Python: Variables*

Arash Pourhabib[†]

1 Basic Variable Types

- Variables in Python are "Objects"
- Basic objects in Python:
  - `int`: Integer numbers such as -3, 5, 126
  - `float`: Decimal numbers such as -3.0, 5.8, 126.0256
  - `string`: Pieces of text such as "Hello World!", 'Hello World!'
  - `complex`: Complex numbers such as 3.6+2.8j

Examples:

```
i = 3      #integer
x = 5.0    #float
text1 = "hello"  #string (the same as text1 = 'hello')
c = 5 - 2.6j     # complex
```

Names are case-sensitive in Python. So, `x` is different from `X`.

"#" denotes a comment.

*References: (1) Langtangen, Hans Petter. A primer on scientific programming with Python, Fourth Edition. Springer, 2014. (2) https://docs.python.org/2/reference/

[†]School of Industrial Engineering and Management, Oklahoma State University

2 Other Variable Types

- List: includes a group of objects
- Tuple: a “constant list”, that is it can be viewed but you cannot change it
- Dictionary: a list where the index can be a text

Examples:

```
l = [8.3, 'hello', 3.5, 'hi', −2.3]  #this is a list
coordinates = −2, 3.5, 8           #this is a tuple
d = {'Oklahoma': 3,878,051 'Ohio': 11,594,163}
                                   #dictionary, population of each state
```

3 Accessing Elements

- To access an element of a string, list or tuple you can use the index of the element

Examples:

```
>>> print(l[0])        #print 1st item of list
    8.3                  #output from Python
>>> print(coordinates[1])  #print 2nd item of tuple
    3.5
>>> text = 'Hello'
>>> print(text[4])
    o
```

Indices start from 0 (zero). So, for example, index 2 refers to the **third** element (not second).

4 Extracting sublists

- `l[i:j]` is the sublist starting with index `i` and counting up to index `j−1`
- Similarly for strings and tuples

Extracting a subset of lists (or strings or tuples) is called “slicing”

Each slice is just a copy of the original data, and has the same type.

Examples:

```
>>> text = 'Hello World!'
>>> print(text[0:3])
    Hel
>>> print(text[3:7])
    lo W
```

A nested list, or a list of lists, can be used to represent matrices or tables.

```
>>> l = [[-2, 12, 6], [7, 2, 9], [4, -3, 15]]
>>> print(l[1][2])
    9
```

We will later use a more efficient way to represent matrices (package **numpy**).

5 Printing

Printing to the screen is accomplished with the **print** command.

```
>>> x = 5.0
>>> print(x)    #Basic printing no formatting
    5.0       #Python output
>>> print('The value of x is', x)
    The value of x is 5.0 #Python output
```

To format the output Python uses *format string syntax*. Format strings contain "replacement fields" surrounded by curly braces .

- Format strings contain "replacement fields" surrounded by curly braces .
- Anything that is not contained in braces is considered literal text.
- the replacement field can start with a *field_name* and a *format_spec*, which is preceded by a colon ':'

```
>>> print('The value of x is {x_val:.2f}.'.format(x_val = x))
    The value of x is 5.00 #Python output
```

".2f" means print a "float" with 2 decimal points.

Other presentation options for formatted printing:

Format	Meaning
s	character string
d	integer
f	float
e or E	exponential notation with either “e” or “E”
<	left align
>	right align
x.yf	x total width, y digits to right of decimal (minimum)
	instead of ”f” can be ”e”, ”E”

For more information see, Python documents:
https://docs.python.org/3/library/string.html#formatstrings