PHY153
Lecture 4

J. Kiryluk, Stony Brook University

Numbers, strings, lists, “for” loop, code tracing

Online resources:
https://docs.python.org/3/tutorial/index.html
https://docs.python.org/3/library/index.html

Read & Practice @ home:

Tutorial exercises:
1) https://docs.python.org/3/tutorial/introduction.html#numbers
2) https://docs.python.org/3/tutorial/introduction.html#strings
3) https://docs.python.org/3/tutorial/introduction.html#lists
4) https://docs.python.org/3/tutorial/controlflow.html#for-statements

Data Types: Numbers

- different types (integer, float, complex)

The integer numbers (e.g. 5, 4, 20) have type **int**, the ones with a fractional part (e.g. 5.0, 1.6) have type **float**

Data conversion

Python does not convert, but creates a new object of a type you want.

`float()` will create a new float (decimal number) from a string or integer.
`int()` will create a new integer from a string or float.
`str()` will create a new string from a number (or any other type).

Functions

```
>>> a = 24
>>> b = float(a)
>>> c = 38.4
>>> d = int(c)
>>> e = int(38.8)
```

What are the values of a, b, c, d, e?

Data Types: Strings

Strings: set of characters in single or double quotation marks.
+ is the string concatenation
• is the repetition operator

Example:
Create a python file in an appropriate directory and an appropriate file name:

quote= “All of physics is either impossible or trivial. It is impossible until you understand it and then it becomes trivial."
author=“Ernest Rutherford”

```
print(quote)
print(quote[0])
print(quote[2:5])
print(quote[2:])
print(quote + author)
```

What do these lines mean?

Data Types: Lists

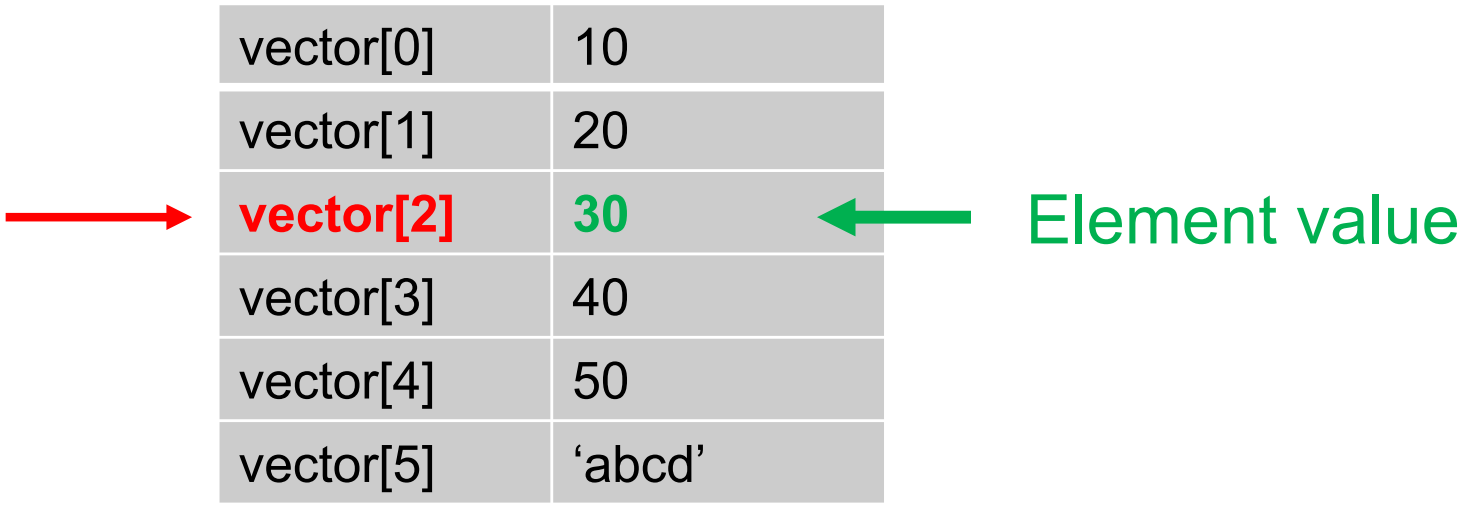
Example of a list :

vector = [10,20,30,40,50,'abcd'] # list declaration with initial values
# vector is a name of your choice, can be any name you like (no spaces)

new_vector = [ ] #creates an empty list

An element of the list named vector can be accessed
Through the index operator using the following syntax: vector[i]
where i is an index

List element at index 2



Data Types: Lists

L4ex1.py

```
#Lists: most versatile data types, contains items like:
list1=['abcd', 1234, '1234',34.6]
list2=[1,2,3,4,5,6]
#Similar to vectors/arrays, but lists elements can be of different types.
#Accessing elements, e.g.:
a=list1[0]
print(list1[0])
print (a)
b= list2[3]
print(list2[3])
print (b)
#Adding lists:
list = list1+list2
print (list)
list = [0,100, 2, 3] + list1 + list2
print (list)
l1= list[:2]   # elements from the beginning to position 2 (excluded)
l2 =list[4:]    # elements from position 4 (included) to the end
print (l1, l2)
```

Introduction to Loops

L4ex2.py

```
# Task: print all elements of a list named vector using a loop
# vector and element are names/variables
vector = [10,20,30,40,50,'abcd']
for element in vector:
    print(element)
```

Python's "for" statement iterates over the items of any sequence (e.g. a list), in the order that they appear in the sequence.

```
# the above code reads as "for each element in vector, print it.

print(vector)
print(vector[0])
print(vector[1])
print(vector[5])

#An element of the list can be accessed through the index operator I
#for example, 5th element of vector:
i=4
print(vector[i])
#vector[i] can be used as variable, for examples
x = vector[3]
print (x)
```

Introduction to Loops / Code Tracing

Code tracing (table) is done on paper:

L4ex3.py:

```
1  xlist=[11,21,31]
2  for x in xlist:
3      y=x+1
4      print (x,y,xlist)
```

All text is the box above is python code (4 lines of code)
Pay attention to indentation ("for" loop)!

Trace all **variables** in the code:

Line number (after this line is executed)	x	y	xlist
	Loop: 1st iteration		
1			
2			
3			
4			
5			
6			
7			

Use the Code Tracing to see what is happening

Line number (NOT part of the code): needed for code tracing

L4ex4.py

```
vector_idx = [0,1,2,3,4]
vector = [10,20,30,40,50]
new_vector = [ ]    #creates an empty list

for element in vector_idx:    #element is a variable!
      print(element)
      print(vector[element])
       #new_vector.append(element + vector[element])
       new_vector.append(vector_idx[element] + vector[element])
       #Append – adds a new element to the end of the list,
       # increasing the size/length of the lists
       #More lists methods later
       print(new_vector [element])

print(new_vector)
```

Trace this code on paper.

Introduction to Loops / Code Tracing

Trace the following code (a “typical” midterm problem)

L4ex5.py

```
y=100
xlist=[11,21,31]
ylist= []
counter = [0,1,2]

for x in counter:
        print x,y,xlist,ylist
        y=xlist[x]+1
        y*=2
        ylist[x]=y
        ylist.append(y)
        print (x,y)
```

Modify L4ex5.py in the following way:
Create an empty zlist, and calculate its elements in a loop according to the formulae: z(x) = x*0.5 + 15, where x are the elements of xlist.

L4ex6.py

Consider the following set of data (average temperature in January in NY):
http://www.weather.gov/media/okx/Climate/CentralPark/monthlyannualtemp.pdf

Use the followning average temperatures (in F) in January:

38.0, 34.5, 29.9, 28.6, 35.1, 37.3, 29.7, 32.5, 27.9, 36.5, 37.5, 40.9, 31.3, 24.7, 27.5, 39.9,
33.6, 31.3, 33.9, 40.0, 32.2, 30.5, 37.5, 25.6, 36.3, 35.7, 34.9, 41.4, 37.4, 29.5, 32.3, 34.1

Based on examples, write a python code (a script) that does the following:

- creates a list that consists of above data
- creates a loop that goes over all values and calculates corresponding values in C degrees.

Fahrenheit to Celsius conversion: C=(F-32) * 5/9

Lecture 4 Assignment/HW2:

Create L4ex1.py, L4ex2.py, L4ex3.py, L4ex4.py, L4ex5.py, , L4ex6.py (see Lecture4 file, under Documents)

- Execute them – if you have errorrs, fix them.
- Add comments explaining the meaning of individual lines of the code
- **E-mail me (one e-mail) running versions of L4ex1.py, L4ex2.py, L4ex3.py, L4ex4.py, L4ex5.py, L4ex6.py and before the end of class or (if you need more time) before Tuesday (Sept 10) class. You email should contain files as an attachment (or multiple attachments)**

Thank you!

Questions, comments:
e-mail: Joanna.Kiryluk@stonybrook.edu