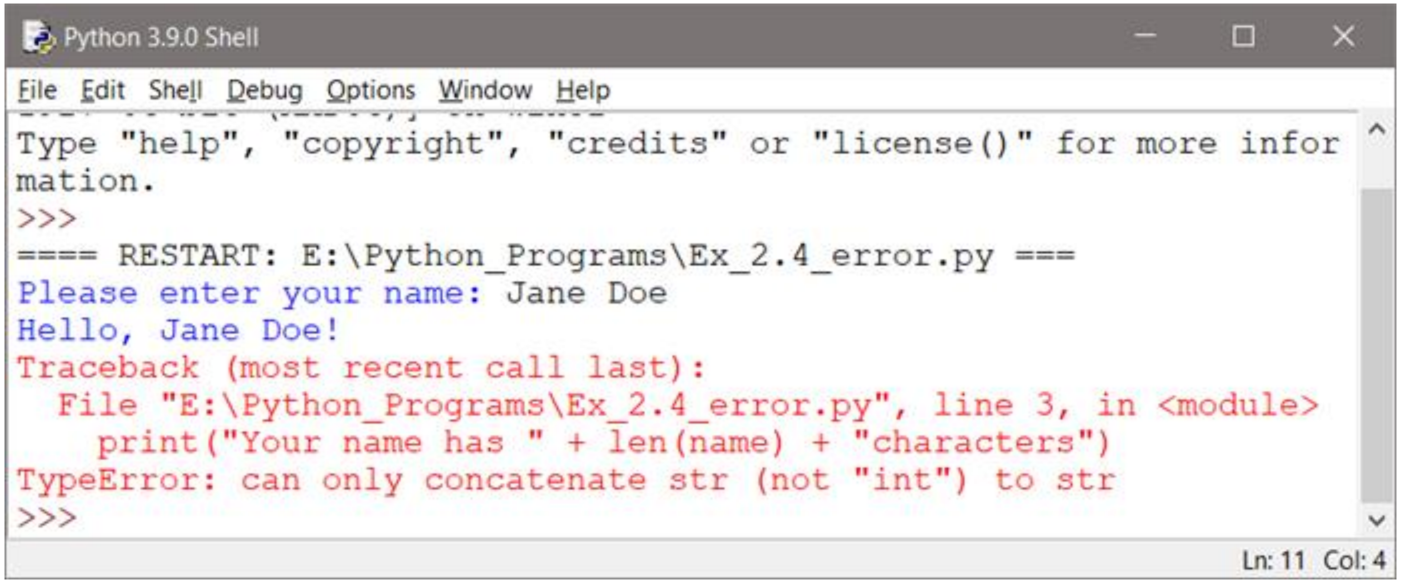
- Working in the IDLE Text Editor
  - When the program runs, the prompt to “Please enter your name” will appear in the Python shell
  - This is where input will be entered (at least for now)



```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:3
4:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" f
or more information.
>>>
==== RESTART: E:\Python_Programs\Ex_2.4_error.py ===
Please enter your name: 
```

Chapter 2 Python

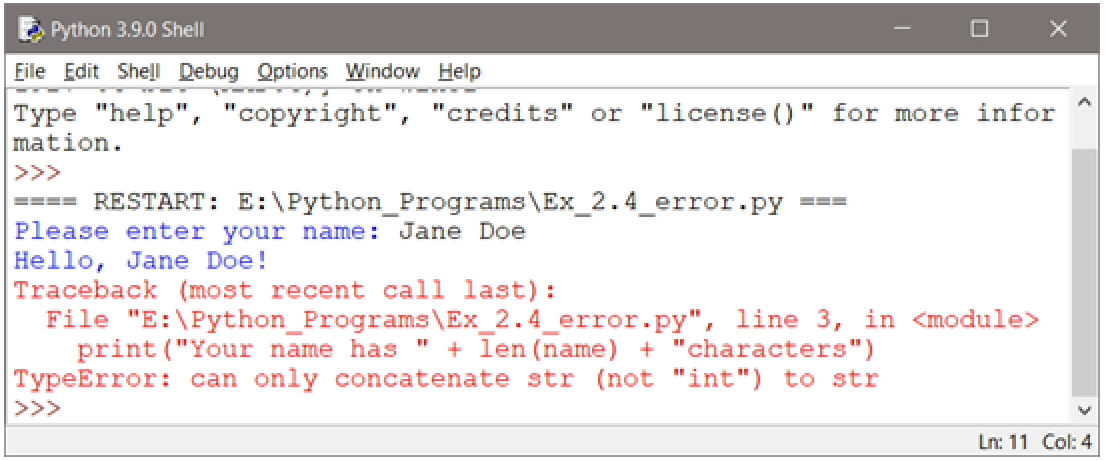
- Working in the IDLE Text Editor
  - When a response to the prompt is typed and *Enter* is pressed, the intentional error in the code will surface
  - In the Python shell, the red text provides information about the error



```
Type "help", "copyright", "credits" or "license()" for more infor
mation.
>>>
==== RESTART: E:\Python_Programs\Ex_2.4_error.py ===
Please enter your name: Jane Doe
Hello, Jane Doe!
Traceback (most recent call last):
  File "E:\Python_Programs\Ex_2.4_error.py", line 3, in <module>
    print("Your name has " + len(name) + "characters")
TypeError: can only concatenate str (not "int") to str
>>>
```

Chapter 2 Python

- Working in the IDLE Text Editor

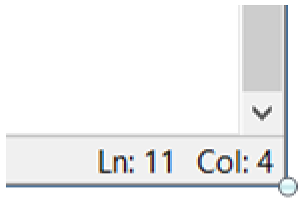


```
Type "help", "copyright", "credits" or "license()" for more infor
mation.
>>>
==== RESTART: E:\Python_Programs\Ex_2.4_error.py ===
Please enter your name: Jane Doe
Hello, Jane Doe!
Traceback (most recent call last):
  File "E:\Python_Programs\Ex_2.4_error.py", line 3, in <module>
    print("Your name has " + len(name) + "characters")
TypeError: can only concatenate str (not "int") to str
>>>
```

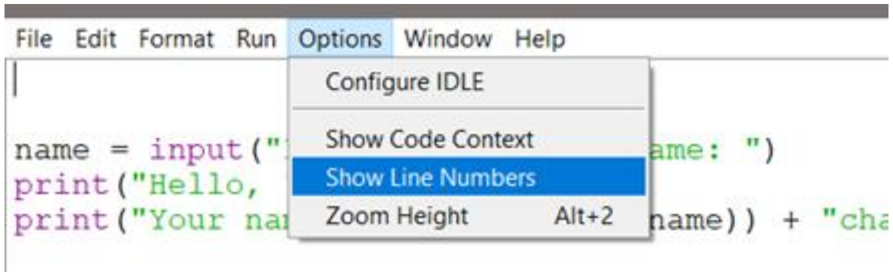
- The error information includes:
  - A “***Traceback***” of the function call or calls causing the error
  - The file name and line number for the error
  - The line of code itself
  - The type of error

Chapter 2 Python

- Working in the IDLE Text Editor
  - The line numbers for the program can be seen at the bottom right of the IDLE editor window

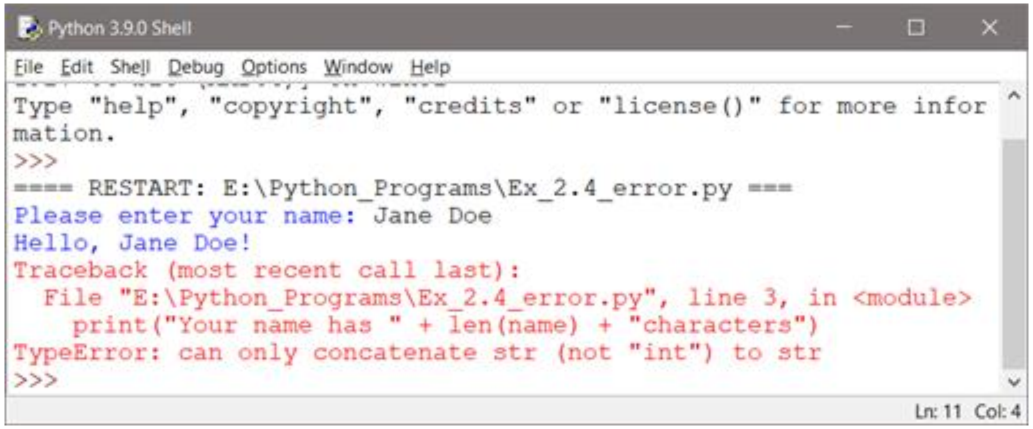


  - Also, selecting the Option menu item will display them along the margin



```
name = input("        ame: ")
print("Hello,
print("Your na        name)) + "cha
```

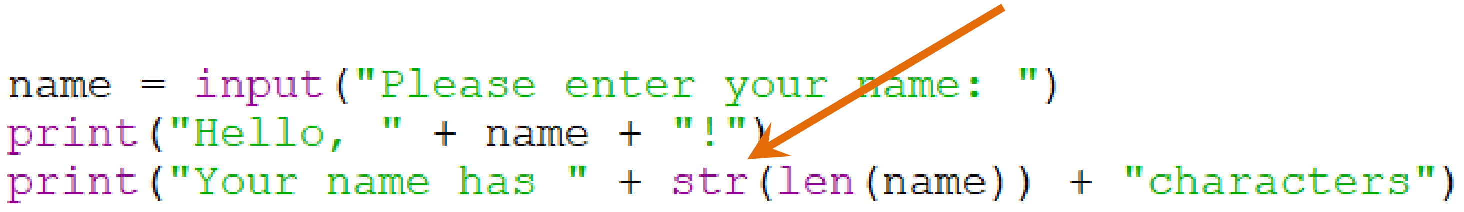
- Working in the IDLE Text Editor
  - The error is the result of the function call *len(*name*)* returning an integer which is the length of the string passed to it in name
  - Python cannot concatenate (join) an integer onto the string *"Your name has ",* so execution stops and the error message appears



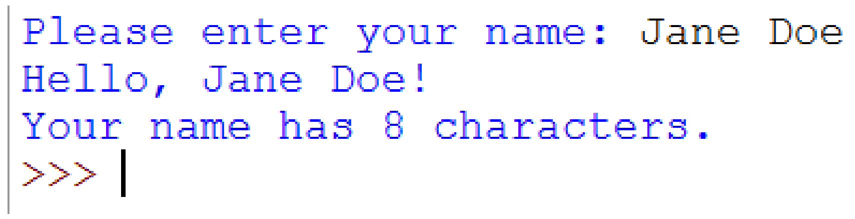
```
Type "help", "copyright", "credits" or "license()" for more infor
mation.
>>>
==== RESTART: E:\Python_Programs\Ex_2.4_error.py ===
Please enter your name: Jane Doe
Hello, Jane Doe!
Traceback (most recent call last):
  File "E:\Python_Programs\Ex_2.4_error.py", line 3, in <module>
    print("Your name has " + len(name) + "characters")
TypeError: can only concatenate str (not "int") to str
>>>
```

Chapter 2 Python

- Working in the IDLE Text Editor
  - To correct this, the return value from the function call to *len()* can be converted to a string using *str*
  - Python can then concatenate (join) the resulting string to the literal string of characters before it, and concatenate the literal string after it



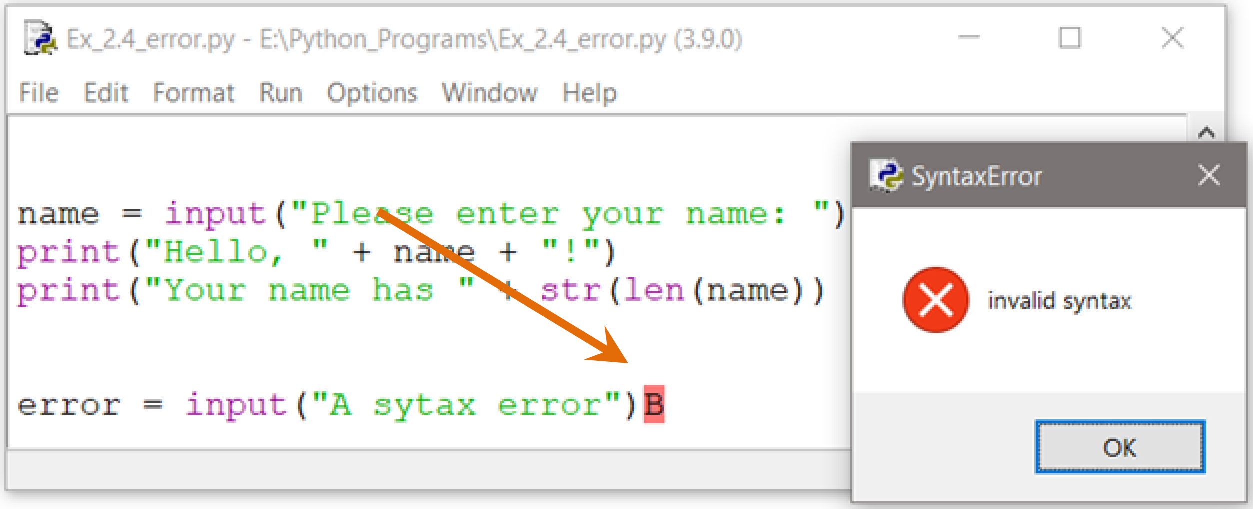
```
name = input("Please enter your name: ")
print("Hello, " + name + "!")
print("Your name has " + str(len(name)) + "characters")
```



```
Please enter your name: Jane Doe
Hello, Jane Doe!
Your name has 8 characters.
>>> |
```

Chapter 2 Python

- Working in the IDLE Text Editor
  - Another type of error is a syntax error
  - IDLE will highlight the actual code where the error occurs by highighting it in red and producing an error dialog
  - In the example below, the character "B" was erroneously typed at the end of the line


Ex_2.4_error.py - E:\Python_Programs\Ex_2.4_error.py (3.9.0)
File Edit Format Run Options Window Help
SyntaxError
OK

```
name = input("Please enter your name: ")
print("Hello, " + name + "!")
print("Your name has " + str(len(name))


error = input("A sytax error")B
```

invalid syntax

Chapter 2 Python

- Errors in Programming
  - ***Syntax error*** – incorrect use of language specific rules like indentation and punctuation which will be found by the interpreter and the code will not execute
  - ***Logic error*** – the program runs, but does not perform the task it was intended to perform or it produces incorrect results
  - ***Runtime error*** – logic error that causes the program to stop executing

Chapter 2 Python

- Errors in Programming
  - The most common errors in programming:
    - Misspelling words
    - Forgetting case sensitivity
    - Forgetting closing quotes
    - Forgetting closing parentheses

- Running IDLE
  - A quick way to run a python program - *Press F5 (Function Key)*
  - Recall that a quick way to save the file is Control-S
  - Combining these 2 quick commands while working in IDLE makes things much quicker and easier

  - *After writing some code or making a change*

Control-S	*Saves the code*
F5	*Runs the program*

Chapter 2 Python

- Exiting Python and IDLE
  - To leave IDLE, just close the windows
    - Since IDLE insists that files are saved before each execution, it's hard to lose changes when exiting IDLE
  - To be really safe, save the program manually before closing the editing window
  - Choose "File" on the menu bar and "Save" from the drop-down menu or use *Control-S*

Chapter 2 Python

Chapter 2 The Python Shell and IDLE