Introduction to Programming, Using Python Class Guide

A loose guide of class progression

Includes associated slides from PowerPoint(R) slide file:

Introduction to Programming Using Python.pptx

Class 1 - Introduction - Getting Going

## Class Introduction - Slides 1-6

### Instructor

### Objectives

### Online / Zoom

### Class Structure / Guideline

### What is Programming?

### Course Outline - Estimate

## Learn By Dong - Slides 10-15

* A Blazing Start to Python
* Get a simple feel to programming / Python
* Try lots of example – small changes

## IDLE (Integrated Development Learning Environment)

##### Run IDLE, creating a window

* Open/print file exercises/python\_introduction.txt so you can follow along, using IDLE attempt the exercise examples
* NOTE: You can use IDLE to open this file (File->Open->...find exercises/python\_introduction.py...)

The python\_indroduction.txt, opened in IDLE, if possible, should be in a separate window This file was written with the attempt you could follow it without much additional instruction. Don't worry about errors. Python/Idle is just trying to help by guessing the problem when it does not understand your input.

## sOME eXERCISES – fOLLOW THE iNSTRUCTOR

**>>>** 1 + 1

**2**

**# Beginning comments**

**1+1, \*, /, -, %, \*\*, //, 1+2;3+4, 2+3\*4,(2+3)\*4**

**"a", 'b', """c""", '''d''',**

**"Ray" + " " + "Smith"**

**print(1, "one", 2, "two")**

**var\_name = 1, a, b**

**NOT 1a, a b**

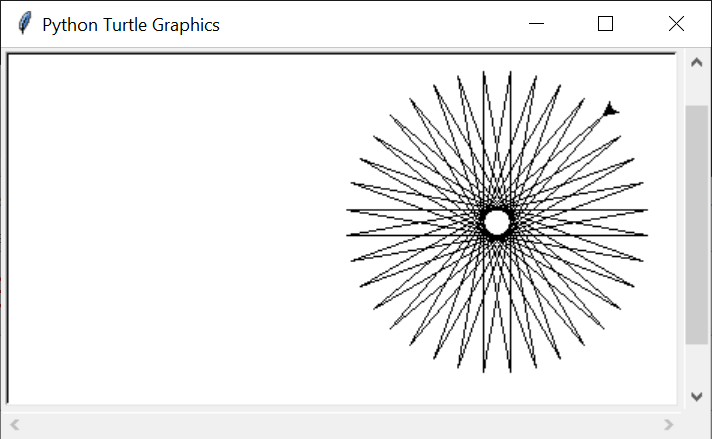
* Graphics

from turtle import \*

for i in range(50):

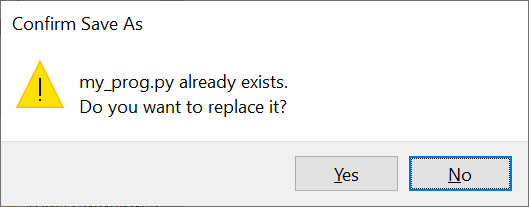
forward(200)

left(170)

**Produces:** 

## Creating a program file - Slides 16-19

* IDLE shell is great for short/simple/one-time examples
* IDLE shell is not so great for longer/complex/reuse
* Putting your Python code in a file is
  + Fairly easy
  + Supports reuse
  + Supports modification
* Create a folder, for example, class\_work for your class work (probably a sister folder in which you stored the exercises folder I sent out as part of the introduction to introduction...email.
* Using IDLE, create an empty "place-holder" file named "my\_prog.py" in your class\_work folder.
  + File 🡺 New File
  + In new untitled window: File🡪Save
  + In Save As window:
    - Navigate to your newly created "class\_work" folder
    - Click Save
      * NOTE If the following Message appears, the file already exists.





## hello\_world.py program – simple but Informative

* Create empty file
  + Simplest way (File-->Recent Files ... look and select my\_prog.py, your "place-holder")
* Save the file as hello\_world.py:
  + **File🡪Save As**
  + **File name:** hello\_world.py
  + **Save**

**REMEMBER** to Save As to **new** name **before** doing any work so as to reduce the possibility of destroying old work.

* Add hello\_world.py code to new file by typing the following:

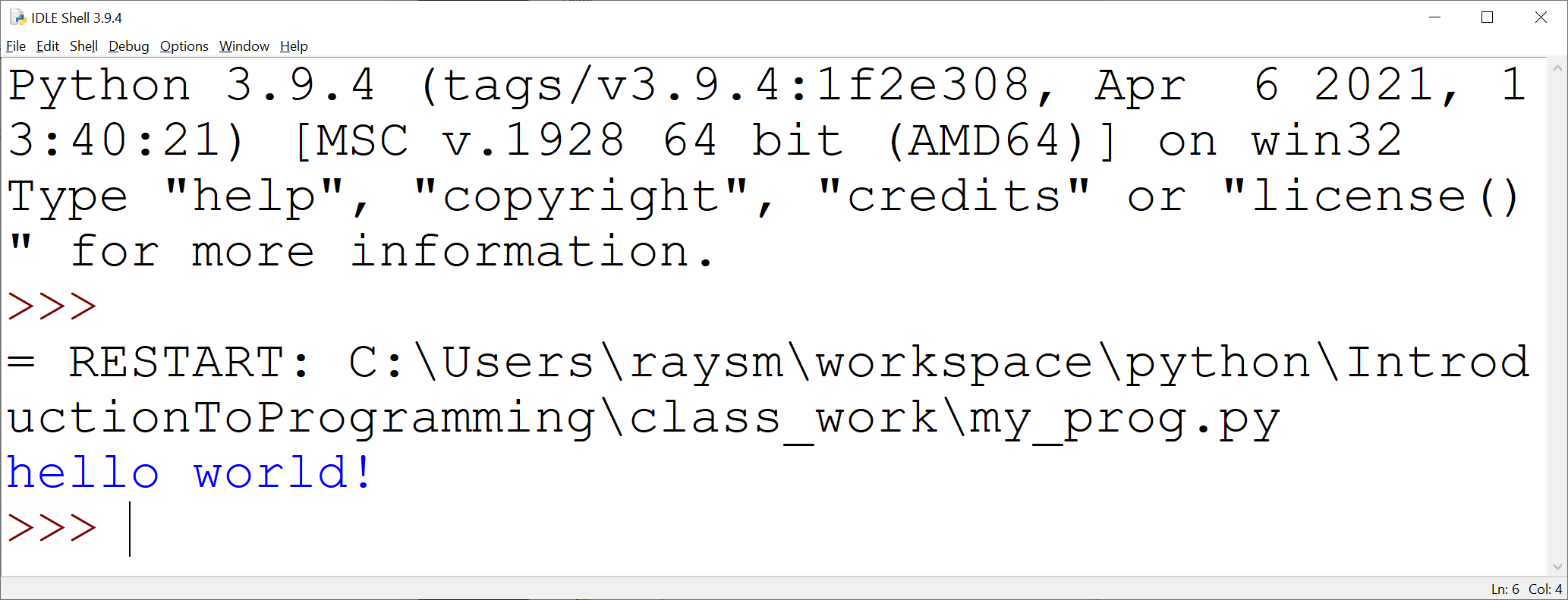
# hello\_world.py 19-July-2021 crs, create original

print("hello world!")

* Save program:
  + File 🡪 Save

Note: we have already named the file above

* Run program:
  + Run 🡪 Run Module



## create and test the goodbye\_world.py

## What should you use as your starting point ?

Remember, when climbing a mountain, it is often easier starting at base camp than starting at the bottom of the ocean. :)

## Programming Tools vs Real Life examples

|  |  |  |  |
| --- | --- | --- | --- |
| **Physical / Real-life Tool** | **Python Programming Tool** | **Use** | **Python Example** |
| Numbers | integer/floating point | Counting / Measurement / Sizing | 1 1.0 1e0 |
| Names | Character strings | Naming / association | "Ray", "town\_address" |
| Calculation | +, -,\*, / ,\*\*, math… | Calculation | 1+2\*3/4\*\*5 math.sqrt(4) |
| Box / Shelve - with label | Variable | Holding something for later use | a, total |
| Shelves | lists | Containing numbers of things |  |
|  | loops | Repeated operation |  |
|  |  |  |  |

## more parts of python (examples in python\_introduction.py) - Slide 20

## Slides 23-27 - Beginning of Course Project - Programming the game of Twenty Questions

## Slide 28 (After doing first.py exercise)

Class 2 More Python, further on Class Project

## Startup Exercise: Make an example of everything we covered.

Create file: exerciseDDMMMYYYY.py [10 min]

## Slides: 13, 20: Review Homework python\_introduction.py

## Any Questions? [10 min]

## Slides: 23-28: Review Twenty Questions Project first.py [5 min]

## Slides: 21, 22: Introduction Lists, Dictionary [10 min]

## Slides: 34-40: Programming Ideas [15 min]

## Slides: 41-42: Python Tools Review [5 min]

## Slides: 47-65: Review / Reinforcement [10 min]

## Slides: 66: Exercise: timestable.py [10 min]

## Slides: 67-69: Exercise:timestableN.py [10 min]

## Slides: 69: Exercise left to the reader:timestableNbyN.py

## Slides: 70 : Second Iteration of Twenty Questions Project

## Exercise: second.py [10 min]

## Slides: 71: Our second.py version

## Slides: 72-74: Thinking of Versions [5 min]

## Slides: 75-78: More Python Review/Reinforcement

## Homework: Twenty QUESTIONS, iterations

## Third: third.py - Say if greater, less or equal

## Fourth: forth.py - Set target to random number

## timestable.py Slides: 66

## timestableN.py Slides: 67-69

## \*\* timestableNbyN.py (Extra Credit) Slides: 69

Class 3 Review Twenty Questions iterations

## Poll class for third.py, go over if < %50

### Third: Say if greater, less or equal

### 

## Poll class for forth.py, go over if < %50

### Fourth: Set target to random number

## import, Using the Web for help Slides: 79-80

## Startup Exercise: list\_friends.py Slides: 82

### 1.create file list\_friends.py

### 2. Make my\_friends = ["first friend", "Ray", "Sara"] # YOUR Friends

### 3. Write "for" loop to print out friends' names

### 4. Run program

## Startup Exercise: add\_friends.py Slides: 83

### 1. create add\_friends.py (starting with list\_friends.py, Save As)

### 2. Add loop to ask for new friend (input()

### 3. Inside the loop:

### a.If entered value is EMPTY (== "") print whole list.

### b.If not, add the input to list (my\_friends.append())

### 

## Functions - Divide and Conqure Slides: 84-88

### Exercise: Improve add\_friends.py to add\_friends\_fun

### by writing and using print\_friends function

### to hold the print friends code

Class 4 Functions

## Poll on Twenty Questions Iteration 5

### Fifth: Say goals, rules before start,

### including "a number between…"

### Review: if < %70

## Home Work:

### Twenty Questions: Multi-game iteration6.py

### Exercise: product.py Slides: 90

### Exercise: print factors.py - improvements slides: 94-95

## Review Functions: 84-89

### Functions - parameters Slides: 91-93

### Functions: keyword parameters - "print" example Slides: 93

### Exercise 2: – Print factors – factors.py Slides: 94-95

Class 5 Functions - Example

## Startup Exercise - Everything you know

### Using chat (to everyone)

### Give Python code, and example of a programming concept /idea / tool

### 2 + 2 ### Computation, addition

### a = 2 ### variable, assignment

### if a > 3: ### test

### stuff

## Lists, A Review Slides: 96

## Polls on Homework:

### Twenty Questions: Multi-game iteration6.py

### Exercise: product.py Slides: 90

### Exercise: print factors.py - improvements slides: 94-95

### 

## Functions - Larger Use Example - friends\_family

### friends\_1 A start: Slides: 98

### functions: list, test list

### functions: add, test add

### testing calls function at end

### friends\_2 Improving Slides: 99

### friends\_3 Checking Slides: 100

### friends\_mod Including Self-Test Slides: 101

### friends\_4 Code in friends\_mod.py, facilitating reuse Slides: 102

### from friends\_mod import \* # ALL

### # Better ???

### from friends\_mod import add\_friends, list\_friends

### 

### Homework:

### Twenty questions:

### Ask again if user types an illegal number iteration7.py Hint: Use try:, except:

### Expand friends\_family:

### 1. Only add a friend only if not already there

### 2. Allow case insensitive match on if\_friend/add friend                       Only need friends\_mod.py + friends\_4.py for testing.

### Start with friends\_mod, 1-3 were used for development

### Note you should save the original files so you can compare / contrast/

### test for errors.

Class 6 – Dictionaries, Classes and things

### Homework Polls:

### 1. Twenty questions iteration7.py:

### Ask again if user types an illegal number iteration7.py Hint: Use try:, except:

### 2. Expand friends\_family:

### 1. Only add friend if not already there

### 2. Allow case insensitive match on if\_friend/add friend Only need friends\_mod.py + friends\_4.py for testing.

### Question: What is (the most) confusing about Python so far?

### Send in Chat

### 

### Question: What to do you want most to learn / be covered?

### Send in Chat

### More detail on Strings Slides 103-105

### 

### Dictionary (AKA hash, associative array) Slides: 106-107

### When would you use a dictionary? When would you use a list?

### friends\_family - Use dictionary inplace of list Slides: 108

### for each file, starting with friends\_1.py

### The adventursome could skip to friends\_mod.py, friends\_4.py

### 1. save as ..\_dk\_N.py (e.g. friends\_1.py goes to friends\_di\_1.py)

### 2. modify my\_friends from a list to a dictionary

### my\_friends = {}

### 3. traverse list:

### for friend\_name in my\_friends:

### friend = my\_friends[friend\_name]

## Questions:

### What is (the most) confusing about Python so far?

### before class: email, Send in Chat

### 

### What to do you want most to learn / be covered?

### before class: email, Send in Chat

## Classes - Reason, Definition, Use Slides: 116-121

## Classes - An Example: Person, PersonGroup Slides: 122-123

### Files: person\_classes/ person.py, person\_group.py

Class 7 – Files and stuff.

## Homework Polls

### Multiplication tables

* 1. Object Oriented version of friends/family

## What Now?

## Files - Data that stays around Slides 109-110

#### Exercise: Create a file Slide 111

#### Exercise: Read a file

#### Files - a simple data base Slides 114-117

#### Files - check for errors - try/except

### Files: file\_io/ create\_file\_try.py, read\_file\_try.py