

# Python Basics

## CME451 Tutorial

Instructor: Francis M. Bui  
GTF: Hao Zhang

Department of Electrical and Computer Engineering  
University of Saskatchewan

January 5, 2017

# Outline

Introduction

Python Installation

Python Basic Syntax

# Introduction

## Why Python?

- ▶ Open source;
- ▶ Easy to learn and fast to write;
  - ▶ Syntax similar to MATLAB
  - ▶ Shorter code than C/C++
  - ▶ Almost like pseudocode
- ▶ Abundant supporting libraries for various applications.
  - ▶ e.g. the well-documented `scikit-learn` library.

# Introduction

## An example

```
1 # quick sort in Python
2 def quicksort(arr):
3     if len(arr) <= 1:
4         return arr
5     pivot = arr[int(len(arr) / 2)]
6     left = [x for x in arr if x < pivot]
7     middle = [x for x in arr if x == pivot]
8     right = [x for x in arr if x > pivot]
9     return quicksort(left) + middle + quicksort(right)
10
11 print(quicksort([3,6,8,10,1,2,1,11,11]))
12 # Prints "[1, 1, 2, 3, 6, 8, 10, 11, 11]"
```

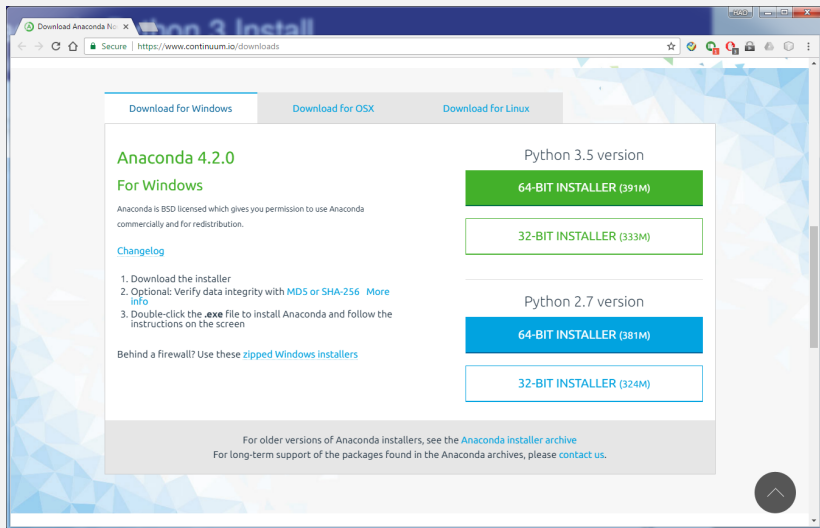
# Python Installation

## Python Distribution

- ▶ Official Python
  - ▶ <https://www.python.org/>
- ▶ **Anaconda Python**
  - ▶ <https://www.continuum.io/downloads>
- ▶ Enthought Canopy, miniconda, ...

# Anaconda Python 3 Install

## Download Installer



The screenshot shows a web browser window with the URL <https://www.continuum.io/downloads>. The page has three tabs: "Download for Windows", "Download for OSX", and "Download for Linux". The "Download for Windows" tab is active. The main content area is titled "Anaconda 4.2.0 For Windows". It includes a paragraph stating "Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution." and a link to the "Changelog". Below this is a list of three steps: 1. Download the installer, 2. Optional: Verify data integrity with MD5 or SHA-256 (with a link to "More info"), and 3. Double-click the .exe file to install Anaconda and follow the instructions on the screen. There is also a note: "Behind a firewall? Use these zipped Windows installers". On the right side, there are two sections for Python versions. The "Python 3.5 version" section has two buttons: "64-BIT INSTALLER (391M)" and "32-BIT INSTALLER (333M)". The "Python 2.7 version" section has two buttons: "64-BIT INSTALLER (381M)" and "32-BIT INSTALLER (324M)". At the bottom, there is a footer with two links: "For older versions of Anaconda installers, see the Anaconda installer archive" and "For long-term support of the packages found in the Anaconda archives, please contact us."

Download Anaconda No. X

Secure | <https://www.continuum.io/downloads>

Download for Windows | Download for OSX | Download for Linux

### Anaconda 4.2.0

#### For Windows

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

[Changelog](#)

1. Download the installer
2. Optional: Verify data integrity with [MD5](#) or [SHA-256](#) [More info](#)
3. Double-click the .exe file to install Anaconda and follow the instructions on the screen

Behind a firewall? Use these [zipped Windows installers](#)

Python 3.5 version

64-BIT INSTALLER (391M)

32-BIT INSTALLER (333M)

Python 2.7 version

64-BIT INSTALLER (381M)

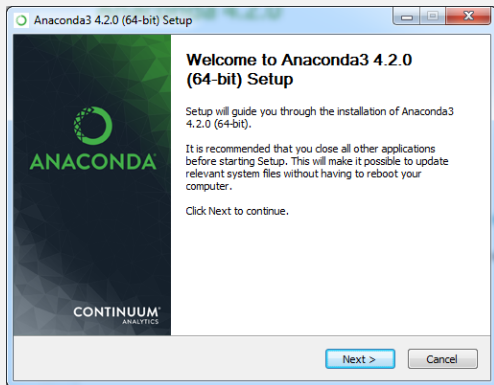
32-BIT INSTALLER (324M)

For older versions of Anaconda installers, see the [Anaconda installer archive](#)

For long-term support of the packages found in the Anaconda archives, please [contact us](#).

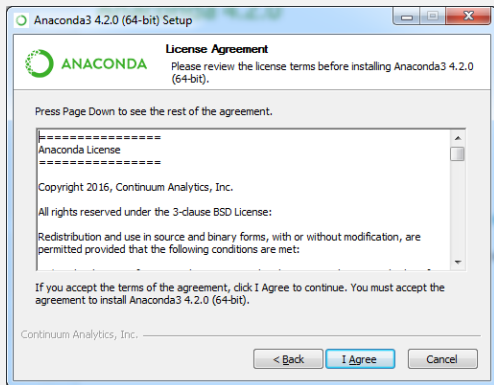
# Anaconda Python 3 Install

## Installation Process



# Anaconda Python 3 Install

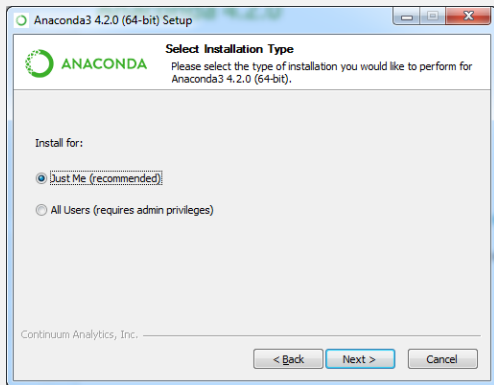
## Installation Process





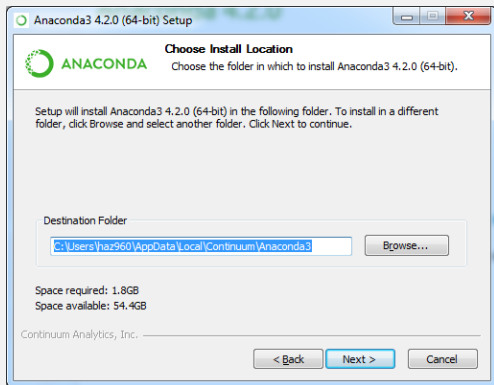
# Anaconda Python 3 Install

## Installation Process



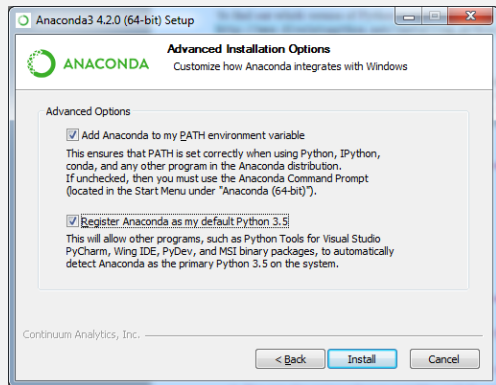
# Anaconda Python 3 Install

## Installation Process



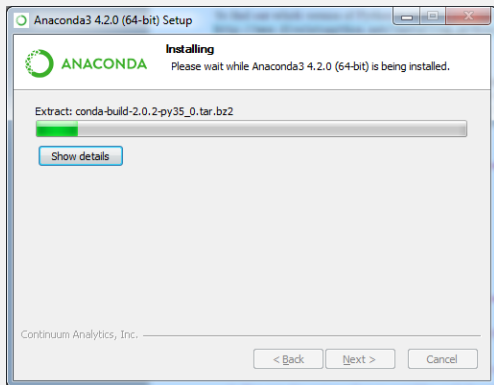
# Anaconda Python 3 Install

## Installation Process



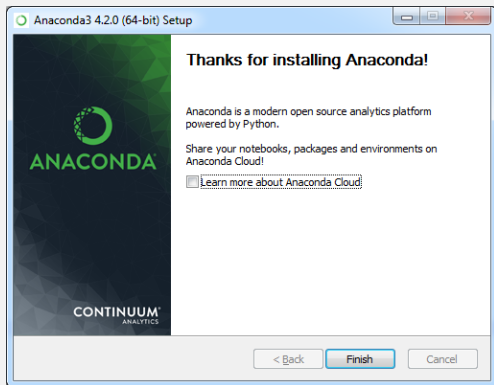
# Anaconda Python 3 Install

## Installation Process



# Anaconda Python 3 Install

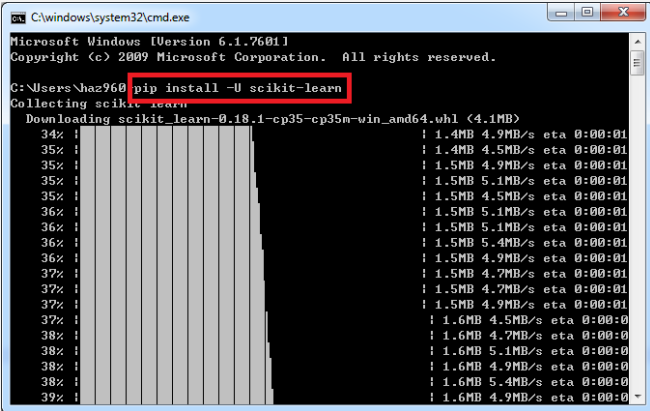
## Installation Process



# Anaconda Python 3 Library

## Scikit-Learn

- <http://scikit-learn.org/stable/install.html>

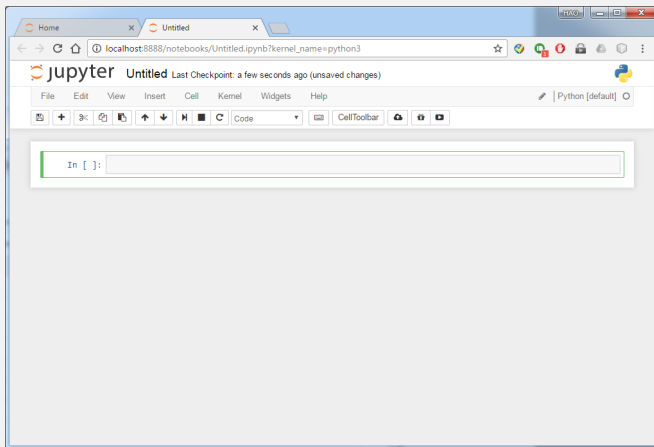


```
C:\windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\haz960> pip install -U scikit-learn
Collecting scikit-learn
  Downloading scikit_learn-0.18.1-cp35-cp35m-win_amd64.whl (4.1MB)
    34% |#####| 1.4MB 4.9MB/s eta 0:00:01
    35% |#####| 1.4MB 4.5MB/s eta 0:00:01
    35% |#####| 1.5MB 4.9MB/s eta 0:00:01
    35% |#####| 1.5MB 5.1MB/s eta 0:00:01
    35% |#####| 1.5MB 4.5MB/s eta 0:00:01
    36% |#####| 1.5MB 5.1MB/s eta 0:00:01
    36% |#####| 1.5MB 5.1MB/s eta 0:00:01
    36% |#####| 1.5MB 5.4MB/s eta 0:00:01
    36% |#####| 1.5MB 4.9MB/s eta 0:00:01
    37% |#####| 1.5MB 4.7MB/s eta 0:00:01
    37% |#####| 1.5MB 4.7MB/s eta 0:00:01
    37% |#####| 1.5MB 4.9MB/s eta 0:00:01
    37% |#####| 1.6MB 4.5MB/s eta 0:00:0
    38% |#####| 1.6MB 4.7MB/s eta 0:00:0
    38% |#####| 1.6MB 5.1MB/s eta 0:00:0
    38% |#####| 1.6MB 4.9MB/s eta 0:00:0
    38% |#####| 1.6MB 5.4MB/s eta 0:00:0
    39% |#####| 1.6MB 4.9MB/s eta 0:00:0
```

# IPython Notebook

- Type `jupyter notebook` from the command line



# Python Basic Syntax

```
1  '''
2  Python multiple lines comments
3
4  Python do not use ';' at the end of each line
5
6  Code blocks (function, loop, if) are defined by indentation
7  '''
8
9  # single line comment
10 # using of non built-in libraries
11 import math
12 import numpy as np
```



# Python Basic Syntax

## Data Type

```
1  # basic datatype:
2  #     integer, float, boolean, string
3  print(type(2))           # print <class 'int'>
4  print(type(2.0))         # print <class 'float'>
5  print(type('two'))      # print <class 'str'>
6  print(type(True))       # print <class 'bool'>
7  print(type(None))       # print <class 'NoneType'>
8
9  # datatype conversion
10 print(float(2))           # print 2.0
11 print(int(2.1))          # print 2
12 print(type(str(2.5)))    # print <class 'str'>
13 print(bool(0))           # print False
14 print(bool(2))           # print True
```

# Python Basic Syntax

## Math Operation

```
1 # basic math operations
2 x = 5
3 print(x + 1)    # add, print 6
4 print(x - 1)    # subtract, print 4
5 print(x * 2)    # multiply, print 10
6 print(x ** 2)   # exponent, print 25
7 print(x % 2)    # modulo, print 1
8 print(x / 2)    # divide, print 2.5, will print 2 in python 2
9 print(x / 2.0)  # divide, print 2.5
10 # other advanced operation should
11 # import math
```

# Python Basic Syntax

## String Operation

```
1 # string operations
2 hello = 'hello'          # define string with ' '
3 world = "world"          # define string with " "
4 print(len(hello))        # string length, print 5
5 hw = hello + ' ' + world # concatenate two strings
6 print(hw)                # print hello world
7 print('%s %s %d' % (hello, world, 12)) # print with format
8
9 print(hello.capitalize()) # capitalize a string, print Hello
10 print(hello.upper())      # convert to upper case, print HELLO
11 print(hello.replace('l', '(ell)')) # replace l with (ell)
12                             # print he(ell)(ell)o
13 print(' world '.strip()) # strip the leading and trailing whitespace
14                             # print world
15 print(hw.find('hello'))   # return index of first occurrence, print 0
16 print(hw[1:6])           # string slice, print 'ello '
17 print(hw.split(' '))     # split string by delimiter
18                             # print ['hello', 'world']
```

► <https://docs.python.org/3.5/library/stdtypes.html#string-methods>.

# Python Basic Syntax

## List Operation

```
1 # list operation
2 # list is equivalent to array, but resizable
3 empty_list = [] # create an empty list
4 simpsons = ['homer', 'marge', 'bart'] # create a list with three elements
5 print(simpsons[0]) # print element 0
6 print(len(simpsons)) # print list length
7 simpsons.append('lisa') # append element to the end
8 simpsons.insert(0, 'maggie') # insert element at index 0
9 simpsons.remove('bart') # search for instance and remove it
10 del simpsons[0] # remove element 0
11 simpsons[0] = 'krusty' # replace element 0
12 simpsons.sort(reverse=True, key=len) # sort by length in descending order
```

# Python Basic Syntax

## Function

```
1  def sign_of_num(x):
2      if x > 0:
3          return 'positive'
4      elif x < 0:
5          return 'negative'
6      else:
7          return 'zero'
8
9  for x in [-1, 0, 1]:
10     print(sign_of_num(x))
11
12  # Prints "negative", "zero", "positive"
```