

# Python for Data Science

---

Sirakorn Lamyai

August 11, 2019

Student, Kasetsart U.



## Sirakorn Lamyai

- DAKDL Laboratory, Kasetsart University
- Research Assistant Intern, 2019, Vidyasirimedhi Institute of Science and Technology
- Research Assistant Intern, 2018, Vidyasirimedhi Institute of Science and Technology
- Love drinking tea
- Knows a little about Python

# I know a little about Python

When I say I know *a little* about Python...

- I think there's some better methods than I'm using
- I think I do sometimes make mistakes
- There are tons of people who know things much more than me
- I think there's much more for me to learn!

# Prerequisite

A basic Python knowledge will do!

# Your expectations from this talk

Data Science

Python

Python environments

Jupyter

Python Data Structures

# Data Science

---

# The Data Science Process: OSEMNI

- **Obtain** data from relevant sources
- **Scrub**, sanitise, and clean the data into machine-understandable formats
- **Explore** significant and meaningful patterns with statistical methods
- **Model** construction for prediction and forecast
- **iNterpret** and use the results obtained
- **Iterate** and rethink about your outputs



# Why data?

Google

oxford shoes under \$200

🔍 🗨️ 🔍

🔍 ทั้งหมด 🖼️ ค้นรูป 📄 ช็อปปิง 📺 วิดีโอ 📅 ข่าวสาร ⋮ เพิ่มเติม 🛒 การตั้งค่า 🖱️ เครื่องมือ

ผลการค้นหาประมาณ 31,400,000 รายการ (0.84 วินาที)

ดูoxford shoes under \$200

ผู้สนับสนุน



Ted Baker Murain oxford shoes in...  
฿3,850.84  
£100.00  
ASOS



รองเท้าหนังวินเทจ oxford style งา...  
฿2,990.00  
Lazada Thailand



Timberland Stormbuck Plain oxford shoes...  
฿3,517.00  
Dressinn.com  
★★★★★ (48)



Ted Baker Ollivur brogue shoes i...  
฿3,850.84  
£100.00  
ASOS



SOLE Dunstan Black Shoes  
฿2,502.66  
£64.99  
Soletrader

12 Best Men's Dress Shoes Under \$200

- Kenneth Cole Reaction Last Laugh. ...
- Nordstrom Cusano Double Monk Shoe. ...
- Cole Haan Briscoe Wingtip. ...



# Why data?

The screenshot shows a Facebook interface with a blue header bar. The left sidebar contains navigation options: News Feed, Messenger, Watch, Marketplace, Shortcuts, and Explore. The main content area displays a sponsored post from 'Curated and Co.' featuring a collection of Berwick Penny Loafers. The post includes a detailed description of the shoes and a 'See More' link. The right sidebar shows a list of friends with their names, mutual friend counts, and 'Add Friend' or 'Remove' buttons. At the bottom, there is a language selector, privacy/terms links, and a chat button.

Facebook interface showing a sponsored post for Curated and Co. Berwick Penny Loafers.

**Curated and Co.** Sponsored

New Arrival Berwick Penny Loafer Collection !

Berwick Penny Loafer ที่มีรุ่นใหม่เข้ามาให้เลือกกันถึง 6 แบบเลยทีเดียวครับ ไม่ว่าจะเป็น Oiled 173 Suede, Polo Brown Suede ที่ได้รับความนิยมมากในต่างประเทศ รวมถึงหนึ่ง Smooth อีก 4 แบบที่สวยงามมากๆ ไม่ว่าจะเป็น Vegano Melize, Moka และสุดท้ายเป็นสีดำ Black Box Calf ที่คลาสสิกตลอดกาล

-----... [See More](#)

**Patinya Yongyai** (NotPty) 81 mutual friends  
[Add Friend](#) [Remove](#)

**Kanoktat Ninklam** 79 mutual friends  
[Add Friend](#) [Remove](#)

**Rung Nattayaporn** 3 mutual friends  
[Add Friend](#) [Remove](#)

**Chakri Lowphansirikul** 5 mutual friends  
[Add Friend](#) [Remove](#)

**Unnop Nushprasert** 15 mutual friends  
[Add Friend](#) [Remove](#)

English (US) · ภาษาไทย · Suomi · 日本語 · Español

Privacy · Terms · Advertising · Ad Choices · Cookies · More

Facebook © 2019

Chat (125)

# Why data?

The screenshot shows a Facebook user interface. On the left is a sidebar with navigation options: News Feed, Messenger, Watch, Marketplace, and Shortcuts. The main content area displays a sponsored advertisement for "Curated and Co." featuring a collection of brown leather loafers. A context menu is overlaid on the ad, providing options: "Hide ad" (Mark as irrelevant or repetitive), "Report ad" (Tell us about a problem with this ad), "Save post" (Add this to your saved items), "Why am I seeing this ad?" (highlighted in blue), "Turn on notifications for post", "Embed", and "More options". The right sidebar shows a list of friends with their names, mutual friend counts, and "Add Friend" or "Remove" buttons. At the bottom, there are links for language (English, Thai, etc.), privacy, terms, and advertising, along with a chat button indicating 125 active chats.

# Why data?



Data is the new oil

## With GUIs

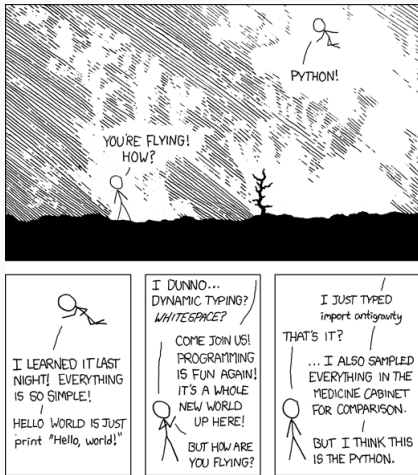
- Spreadsheets
  - Excel
  - Google Spreadsheets
  - Lotus 1-2-3
- Modelling and Visualisation
  - RapidMiner Studio
  - Weka
  - Tableau

## As programming languages

- For data insights
  - R
  - Python
- For data retrieval
  - SQL

# Python

---



Courtesy: xkcd (<https://xkcd.com/353/>)



## I *loved* Python...

- Read it, understand it
- Multiparadigm
- Batteris included
- Lots of great, great libraries!



pip

pip

- **PyPA** (**P**ython **P**ackaging **A**uthority)'s recommended package installer

pip

- **PyPA** (**P**ython **P**ackaging **A**uthority)'s recommended package installer
- Obtains packages from **PyPI** (**P**ython **P**ackaging **I**ndex)

pip

- **PyPA** (**P**ython **P**ackaging **A**uthority)'s recommended package installer
- Obtains packages from **PyPI** (**P**ython **P**ackaging **I**ndex)
- Many useful packages for us to use!







- Cross-platform Python Distribution





- Cross-platform Python Distribution
- Ships with its own package and environment manager



- Cross-platform Python Distribution
- Ships with its own package and environment manager
  - Its environment manager capability is not found in Python vanilla installation



- Cross-platform Python Distribution
- Ships with its own package and environment manager
  - Its environment manager capability is not found in Python vanilla installation
  - Fetches the packages from its own repository, not PyPI



- Cross-platform Python Distribution
- Ships with its own package and environment manager
  - Its environment manager capability is not found in Python vanilla installation
  - Fetches the packages from its own repository, not PyPI
- Aims for Data Science use



- Cross-platform Python Distribution
- Ships with its own package and environment manager
  - Its environment manager capability is not found in Python vanilla installation
  - Fetches the packages from its own repository, not PyPI
- Aims for Data Science use
- **Entirely separated Python**

## Environments 101: \$PATH

```
$ echo $PATH
/home/srakrn/.pyenv/plugins/pyenv-virtualenv/shims:/home/
srakrn/.pyenv/shims:/home/srakrn/.pyenv/bin:/home/srakrn
/.local/bin:/usr/local/bin:/usr/local/sbin:/home/srakrn/.
local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/
local/games:/snap/bin
```

## Different machines, different Pythons

On my laptop...

```
srakrn@epsilon-ubuntu:~$ which python  
/home/srakrn/.pyenv/shims/python
```

On my <https://charles.srakrn.me/> server...

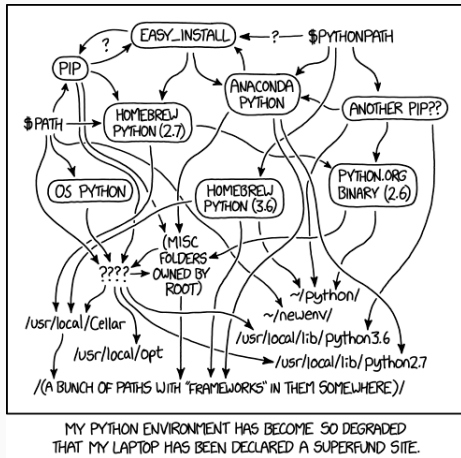
```
srakrn@charles:~$ which python  
/usr/bin/python  
srakrn@charles:~$ which python3  
/usr/bin/python3
```



## Installed pip

```
$ pip -V
pip 8.1.1 from /usr/lib/python2.7/dist-packages (python 2.7)
$ pip3 -V
pip 8.1.1 from /usr/lib/python2.7/dist-packages (python 3.6)
```

Perhaps now you understand me...



Courtesy: xkcd (<https://xkcd.com/1987/>)







Interactive computing environment











- Think of an online Jupyter Notebook provided by Google



- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server fram



- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server frame
  - In other words, your code are remotely executed



- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server fram
  - In other words, your code are remotely executed
- Could be more powerful for some tasks (like Deep Learning) than your computer



- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server fram
  - In other words, your code are remotely executed
- Could be more powerful for some tasks (like Deep Learning) than your computer
- Free!

`https://colab.research.google.com/`



+ Code + Text



RAM



Disk



Editing



```
[1] from datetime import datetime
```

```
name = input("Please input your name: ")
```

```
... Please input your name:
```

```
Tan
```

```
print("Hello, {}".format(name))  
print("It is now {}".format(datetime.now()))
```



## Caveats 1: Execution order

```
[2] a = 10
```

```
[1] a = 5
```

```
[3] print(a)
```

```
10
```





## Caveats 1: Execution order

```
[2] a = 10
```

```
[1] a = 5
```

```
[3] print(a)
```

```
↳ 10
```



You'll do a lot of out-of-order code execution!

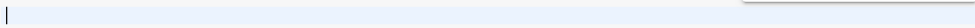
## Caveats 2: Cell edits

```
[1] a = 10
```

```
[2] a *= 2
```

```
[4] print(a)
```

```
↳ 60
```



## Caveats 2: Cell edits

```
[1] a = 10
```

```
[2] a *= 2
```

```
[4] print(a)
```

```
↳ 60
```



You might sometimes remove a cell, and that shows no visible trace without explicit query.

## Caveats 2: Cell edits

```
[1] a = 5
```

```
[2] a = 20
```

```
[3] print(a)
```

```
10
```



## Caveats 2: Cell edits

```
[1] a = 5
```

```
[2] a = 20
```

```
[3] print(a)
```

```
10
```



Jupyter Notebook offers no cell edited marks, while Colab offers them

## Caveats 2: Cell edits

```
[1] a = 5
```

```
[2] a = 20
```

```
[3] print(a)
```

```
↳ 10
```



Jupyter Notebook offers no cell edited marks, while Colab offers them (note: observe the greyed out cell number)

## Caveats 3: Be neat and tidy

Jupyter Notebook and Colab, unlike IDE and code editors, offers a relatively poor **clean code** tools

- Syntax error highlighting
- Autocomplete
- Linting
- Code formatter

### Sirakorn's Workflow Demo



### Sirakorn's Workflow Demo

*(Please don't be amused, this is **very** normal.)*

# Python Data Structures

---

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.
- Can store multiple data types.

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.
- Can store multiple data types.
  - This includes storing lists in a list

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.
- Can store multiple data types.
  - This includes storing lists in a list
  - So-called a **nested list**

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.
- Can store multiple data types.
  - This includes storing lists in a list
  - So-called a **nested list**
- Can be resized.



# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.
- Can store multiple data types.
  - This includes storing lists in a list
  - So-called a **nested list**
- Can be resized.
  - No need to declare its size on the first declaration.

```
1 a = [1, 2, 3, 4, 5]
2 a[0]      # Accessing elements
3 a[1:3]    # Slicing
```

Accessing list

```
1 a = [1, 2, 3, 4, 5]
2 a[0]      # Accessing elements
3 a[1:3]    # Slicing
```

Accessing list

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 a[0]      # Accessing elements
3 a[1:3]    # Slicing
```

Accessing list

- **Elementwise:** accessing one elements at a time)

```
1 a = [1, 2, 3, 4, 5]
2 a[0]      # Accessing elements
3 a[1:3]    # Slicing
```

## Accessing list

- **Elementwise:** accessing one elements at a time)
- **Slicing:** accessing a sublist

## List Functions

```
1 vowels = ["a", "e", "o", "u"]
2
3 # Get a's length
4 len(a)
5 # Append the new element to the end of a
6 a.append("y")
7 # Deletes the first occurrence of the element from a
8 a.remove("y")
9 # Inserts the item into a list with a specified index
10 a.insert(2, "i")
```

## List Functions

```
1 vowels = [1, 3, 2, 5, 4]
2 # Get the first index of a specified element
3 a.index(4)
4 # Sort a list and store into a new list
5 sorted_a = sorted(a)
6 # Sort a list, making changes directly to the old one
7 a.sort()
```