# Python refresher

Jeffrey Salmond

July 3, 2017

## Introduction

## What is Python?

- A *general purpose* programming language
- First released in 1991
- Designed to emphasise readability of code

### Key features

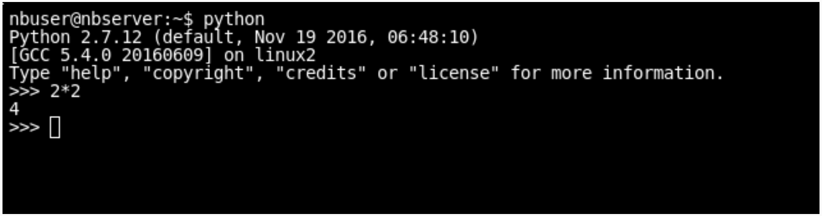
- an *interpreted* language
- Dynamic typing
- Automatic memory management
- “Batteries included” comes equipped with a large library

### Why use Python?

- Easy to learn
- Huge ecosystem of packages and libraries
- Very popular among data scientists

## Running Python

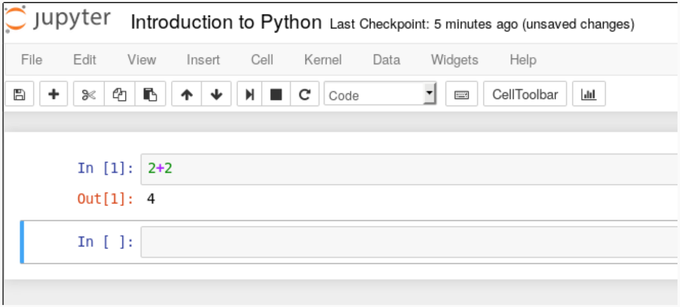
## Running python: The terminal



```
nbuser@nbserver:~$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 2*2
4
>>>
```

- works best on *nix type systems
- everything is under full control
- easy to setup

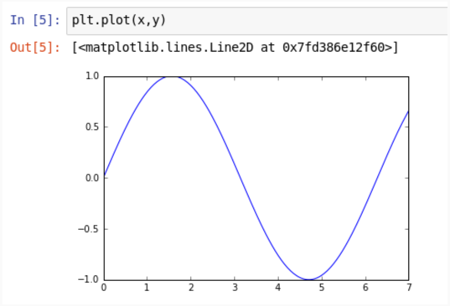
## Running python: Jupyter notebooks


jupyter
Introduction to Python Last Checkpoint: 5 minutes ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Data
Widgets
Help
Code
CellToolbar

```
In [1]: 2+2
Out[1]: 4
In [ ]:
```

- works everywhere you have a web browser
- some details are hidden from you
- setting up a webservice can be tricky

## Running python: Jupyter notebooks



```
In [5]: plt.plot(x,y)
Out[5]: [<matplotlib.lines.Line2D at 0x7fd386e12f60>]
```

- works everywhere you have a web browser
- some details are hidden from you
- setting up a webservice can be tricky
- can render plots and other complex visualisations

## Running python: In the cloud

Microsoft Azure `https://azure.microsoft.com`

- Microsoft's cloud computing platform
- Competitor to Amazon AWS, Google Cloud, etc...

Azure Cloud Notebooks

- A free(!) service
- Hosts Jupyter notebooks in the cloud
- runs on powerful compute hardware
- This is where we will run python for this course

## Logging in to Azure Cloud Notebooks

1. Go to `notebooks.azure.com`
2. Either sign in or create a new microsoft account
3. go to `notebooks.azure.com/jeffrey-salmond/libraries/python-data-science`
4. click ‘clone’ to get your own copy of the course material
5. open the ‘python refresher’ notebook

## Python Basics

## Python source

```python
def say_hello(who):
  print("hello, ", who)

who = "KBTU"
say_hello(who) #call the function
```

Python is

- dynamically typed
- code comments start with #
- white-space is important

## Basics: Numbers

```
a = 1
b = 1.0

x = 1 + 1   # x = 2
y = 2 * 4   # y = 8
z = 1.0 * 8 # z = 8.0
3/2 # 1.5

2 > 4  # False
5 >= 2 # True

import math
math.sqrt(2) # 1.414...
```

## Basics: Strings

```
x = "Hello"
y = 'KBTU'

x + ' ' + y # "Hello KBTU"
x*3         # "HelloHelloHello"

z = """a really
really
long string"""

z[0] # 'a'

n = 99
"%d red balloons" % n # "99 red balloons"
```

## Basics: Lists

```
fibs = [1,1,2,3,5,8]

fibs[4] # 5
fibs[-2] # 5

[2,4]*4 # [2,4,2,4]

x = [1,2,3]
x + [4,5] # [1,2,3,4,5]
[99,98,97].append(96) # [99,98,97,96]

fibs[0] = 99
fibs # [99,1,2,4,5,8]
```

## Basics: Dictionaries

```
x = {'Harry': 'Potter',
     'Ron': 'Weasley',
     'Hermione': 'Granger'}

x['Harry'] # 'Potter'

x['James'] = 'Bond'
x = {..., 'James': 'Bond', ...}
```

## Basics: Loops

```
y = 0
for x in [1,2,3]:
  y += x
y # 6

z = 0
while z < 100:
  z = z + 1

b = []
for a in [1,2,3]
  b.append(a*2)
b # [2,4,6]

b = [a*2 for a in [1,2,3]]
```

## Basics: Functions

```
sorted([3,4,2]) # [2,3,4]

def times2(x):
  return x*2
times2(4) # 8

a = 3
def plus_a(x):
  return x+a
plus_a(1) # 4
a = 4
plus_a(1) # 5
```

## Python 2 vs Python 3

Python 2

```
print "hello"

3/2 # 1

u"⍰⍰" #default ascii
```

Python 3

```
print("hello")

3/2  # 1.5
3//2 # 1

"⍰⍰" #default utf-8
```

In this course, we will use Python 3 only!

## jupyter Notebooks

when we are running inside a notebook we have special functions

- time execution of a function

  ```
  %timeit f(x)
  # 2.22 µs ± 26.2 ns per loop (mean ± std. dev. of 7 r
  ```

- load code from a file

  ```
  %load myscript.py
  ```

- setup plots to appear inline (we will see more of this later!)

  ```
  %matplotlib inline
  ```

## Errors!

```
sorted(3) #an error!
```

```
Traceback (most recent call last): File "<stdin>",
line 1, in <module> TypeError: 'int' object is not
iterable
```

## Exercise: Fibonacci

Calculating the Fibonacci sequence

$$f_n = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ f_{n-1} + f_{n-2} & \text{otherwise} \end{cases}$$

## Exercise: Fibonacci

Method 1: Recursive

```python
def fib_1(n):
  if n < 2:
    return n
  else:
    return fib_1(n-2) + fib_1(n-1)
```

Method 2: Iterative

```python
def fib_2(n):
  a, b = 0, 1
  for i in range(n)
    a, b = b, a+b
  return b
```

- enter both methods into a notebook
- write a loop to display the first 20 Fibonacci numbers
- how long does it take for each method to calculate the 33rd number in the sequence?

## Exersise: Largest element

write a function to determine the largest element of a list

```
def largest_element(xs):
  # your code here

largest_element([4,3,6,9,1,2]) # 9
```