BT furtHER
Digital Intensive Programme

Module 02 - Python Fundamentals

Day 01

Welcome!

About the Instructor - Coding:

- Using C since 2006
- Undergraduate project used matlab to create a TTS software 2008
- Using Python since 2009 (Test processing & TTS)
- Frequently uses Python, Linux Bash, tcsh shell from 2012 (ASR, ML)
- PURE DATA from 2012 (music, speech synthesis)
- Java from 2012
- SQL from 2017 - learned through work
- Very occasionally uses C++, never used C# (but can call a world top 1% c#/java programmer for help)

About the Instructor - Teaching/Working:

- Teaching
  - Uni teaching/demonstrating PURE DATA & Java (2013)
  - Uni teaching/demonstrating Python (2014-2018)
  - C for saudi programme -design & deliver, won an international awards (2017-2018)
  - Published 6 GCSE lesson packs for CS and Automatic Speech Recognition (2018)
  -
- Working
  - Secretary in a agriculture business company in Chicago (2011-2012)
  - Residence mentor and run welfare officer, taking care of 100 1st year Uni students (2016)
  - Software hut project manager (2016-2017)
  - Using Python, Bash/tcsh and C++ in PhD (2013-2018)
  - Using Python and SQL in data scientist job (2017-2018)

However there still many things I don't know!

My English is not perfect and I can still make silly mistakes.

Computer science has grown rapidly in the last 20 years and this will continue.

The aim of the course is **coaching you to build an independent learning skill** -- make use of google and stackoverflow, be brave for solving problems!

Yoda said: "Do or do not, there is no try." -- So let's **do it and refactor** after doing something!

Introduce yourself

- Correct pronunciation of name.
- Where did you come from?/Where are you currently based?
- Previous experience?
- Why you joined the course?
- What you want to get from the course?
- Future plans?

Installations

Spyder - large black laptop

- Open Spyder
- Basic instructions of Spyder:
  - File editor
  - Console
  - Tool-> Preferences-> Run (configure)
  - Tool-> Preferences-> Syntax coloring

Pip - small laptop for python libraries ch14 onwards

- Open Anaconda prompt
- Type: pip --version
- Type: pip install Flask
- Type: python
- Type: import flask

Module 2/ch01 folder

Open bash (your main laptop):

Make sure you have: **Desktop/module2/ch01**

Q/A

Any questions about installation, spyder ?

Day 01
Learning outcome

LEARNING OUTCOME - Curriculum Chapter 01:

- Introduction to Python.
  - Clean code rules!
  - How to use the python console?
  - How to use the editor and run .py files?
  - Basic data types (int, float, str)
- Object Oriented Programming concept.
- Python online editor.

Introduction to Python

Introduction to python

Python is a powerful, open-source and easy to learn programming language. wiki/Python

Top 1 or 2 in any ranking of programming languages

Introduction to python - Advantages:

- Readability.
- Express concepts in fewer lines (compared to C++/C/java etc).
- String manipulation (NLP, search engine, info retrieval, auto translation).
- Data analysis (data scientist, machine learning, matplotlib, auto report).
- Back end web dev (Flask, Django)
- Took advantages of other languages and included them in python (Core developer Raymond hettinger).

CLEAN CODE RULES - Zen of Python:

- Beautiful is better than ugly (pure data example)
- Explicit is better than implicit (naming, structure etc)
- Simple is better than complex
- Complex is better than complicated
- Flat is better than nested (time/space)
- Readability counts (naming convention)
- If the implementation is hard to explain, it's a bad idea (assessment)
- If the implementation is easy to explain, it may be a good idea (assessment)

How to use python console

Type the following operations into your **python console** to see what happens (lower right window), anyone can tell me the results before demo?

```
2 + 3
2.9 + 3.2
(3.2 / 2.0) + 7
2**3
4**(1/2)
str(6) + ‘a’
```

Run .py file

Type the following operations into your **python editor** to see what happens (left editor and save the file ch1_yourFirstName.py):

```
2 + 3
2.9 + 3.2
(3.2 / 2.0) + 7
2**3
4**(1/2)
str(6) + 'a'
```

Run .py file

Save the file as ch1_yourFirstName.py

Beware -- do not name your file ‘python’ or any other key words!

Run the .py file by clicking the green button 

Run .py file

Type the following operations into your **python editor:**

```
A = 2 + 3
B = 2.9 + 3.2
C = (3.2 / 2.0) + 7
D = 2**3
E = 4**(1/2)
F = str(6) + 'a'
```

Then run the code by pressing the run button, see what happens?

(see right top window)

Run .py file

Now try this version in your **python editor:**

```
A = 2 + 3
B = 2.9 + 3.2
C = (3.2 / 2.0) + 7
D = 2**3
E = 4**(1/2)
F = str(6) + 'a'
print (A)
print (B)
```

Anything you have noticed?

Anything you have observed?

Number data types:

int data type: 2, 3, 10, -10, 300

float data type: 2.9, 3.2, -4.5

Operation?

** and **(1/2) ?

Anything you have observed?

str data type:

‘Python course’

‘123456’

‘ ‘

str(6) + ‘a’

Anything you have observed?

First function learned:

str()

Casts any other data into str data type.

-- Try casting some int or float data into str

Can you cast ‘6’ back to int or float?

Can you cast float to int and vice versa?

Anything you have observed?

Second function learned:

print()

You can put number, string, operation, variable etc in it to let your python console give responses.

Beware -- whatever is putting in print() needs to be the **same data type**, **preferably str**: e.g. print (str("4+5")+' '+'abcd')

Anything you have observed?

Single quote and double quote for str type:

print (str(‘4+5’)+’ '+'abcd')

print (str("4+5”) + “ ” + “abcd”)

print (str("4+5’)+” ”+'abcd') -- this will fail, where is the error?

Anything you have observed?

What else?

Anything you have observed?

Variable explorer

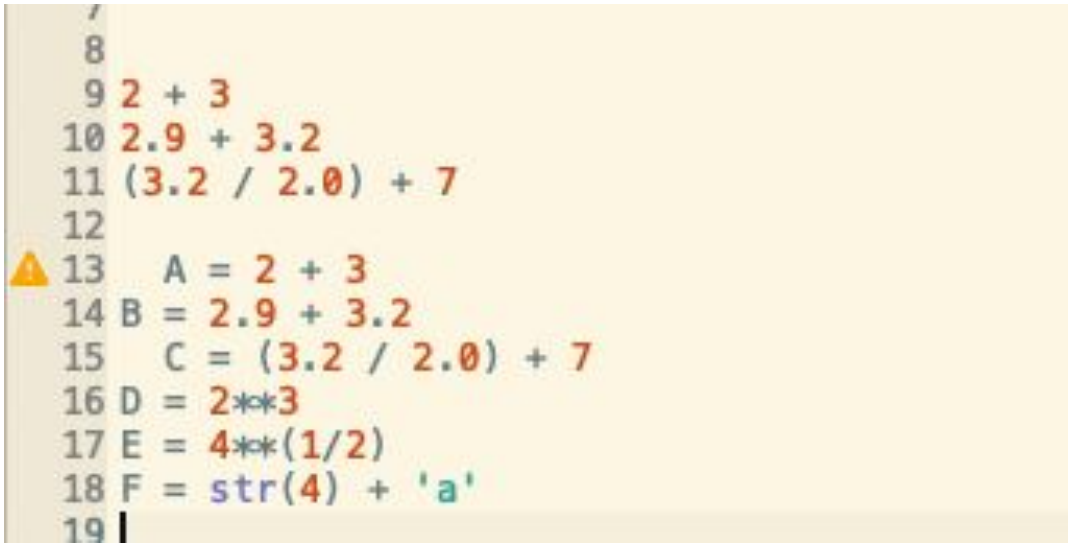
Name	Type	Size	Value
A	int	1	5
B	float	1	6.1
C	float	1	8.6
D	int	1	8
E	float	1	2.0
F	str	1	4a

Console 1/A | ch1_consoleDemo.py/A

```
In [7]: runfile('/Users/wuchenhao/Desktop/CFG/ch1_consoleDemo/
ch1_consoleDemo.py', wdir='/Users/wuchenhao/Desktop/CFG/ch1_consoleDemo')

In [8]: A
Out[8]: 5

In [9]: print (4_5)
45

In [10]: print (4+5)
9

In [11]: print (str("4+5")+' '+'abcd')
4+5 abcd
```

INDENTATION

If you add some spaces before writing the code, what will happen?



Be warned!

```
11 (3.2 / 2.0) + 7
12
13   A = 2 + 3
14 B = 2.9 + 3.2
```

```
  File "/Users/wuchenhao/Desktop/CFG/ch
line 13
    A = 2 + 3
    ^
IndentationError: unexpected indent
```

— **Indentation** is very important in Python. If you precede your expression with even one stray space character, the Python shell will complain of an Indentation Error and refuse to execute your code.

Be warned!

— **Save your Python file** every time you make some changes!!!

Day 01 learning outcome - Curriculum Chapter 01:

- Introduction to Python.
  - Clean code rules!
  - How to use the python console?
  - How to use the editor and run .py files?
  - Basic data types (int, float, str)
- Object Oriented Programming concept.
- Python online editor.

Familiar with python console and editor

OOP concept

Object Oriented Programming concept

Homework 1: find out what object oriented programming is and let me know tomorrow!

Private, public, local, global, abstraction..etc

Object Oriented Programming concept

Object Oriented Programming (OOP) has become the dominant programming paradigm over the last 20 years.

OOP was developed to make it easier to create and/or modify large, complex software systems.

Object Oriented Programming

Everything is an object in python:

e.g. ‘hello code first’ is an object of the str class (str data type)

run ‘hello code first’.title() and 'hello code first'.split() in your python console to see what happens?

Object Oriented Programming

str class, ‘hello code first’ object:

.title() and .split() are methods (or functions) which only apply to str class objects.

Object Oriented Programming

Class: describes a type of object. e.g. str

Object: example of that type. e.g. ‘python course’

.method(): used only for that type of object. title() and .split()

Object Oriented Programming

You will come across it in Python in chapter 5

There will be an OOP project recap section in chapter 13.

You will be encouraged to use OOP throughout module 3.

-- You still need to do your homework and tell me what it is tomorrow!

Python online editor

VISUALIZE CODE AND GET LIVE HELP

http://pythontutor.com/

Multiple languages

https://repl.it/

-- Try with python 2 and tell me the difference!

Anything you have observed?

Python 2 operation. You need to be aware of the difference:

(3 / 2) + 7 vs (3.0 / 2) + 7

```
> (3 / 2) + 7
=> 8
> (3 / 2.0) + 7
=> 8.5
>
> print 'haha'
haha
> print ('haha')
haha
```

Q / A?