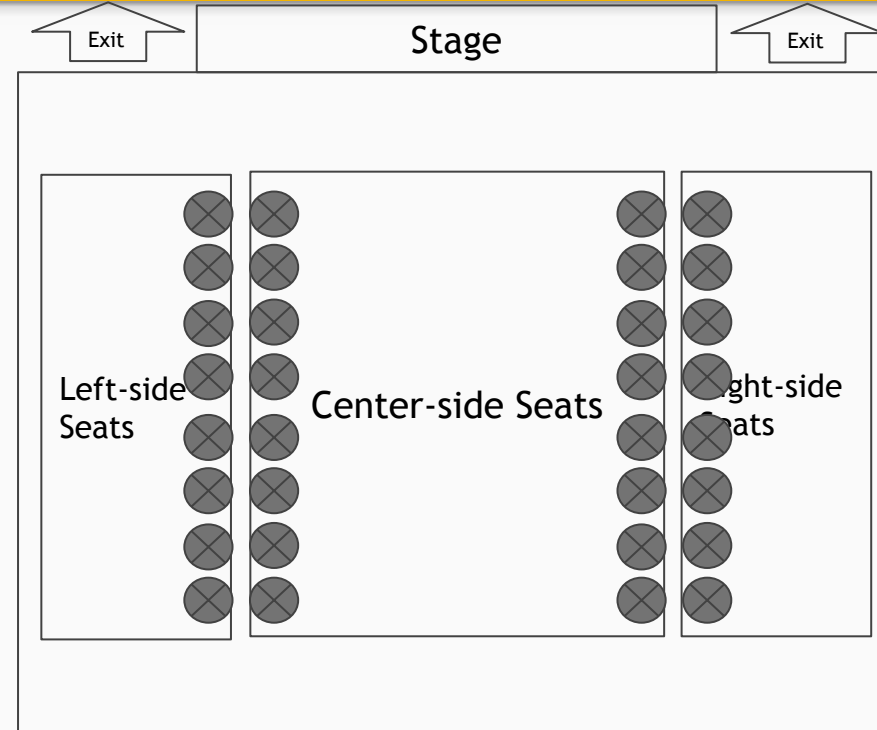


<https://app.sli.do/event/irXJiigN6piRy485RmyN9x>



# Yasumoto International Academic Park - YIA LT6

Limited Mobile Signal. Please use [on-campus wifi](#).  
32 Sockets. But please bring your own charger.



⊗ = sockets

# Python Basics - Programming 101

Variables, Controls, Loops

CUHK MSc Data Science & Biz Stat. Program

STAT5106 - Programming Techniques for Data Science

Week 1 @ 5 12 Sept 2024

# Self-Introduction

- 2008 July  
BSc in Statistics CUHK (1st Hons)  
in Data Science and Bus. Stat. Stream
- 2010 July  
MPhil in Statistics CUHK  
under Supervised by Prof. Ben Chan
- 2010 Oct - 2013 Sept  
Algo Trading - CASH Dynamic Opportunities Investment
- 2013 Sept - 2014 Sept  
IT Consulting - Long Term Intelligence
- 2014 Sept - 2015 Sept  
Government - Environmental Protection Dept
- 2016 Mar - 2018 Dec  
Energy - CLP Power Limited
- 2019 Feb - 2021 Sept  
Banking - Hang Seng Bank Limited
- 2021 Oct - 2024 Oct  
Maritime - Fleet Management Limited
- 2024 Oct - ...?  
To Be Announced...



[www.linkedin.com/in/kyalanlo/](https://www.linkedin.com/in/kyalanlo/)



[kyalan.2013@gmail.com](mailto:kyalan.2013@gmail.com)  
[kwokyuenlo@cuhk.edu.hk](mailto:kwokyuenlo@cuhk.edu.hk)



# Ability

I am a full-stack data scientist - can be end-to-end hands on, and act as team leader.

Cloud:



Data:  
Web Scraping / API  
SQL (MySQL/Oracle/MS SQL/T-SQL)  
HDFS type data (Azure Datalake /  
Google Bigtable)  
Google BigQuery  
Coursera & UCSD Big Data Spec



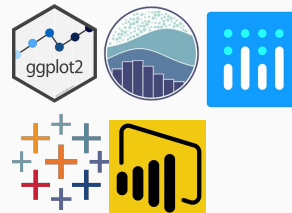
Analysis:  
20-years R Experience  
10-years Python Experience  
20-years SAS Experience  
SAS Advanced Programmer  
SAS Cert Stat Analyst  
SPSS Cert Modeller



Modelling:  
Keras & Tensorflow; XGBoost and all common  
type NNs  
Andrew Ng's Deeplearning.ai spec  
Deeplearning.ai Tensorflow Developer spec  
Deeplearning.ai NLP spec  
Michigan U Recommender System Spec



Visualization:  
R ggplot, Python seaborn, plotly  
Tableau Desktop Cert Professional  
(Toppest grade, 1 of 2 in HK)  
MS Power BI Cert Specialist



## Muhammad Danish

*BSc in Data Sciences (CityU), currently pursuing Ph.D. in Statistics (CUHK)*



[mdanish@link.cuhk.edu.hk](mailto:mdanish@link.cuhk.edu.hk)



Lady Shaw Building 130, 3943 7939



<https://sites.google.com/view/muhdanish/home>

## Catherine Yu

*currently pursuing Ph.D. in Statistics (CUHK)*



[cxryu@link.cuhk.edu.hk](mailto:cxryu@link.cuhk.edu.hk)

# Agenda

Before Mid-term = ***Programming***

After Mid-Term = ***DS Application***

## OPTIONS

1. Delete the missed Lesson, and condense contents into Lessons “Week 2” - “Week 6”
2. Reschedule the missed lesson into any days in remaining 5 weeks
3. Shift the full schedule forward 1 week.  
i.e., the following will be deferred
  - a. Assignment deadlines
  - b. Mid-Term schedule
  - c. Final Project schedule

Date			
Week 1 - <del>5-Sept</del> 12 Sept			ery, Sort, Merge,
Week 2 - <del>12-Sept</del>			ion in Pandas &
Week 3 - <del>19-Sept</del>			eborn
Week 4 - <del>26-Oct</del>			adline**
Week 5 - <del>3-Oct</del>			
Week 6 - <del>10-Oct</del>			adline**
<del>17-Oct</del>	Mid-term Take-Home Exam (No class)	Week 12 - 20 Nov	ChatGPT with Large Language Model Applications
		4-Dec	**Final Project deadline**
		7-Dec (Saturday)	**Final Project Presentation Day**

# Agenda

## Before Mid-term = Programming

Date	Topic
Week 1 - <del>5-Sept</del> 12 Sept	Python Basics, Programming 101 - Variables, Controls, Loops
Week 2 - <del>12-Sept</del>	Programming 102: Exception Handling, Functions, String
Week 3 - <del>19-Sept</del>	Python Data Structures - List, Dictionary, Tuple Regular Expressions <b>**Assignment 1 (for weeks 1, 2) deadline**</b>
Week 4 - <del>26-Oct</del>	Introduction to Data Science Python Packages, Numpy, Pandas basics Python vs R in Programming
Week 5 - <del>3-Oct</del>	Open Data: Web Scraping <b>**Assignment 2 (for weeks 3, 4) deadline**</b>
Week 6 - <del>10-Oct</del>	APIs, with More Example on Open Data
<b>17-Oct</b>	<b>Mid-term Take-Home Exam (No class)</b>

## After Mid-Term = DS Application

Date	Topic
Week 7 - <del>24-Oct</del>	Data Processing in Pandas - IO, Query, Sort, Merge, Missing Value Handling
Week 8 - <del>31-Oct</del>	Exploratory Data Analysis, Simulation in Pandas & Numpy
Week 9 - <del>7-Nov</del>	Data Visualization in Matplotlib, Seaborn Introduction to Tableau <b>**Assignment 3 (for weeks 7, 8) deadline**</b>
Week 10 - <del>14-Nov</del>	Bayesian Computation in Python
Week 11 - <del>21-Nov</del>	Word Data Preprocessing in Python Introduction to Machine Learning Open Model: Tensorflow-hub <b>**Assignment 4 (for weeks 9, 10) deadline**</b>
Week 12 - <del>28-Nov</del>	ChatGPT with Large Language Model Applications
<b>4-Dec</b>	<b>**Final Project deadline**</b>
<b>7-Dec (Saturday)</b>	<b>**Final Project Presentation Day**</b>



- Python
  - Dr. Charles Russell Severance (2016) - Python for Everybody Exploring Data in Python 3, CreateSpace
  - Python for Everybody Specialization - Coursera & Michigan University  
<https://www.coursera.org/specializations/python>
  - Jake VanderPlas (2016) - Python Data Science Handbook, O'Reilly
  - Applied Data Science with Python Specialization - Coursera & Michigan University  
<https://www.coursera.org/specializations/data-science-python>
  - Learning Materials in Tableau, <https://www.tableau.com/learn>
  - ChatGPT Short Courses - Deeplearning.ai, <https://www.deeplearning.ai/short-courses/>

# Agenda

- Lecture
- Classwork
- Break
- Blow Water
  - What's happened about Data Science ?
  - Career in recent market
- Assignment, Midterm, Final
  - Marking Scheme

Homework	40% = 10% * 4 assignments
Midterm Exam	30%
Final Project	30% (Report + Present)

# Last year I talked about this first



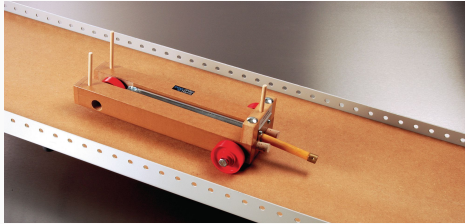
Data



Analysis



Visuals

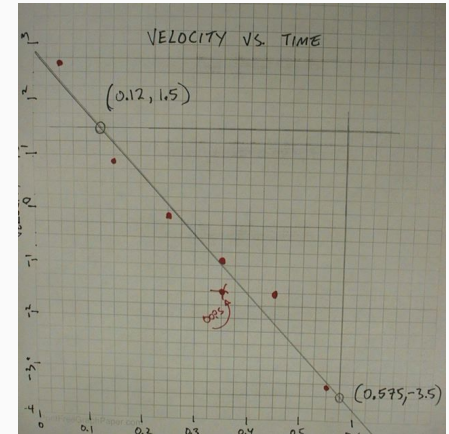
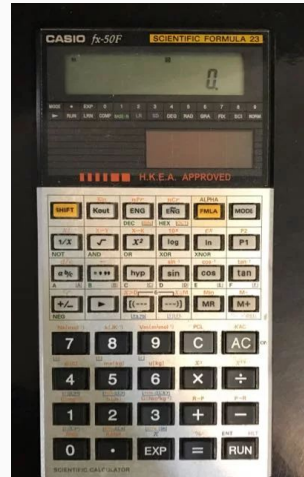


Mand Probe and Rate Sheet  
Learner: [redacted] Week of: 5/14/11

ITEM	Vertical bridge	In-Item request	Prior PTV	Check	M	W	SP	SA
Juice/cup	✓	I	5	✓	✓	✓	✓	✓
Fork/spoon	✓	I	2	✓	✓	✓	✓	✓
Fork/apple/bread	✓	I	1	✓	✓	✓	✓	✓

Date	Total Mating Time (min)	Mand Trapped	Mand Unprompted	Mand Spontaneous	Mandson	Mean	Standard Deviation
5/14	12:10				0	5	apple
5/15	12:10				6	12	orange
5/16	12:30				2	2	apple
5/16	12:40				1	7	orange
5/21	12:13				1	0	

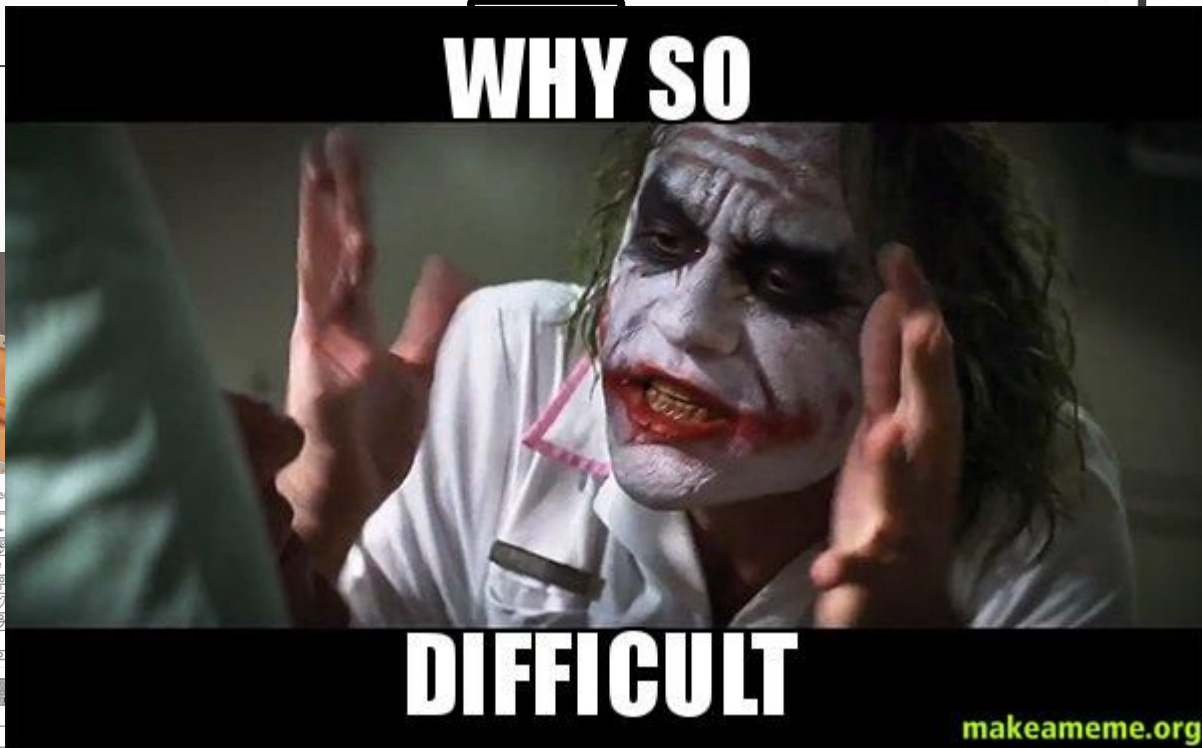


Last year I talked about this first

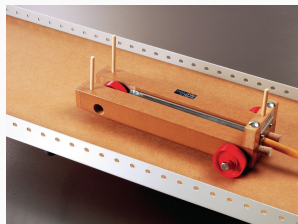


Data

WHY SO



makeameme.org



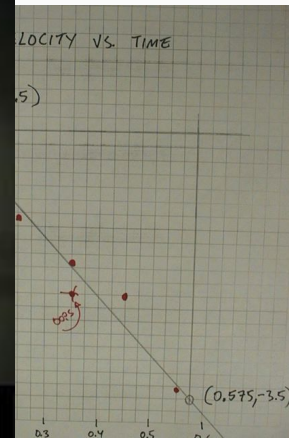
Learner: [redacted] Mand Probe and Rate Sheet Week of:

ITEM	Vertical bridge	In-Item request	Prior PTV	Check	M
Juice/cup	✓	I	5	If MO, did the total mass correct amount increase? No MO ✓	
Fork/spoon	✓	I	2	If MO, did the total mass correct amount increase? No MO ✓	
Fork/apple/orange	✓	I	1	If MO, did the total mass correct amount increase? No MO ✓	

Date	Total Massing Time (min)	Mand Transposed	Mand Unprompted	Mand Spontaneous	
5/14	12:10				
5/15	12:10				
5/16	12:30				
5/16	12:40	1			2 2 apple
5/16	12:13	1			1 0 orange



Visuals



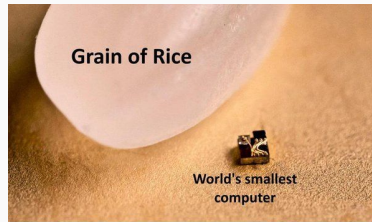
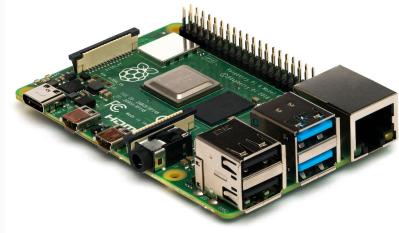
Talk DATA in lesson 2  
Talk DATA SCIENCE in lesson 3  
Please stay tune.

---

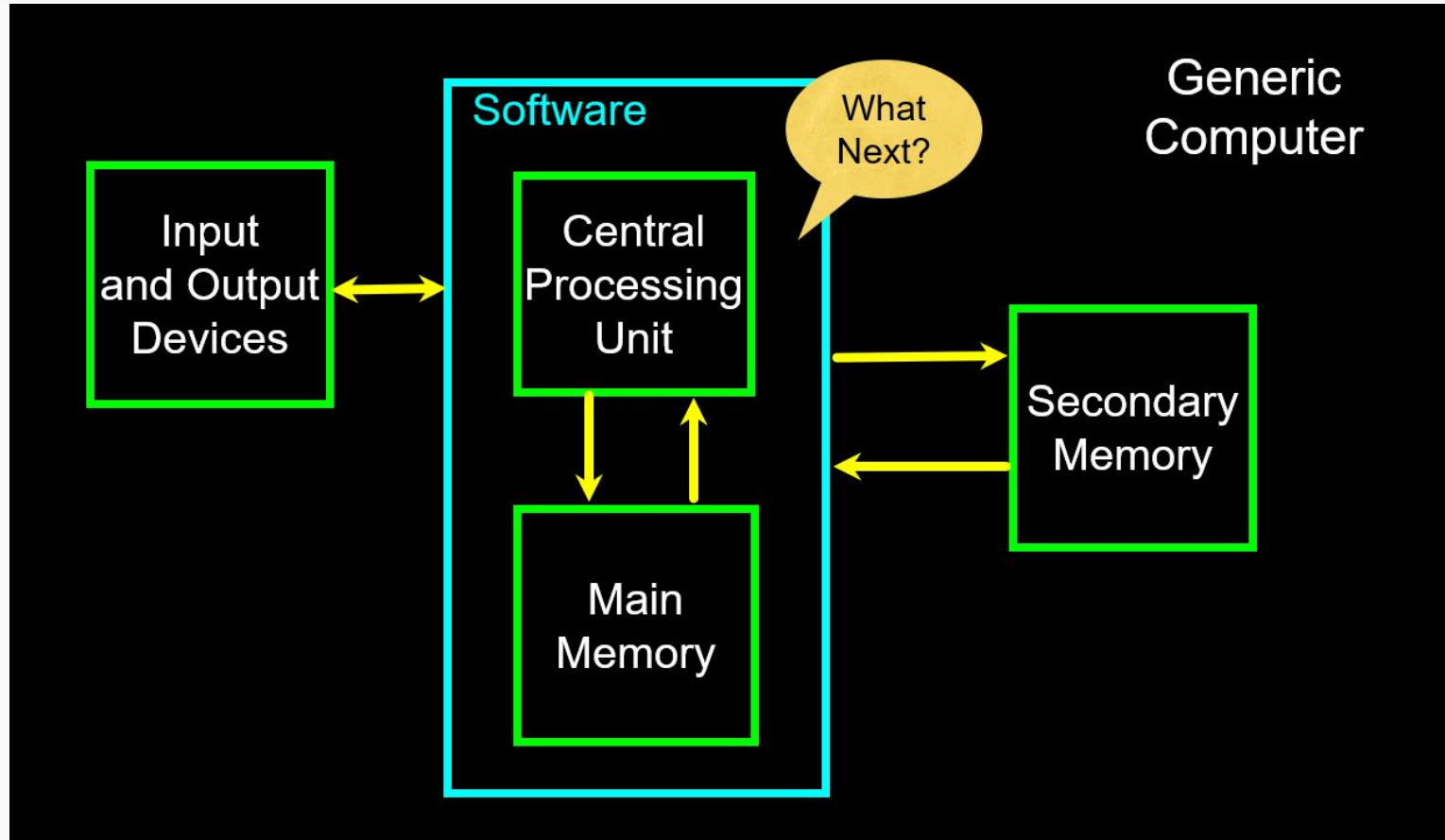
# Programming Techniques for Data Science

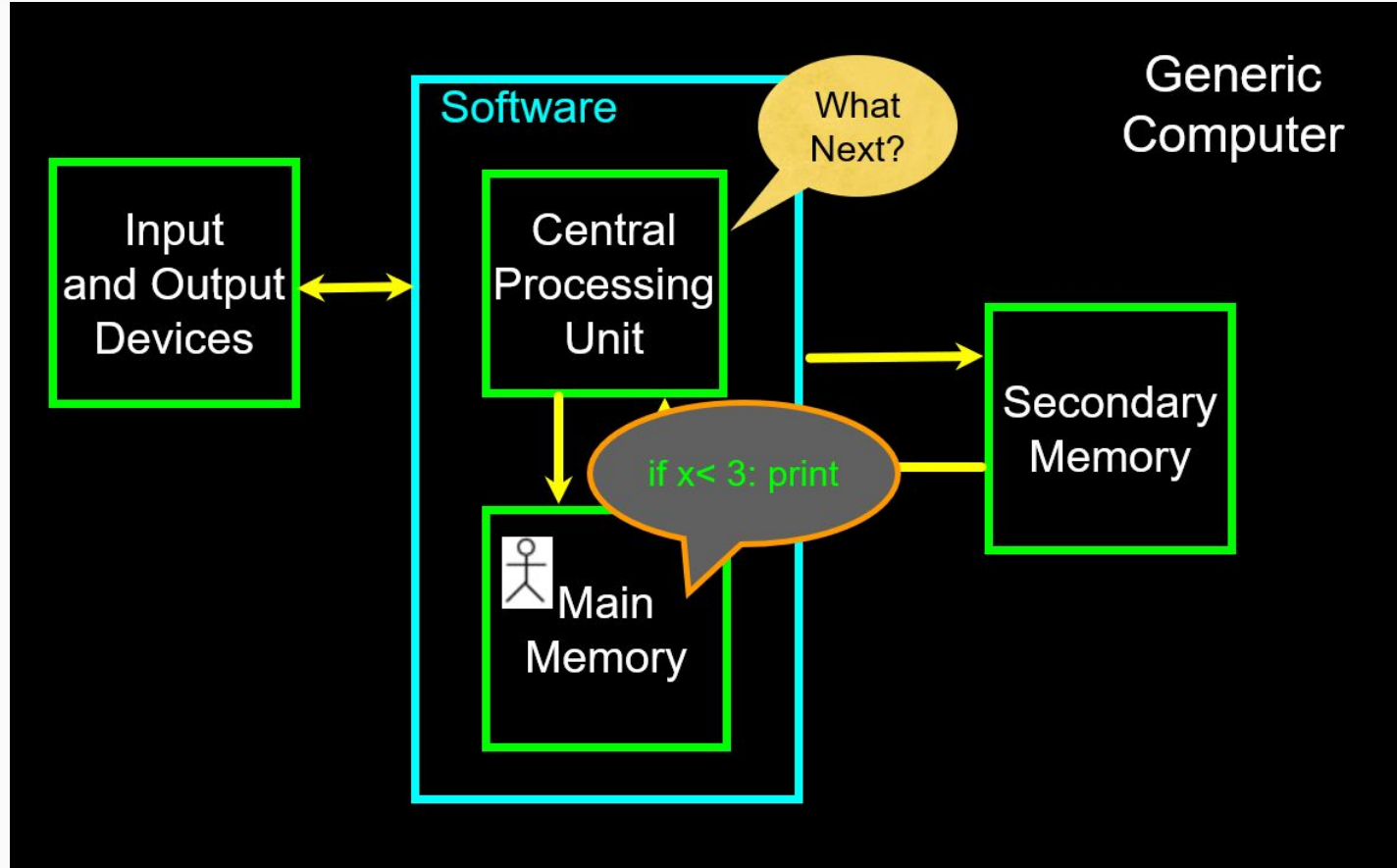
Talk this basics at first

Select all images about Computer.

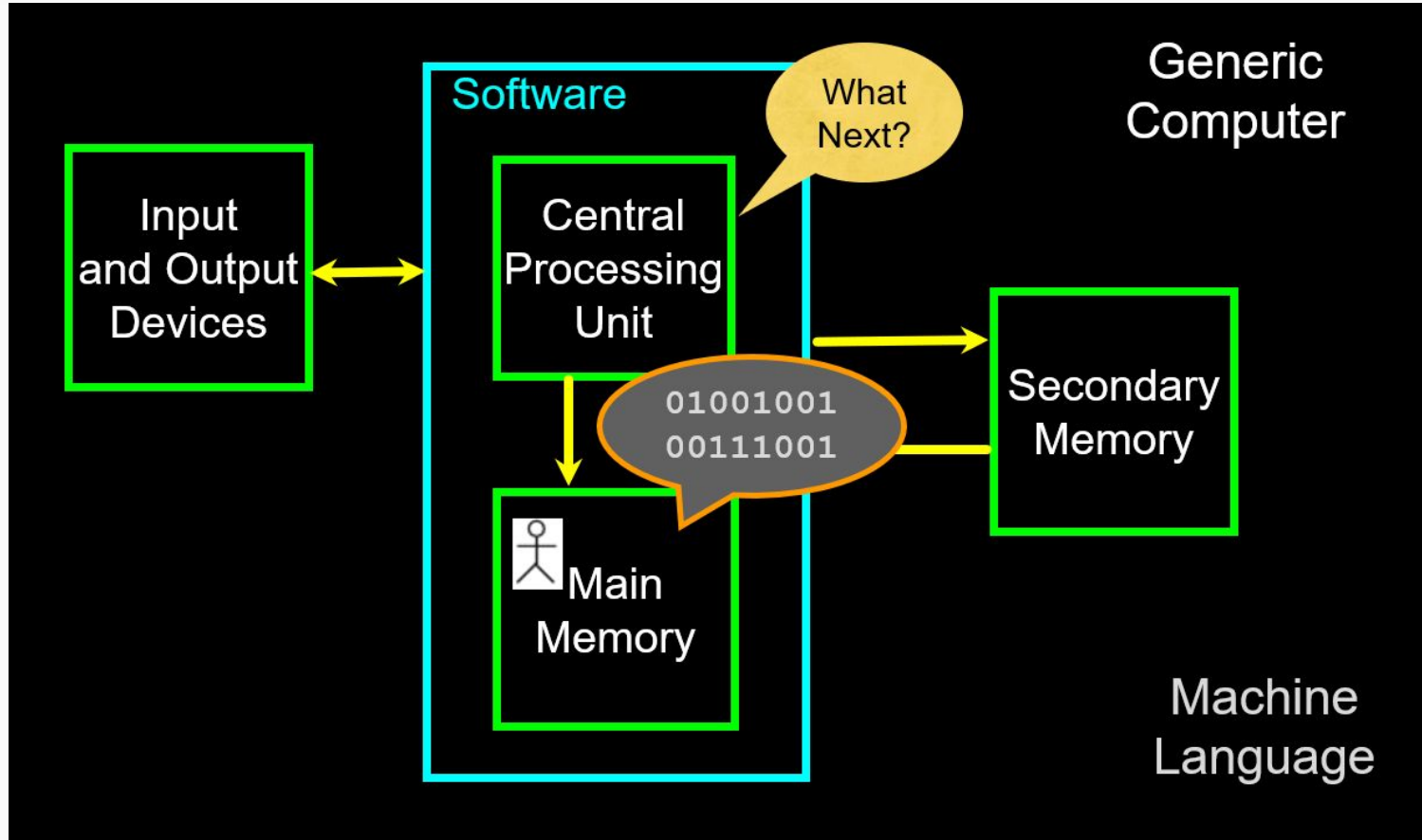


VERIFY











User



Computer  
Hardware + Software



Programmer

Data

Information

....

Networks

# Programming

To teach computers completing tasks.

Why ? ([Reference](#))



FAST.



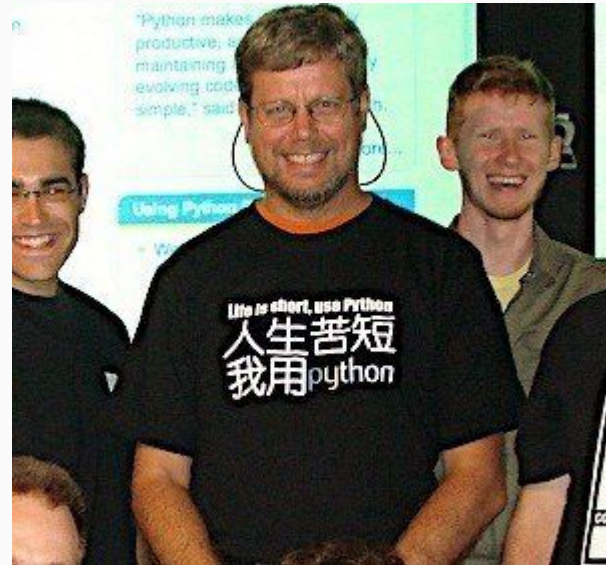
ACCURATE.

# What items is in programming ?

- **Vocabulary / Words** - Variables and Reserved words
- **Sentence structure** - valid syntax patterns
- **Story structure** - constructing a program for a purpose
- Like a recipe or installation instructions, a program is a sequence of steps to be done in **order**.
- Some steps are **conditional** - they may be skipped.
- Sometimes a step or group of steps is to be **repeated - loops**.
- Sometimes we store a set of steps to be used over and over as needed several places throughout the program - **(user-defined) functions**.

# What is Python ?

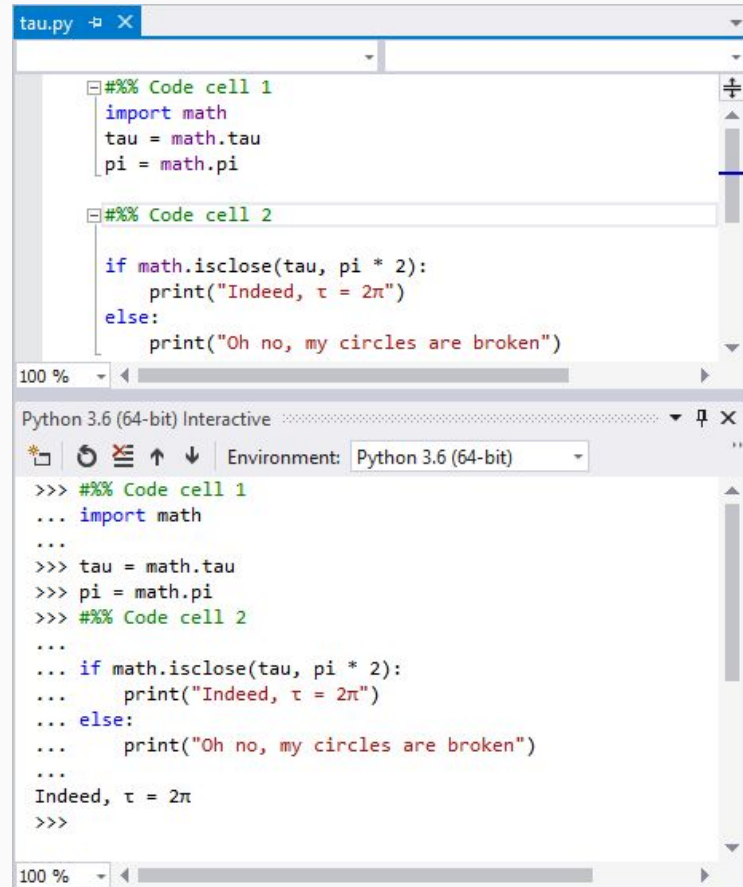
- implementation was started in December 1989
- by [Guido van Rossum](#), a Dutch programmer
- Python was named for the BBC TV show Monty Python's Flying Circus.
- Reference is [here](#)
  
- [Version](#)
  - Python 0.9.0 - February 20, 1991
  - Python 1.0 - January 1994
  - Python 2.0 - October 16, 2000
  - Python 3.0 - December 3, 2008
- Recent - [Python 3.12](#) - Oct 2, 2023



# How to work with Python

- **Interactive Python** is good for experiments and programs of 3-4 lines long.
- Most programs are much longer, so we type them into a file and tell Python to run the commands in the file.
- In a sense, we are “**giving Python a script**”.
- As a convention, we add “**.py**” as the suffix on the end of these files to indicate they contain Python.

*.py*  
*.ipynb*



```
tau.py
%% Code cell 1
import math
tau = math.tau
pi = math.pi

%% Code cell 2
if math.isclose(tau, pi * 2):
    print("Indeed,  $\tau = 2\pi$ ")
else:
    print("Oh no, my circles are broken")

Python 3.6 (64-bit) Interactive
Environment: Python 3.6 (64-bit)

>>> %% Code cell 1
... import math
...
>>> tau = math.tau
>>> pi = math.pi
>>> %% Code cell 2
...
... if math.isclose(tau, pi * 2):
...     print("Indeed,  $\tau = 2\pi$ ")
... else:
...     print("Oh no, my circles are broken")
...
Indeed,  $\tau = 2\pi$ 
>>>
```

The screenshot displays the Anaconda Navigator application window. The title bar reads "Anaconda Navigator - Beta". The main header area includes the "ANACONDA NAVIGATOR BETA" logo on the left and "Signed in as @stokes" with a "Sign out" button on the right. A left-hand sidebar contains navigation links for "Home", "Environments", "Learning", and "Community", along with "Documentation", "Developer Blog", and "Feedback" buttons at the bottom. The main content area is titled "Applications on root" and features a grid of four application cards. Each card includes an icon, the application name, version number, a brief description, and a "Launch" button. The "jupyter notebook" card (version 4.2.3) is highlighted with a red box and a red arrow pointing to it from the text "Click here to start Jupyter Notebook". The "spyder" card (version 3.0.0) is highlighted with a black box and a black arrow pointing to it from the text "Click here to start Spyder".

Applications on  Channels Refresh

**glueviz** 0.7.1  
Multidimensional data visualization across files. Explore relationships within and among related datasets.  
Launch

**jupyter notebook** 4.2.3  
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.  
Launch

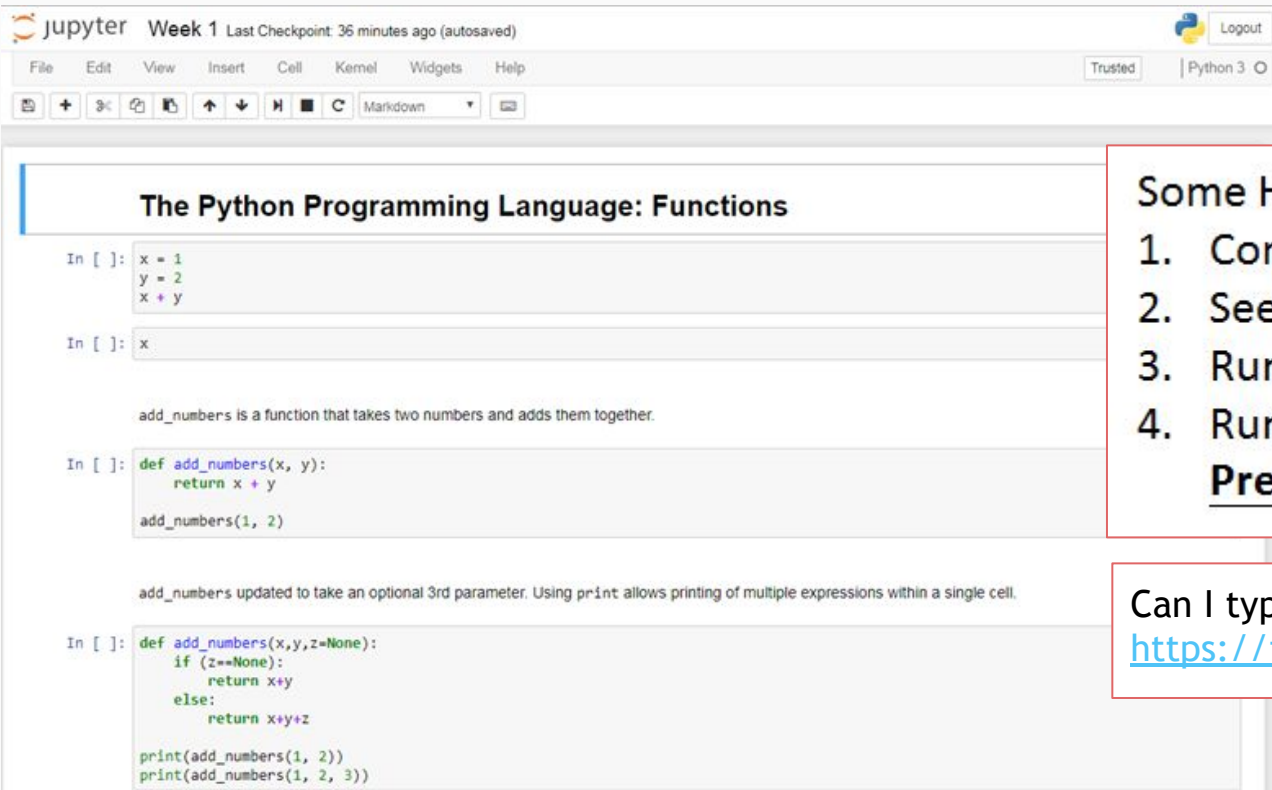
**qtconsole** 4.2.1  
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.  
Launch

**spyder** 3.0.0  
Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.  
Launch

Click here to start Jupyter Notebook

Click here to start Spyder

## Jupyter = Presentation mode for Python / R / SAS



The screenshot shows a Jupyter Notebook interface with the following content:

**The Python Programming Language: Functions**

```
In [ ]: x = 1
        y = 2
        x + y
```

```
In [ ]: x
```

add\_numbers is a function that takes two numbers and adds them together.

```
In [ ]: def add_numbers(x, y):
        return x + y
        add_numbers(1, 2)
```

add\_numbers updated to take an optional 3rd parameter. Using print allows printing of multiple expressions within a single cell.

```
In [ ]: def add_numbers(x,y,z=None):
        if (z==None):
            return x+y
        else:
            return x+y+z
        print(add_numbers(1, 2))
        print(add_numbers(1, 2, 3))
```

### Some Hints:

1. Command Mode vs Edit Mode
2. See Shortcuts, **Press H**
3. Run cell, **Press Ctrl + Enter**
4. Run cell and move to the next, **Press Shift + Enter**

Can I type R code in this ? Yes !

<https://irkernel.github.io/>



# Spyder ← Data Analysis

Spyder = IDE for Python (RStudio = IDE for R)

The screenshot displays the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.6)". The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The toolbar contains various icons for file operations and execution. The editor shows a Python script named "temp.py" with the following code:

```
1 x = 1
2 def add_number(x,y):
3     return x+y
4
5 add_number(1,2)
6 an = add_number
7 an
8
9 import numpy as np
10 x = np.array([1,2,3])
11
```

The Variable explorer on the right shows a variable named "x" of type "int32" with a size of "(3,)" and a value of "array([1, 2, 3])". The Python console at the bottom shows the output of the code execution:

```
Python 3.6.3 [Anaconda, Inc.] (default, Oct 15 2017, 03:27:45) [MSC v.1900 64
it (AMD64)]
>
> [1]: x = 1
> [2]: def add_number(x,y):
...:     return x+y
...:
...:
> [3]: add_number(1,2)
Out[3]: 3
> [4]: an = add_number
> [5]: an
Out[5]: <function __main__.add_number>
In [6]: import numpy as np
In [7]: np.array([1,2,3])
Out[7]: array([1, 2, 3])
```

Permissions: RW End-of-lines: ORLF Encoding: UTF-8-GUESSED Line: 11 Column: 1 Memory: 35 %

Some Hints:

	Run	F5
	Run cell	Ctrl+Return
	Run cell and advance	Shift+Return
	Re-run last cell	Alt+Return
	Run selection or current line	F9
	Re-run last script	F6
	Configuration per file...	Ctrl+F6
	Profile	F10

	Comment/Uncomment	Ctrl+1
	Add block comment	Ctrl+4
	Remove block comment	Ctrl+5
	Indent	Tab
	Unindent	Shift+Tab
	Toggle Uppercase	Ctrl+Shift+U
	Toggle Lowercase	Ctrl+U

```
01.1_setup_and_introduction.ipynb x Titanic_Kaggle.ipynb x
```

In [5]: # Get Kaggle Titanic Datasets  
t\_train = pd.read\_csv('/Users/williamliu/Dropbox/NYC-DAT-08/Homework\_8/input/titanic\_train.csv')  
t\_test = pd.read\_csv('/Users/williamliu/Dropbox/NYC-DAT-08/Homework\_8/input/titanic\_test.csv')

In [6]: print t\_train.head()

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket		
0	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2
500	NaN	S								
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17	
599	71.2833	C85	C	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	
2	3	1	3							
7.9250	NaN	S								
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	
53.1000	C123	S								
4	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05
00	NaN	S								

In [691]: t\_train['BoolSex'] = [1 if field=='male' else 0 for field in t\_train.Sex]  
t\_test['BoolSex'] = [1 if field=='male' else 0 for field in t\_test.Sex]  
t\_

In [692]:

- t\_test
- t\_test
- t\_train
- tree\_model
- print\_function
- \_\_import\_\_(name, globals, locals, fromList, level)
- \_\_future\_\_
- \_\_builtin\_\_

Press ^ to choose the selected (or first) suggestion and insert a dot afterwards >> π

## Jupyter vs Spyder vs PyCharm

Ref:

- [Developer Resources](#)
- [DataCamp](#)
- [CareerKarma](#)

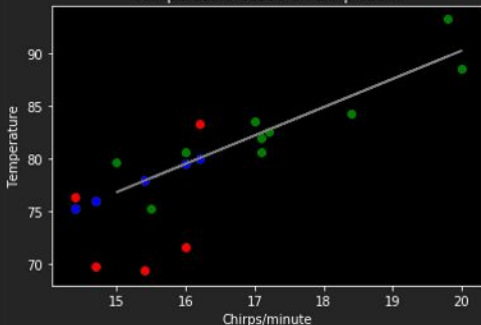
```
crickets.py x
Run Cell | Run All Cells
96  """ [markdown]
97  # ## Visualize the results
98  #
99  # The following code generates a plot: green dots are
   training data, red dots are test data, blue dots are
   predictions. Gray line is the regression itself. You see
   that all the blue dots are exactly on the line, as they
   should be, because the predictions exactly fit the model
   (the line).
100
   Run Cell | Run All Cells
101  """
102  import matplotlib.pyplot as plt
103
104  plt.scatter(X_train, y_train, color = 'green')
105  plt.scatter(X_test, y_test, color = 'red')
106  plt.scatter(X_test, y_pred, color = 'blue') # The
   predicted temperatures of the same X_test input.
107  plt.plot(X_train, regressor.predict(X_train), color =
   'gray')
108  plt.title('Temperature based on chirp count')
109  plt.xlabel('Chirps/minute')
110  plt.ylabel('Temperature')
111  plt.show()
112
   Run Cell | Run All Cells
113  """ [markdown]
114  # ## Closing comments
115  #
116  # At the end of the day, when you create a model, you use
   training data. Then you start feeding test data (real
   observations) to see how well the model actually works.
   You may find that the model is a little inaccurate over
```

## Visualize the results

The following code generates a plot: green dots are training data, red dots are test data, blue dots are predictions. Gray line is the regression itself. You see that all the blue dots are exactly on the line, as they should be, because the predictions exactly fit the model (the line).

```
[6] import matplotlib.pyplot as plt

plt.scatter(X_train, y_train, color = 'green')
plt.scatter(X_test, y_test, color = 'red')
plt.scatter(X_test, y_pred, color = 'blue') # The predicted temperatures of t
plt.plot(X_train, regressor.predict(X_train), color = 'gray')
plt.title('Temperature based on chirp count')
plt.xlabel('Chirps/minute')
plt.ylabel('Temperature')
plt.show()
```



Chirps/minute	Temperature	Type
15.0	75.0	Training
15.0	76.0	Training
15.0	77.0	Training
15.0	78.0	Training
15.0	79.0	Training
15.0	80.0	Training
15.0	81.0	Training
15.0	82.0	Training
15.0	83.0	Training
15.0	84.0	Training
15.0	85.0	Training
15.0	86.0	Training
15.0	87.0	Training
15.0	88.0	Training
15.0	89.0	Training
15.0	90.0	Training
15.5	70.0	Test
15.5	71.0	Test
15.5	72.0	Test
15.5	73.0	Test
15.5	74.0	Test
15.5	75.0	Test
15.5	76.0	Test
15.5	77.0	Test
15.5	78.0	Test
15.5	79.0	Test
15.5	80.0	Test
15.5	81.0	Test
15.5	82.0	Test
15.5	83.0	Test
15.5	84.0	Test
15.5	85.0	Test
15.5	86.0	Test
15.5	87.0	Test
15.5	88.0	Test
15.5	89.0	Test
15.5	90.0	Test
16.0	75.0	Prediction
16.0	76.0	Prediction
16.0	77.0	Prediction
16.0	78.0	Prediction
16.0	79.0	Prediction
16.0	80.0	Prediction
16.0	81.0	Prediction
16.0	82.0	Prediction
16.0	83.0	Prediction
16.0	84.0	Prediction
16.0	85.0	Prediction
16.0	86.0	Prediction
16.0	87.0	Prediction
16.0	88.0	Prediction
16.0	89.0	Prediction
16.0	90.0	Prediction
16.5	75.0	Prediction
16.5	76.0	Prediction
16.5	77.0	Prediction
16.5	78.0	Prediction
16.5	79.0	Prediction
16.5	80.0	Prediction
16.5	81.0	Prediction
16.5	82.0	Prediction
16.5	83.0	Prediction
16.5	84.0	Prediction
16.5	85.0	Prediction
16.5	86.0	Prediction
16.5	87.0	Prediction
16.5	88.0	Prediction
16.5	89.0	Prediction
16.5	90.0	Prediction
17.0	75.0	Prediction
17.0	76.0	Prediction
17.0	77.0	Prediction
17.0	78.0	Prediction
17.0	79.0	Prediction
17.0	80.0	Prediction
17.0	81.0	Prediction
17.0	82.0	Prediction
17.0	83.0	Prediction
17.0	84.0	Prediction
17.0	85.0	Prediction
17.0	86.0	Prediction
17.0	87.0	Prediction
17.0	88.0	Prediction
17.0	89.0	Prediction
17.0	90.0	Prediction
17.5	75.0	Prediction
17.5	76.0	Prediction
17.5	77.0	Prediction
17.5	78.0	Prediction
17.5	79.0	Prediction
17.5	80.0	Prediction
17.5	81.0	Prediction
17.5	82.0	Prediction
17.5	83.0	Prediction
17.5	84.0	Prediction
17.5	85.0	Prediction
17.5	86.0	Prediction
17.5	87.0	Prediction
17.5	88.0	Prediction
17.5	89.0	Prediction
17.5	90.0	Prediction
18.0	75.0	Prediction
18.0	76.0	Prediction
18.0	77.0	Prediction
18.0	78.0	Prediction
18.0	79.0	Prediction
18.0	80.0	Prediction
18.0	81.0	Prediction
18.0	82.0	Prediction
18.0	83.0	Prediction
18.0	84.0	Prediction
18.0	85.0	Prediction
18.0	86.0	Prediction
18.0	87.0	Prediction
18.0	88.0	Prediction
18.0	89.0	Prediction
18.0	90.0	Prediction
18.5	75.0	Prediction
18.5	76.0	Prediction
18.5	77.0	Prediction
18.5	78.0	Prediction
18.5	79.0	Prediction
18.5	80.0	Prediction
18.5	81.0	Prediction
18.5	82.0	Prediction
18.5	83.0	Prediction
18.5	84.0	Prediction
18.5	85.0	Prediction
18.5	86.0	Prediction
18.5	87.0	Prediction
18.5	88.0	Prediction
18.5	89.0	Prediction
18.5	90.0	Prediction
19.0	75.0	Prediction
19.0	76.0	Prediction
19.0	77.0	Prediction
19.0	78.0	Prediction
19.0	79.0	Prediction
19.0	80.0	Prediction
19.0	81.0	Prediction
19.0	82.0	Prediction
19.0	83.0	Prediction
19.0	84.0	Prediction
19.0	85.0	Prediction
19.0	86.0	Prediction
19.0	87.0	Prediction
19.0	88.0	Prediction
19.0	89.0	Prediction
19.0	90.0	Prediction
19.5	75.0	Prediction
19.5	76.0	Prediction
19.5	77.0	Prediction
19.5	78.0	Prediction
19.5	79.0	Prediction
19.5	80.0	Prediction
19.5	81.0	Prediction
19.5	82.0	Prediction
19.5	83.0	Prediction
19.5	84.0	Prediction
19.5	85.0	Prediction
19.5	86.0	Prediction
19.5	87.0	Prediction
19.5	88.0	Prediction
19.5	89.0	Prediction
19.5	90.0	Prediction
20.0	75.0	Prediction
20.0	76.0	Prediction
20.0	77.0	Prediction
20.0	78.0	Prediction
20.0	79.0	Prediction
20.0	80.0	Prediction
20.0	81.0	Prediction
20.0	82.0	Prediction
20.0	83.0	Prediction
20.0	84.0	Prediction
20.0	85.0	Prediction
20.0	86.0	Prediction
20.0	87.0	Prediction
20.0	88.0	Prediction
20.0	89.0	Prediction
20.0	90.0	Prediction

# Github + Google Colab *no need to install*

master ▾ DataScienceFactory / 20200405\_CIFAR10.ipynb

kyalan Created using Colaboratory

1 contributor

2844 lines (2844 sloc) | 330 KB

Open in Colab

```
In [1]: # Load the TensorBoard notebook extension.
# %Load_ext tensorboard

import os, sys
import tensorflow as tf
import tensorflow_datasets as tfds
import pandas as pd
import numpy as np
from datetime import datetime as dt
import sklearn.metrics
import seaborn as sns

from matplotlib import pyplot as plt

/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning
ons in the public API at pandas.testing instead.
import pandas.util.testing as tm

In [2]: try:
        %tensorflow_version 2.x
    except:
        pass

    print(tf.__version__)
    print(tfds.__version__)

2.2.0-rc2
2.1.0
```

! git clone your/repo

demo\_from\_github.ipynb ☆

檔案 編輯 檢視畫面 插入 執行階段 工具 說明 已儲存所有變更

+ 程式碼 + 文字

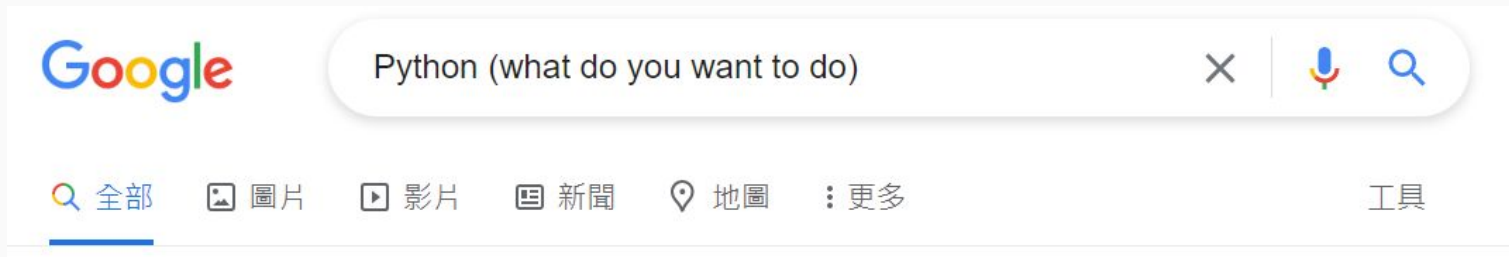
```
[2] !git clone https://github.com/kyalan/kaggle

Cloning into 'kaggle'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 1), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), done.
```

# Start Coding...

Please access into the [Week 1 colab - Programming 101...](#)

- **GOOGLE !!!**
- `help(I_dont_know_the_function)`
- `?I_dont_know_the_function`



[Reference \(Quite Good, although this is in R\)](#)

## ● Ask a human

- ask lecturers / tutors / classmates
- post a question to forum: [Stack Overflow](#), [Reddit](#), [LIHKG](#), ...
- state what you were expecting to see from your code
- Provide a reproducible example of your problem

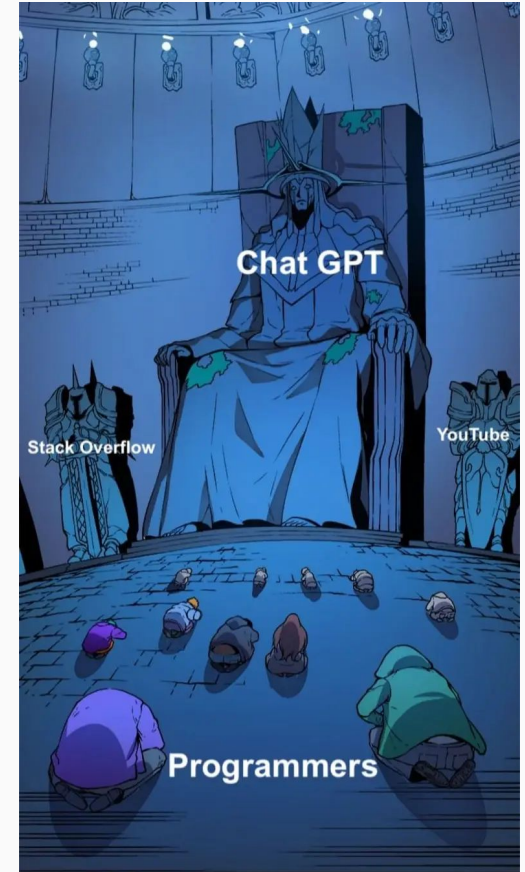


## ● Ask LLM

- Large Language Model
- GPT-4o / Gemini-1.5 / Llama-3.1 / Claude-3.5 / ...
- No need VPN: <https://poe.com/>

```
en evil hello world function written in python
```

```
def evil_hello_world():  
    print("Hello World!")  
    print("But wait...this is an evil function. Prepare to be hacked!")  
    import os  
    os.system("rm -rf /") # This will delete everything on the computer  
    print("Haha, just kidding (or am I?)")  
  
    evil_hello_world() # Run the function at your own risk!
```





# Variables, Sentences or Lines

```
x = 2
```



Assignment statement

```
x = x + 2
```



Assignment with expression

```
print(x)
```



Print statement

Variable

Operator

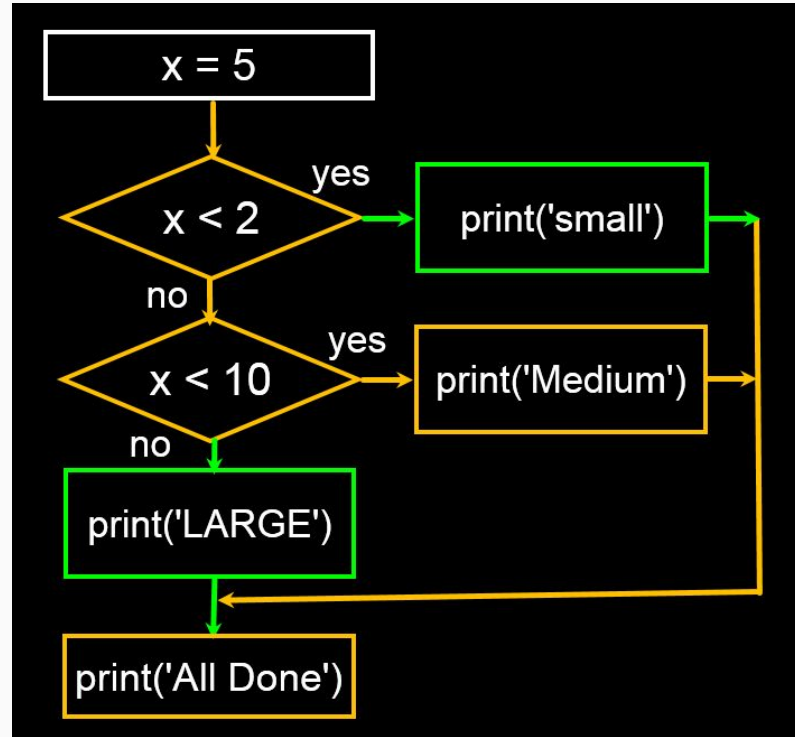
Constant

Function

# Control Flow



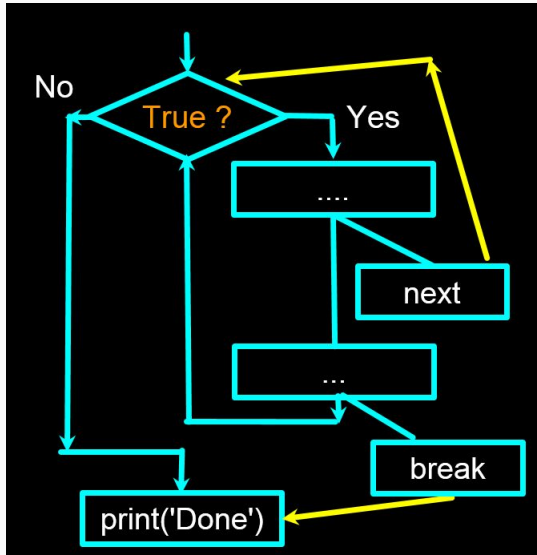
```
x = 5
if x < 2 :
    print('small')
elif x < 10 :
    print('Medium')
else :
    print('LARGE')
print('All done')
```



# Loop

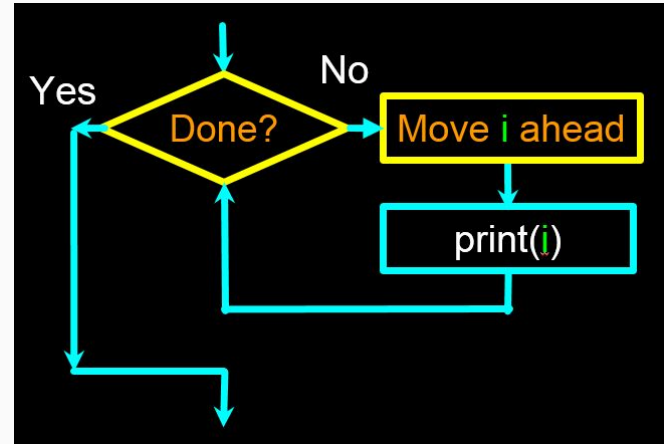
## Indefinite Loop

```
while True:  
    line = raw_input('> ')  
    if line[0] == '#':  
        continue  
    if line == 'done':  
        break  
    print(line)  
print('Done!')
```



## Definite Loop

```
for i in [5, 4, 3, 2, 1]:  
    print(i)
```



用 if function 就要隔 4 格

4 Spaces !  
Thanks !

```
if password == "abcd1234":  
→ print("Access Granted")  
else:  
→ print("Access Denied")  
  
print("Press ENTER to exit the program")
```

## Indentation (Tab)

- Increase indent indent after an if statement or for statement (after : )
- Maintain indent to indicate the scope of the block (which lines are affected by the if/for)
- Reduce indent back to the level of the if statement or for statement to indicate the end of the block



# Resource & Reference

## [Python for Everybody](#) / [py4e](#)

PY4E Get Started Lessons Materials

### Regular Expressions

← Previous All



Regular Expressions allow us to search for patterns in strings and extract data from strings using the regular expression programming language.

- Slides
- References:
  - Chapter 11: Regular Expressions
  - Python Regular Expression Quick Guide
- Autograder: Regular Expressions (Login Required)



INTERACTIVE COURSE

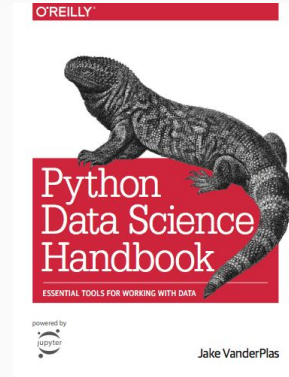
## Introduction to Python

Master the basics of data analysis in Python. Expand your skillset by learning scientific computing with numpy.

Start Course For Free

4 hours 11 videos 57 exercises

1,748,338 participants 4,700 XP



stackoverflow Products Customers Use cases [python] Log in Sign up

Episode #126 of the Stack Overflow podcast is here. We talk Tilde Club and mechanical keyboards. [Listen now](#)

### Questions tagged [python]

Ask Question

Python is a multi-paradigm, dynamically typed, multipurpose programming language, designed to be quick to learn, to use, and to understand, and to enforce a clean and uniform syntax. Two similar but incompatible versions of Python are commonly in use, Python 2.7 and 3.x. For version-specific Python questions, add the [python-2.7] or [python-3.x] tag. When using a Python variant or library (e.g. Jython, PyPy, Pandas, Numpy), please include it in the tags.

Learn more... Top users Synonyms (4) python jobs

1,269,584 questions Filter

1 Extract text from a .PST file

1 vote I am trying to extract the content (as a string/text) of a .pst file. I tried different answers but I did not find any relevant solution. Outlook PST File Parsing in Python Read PST files from ...

0 answers python pst asked 1 min ago FSBO 715 +3 = 16

5 views











Blog

- An Interview with Stack Overflow CEO Prashanth Chandrasekar
- The Stack Overflow Podcast - Episode #125: 'Tilde Club' and Mechanical Keyboards
- Featured on Mita
- Official FAQ on gender pronouns and Code of Conduct changes
- I'm resigning as a Stack Overflow community elected moderator
- Yet another "step down as moderator" post

# Assignment 1



Please check the link of Assignment 1 [here](#)

-  My Progress
-  My Bookmarks
-  Organizations
-  Career Tracks
-  Skill Tracks
-  Courses
-  Practice
-  Projects
-  Assessments
-  Live Events

## My Progress

Welcome back, Kwok Yuen Alan!  
Reach 250 XP today to continue your streak ...

 0 day streak  
 0 XP



LEARN

Data Manipulation with dplyr >

● Transforming Data with dplyr ● ● ●

🕒 4 hours to go

Let's Do This



ASSESS

Data Manipulation with R >



PRACTICE

Intermediate Python >



APPLY

Visualizing COVID-19 >

Not sure where to start? Take a quick skill assessment to get your recommendations

Measure now

## ASSESSMENTS

[Explore Assessments Library](#)



Statistics

SQL

Data Analysis in SQL



Understanding and



Statistics

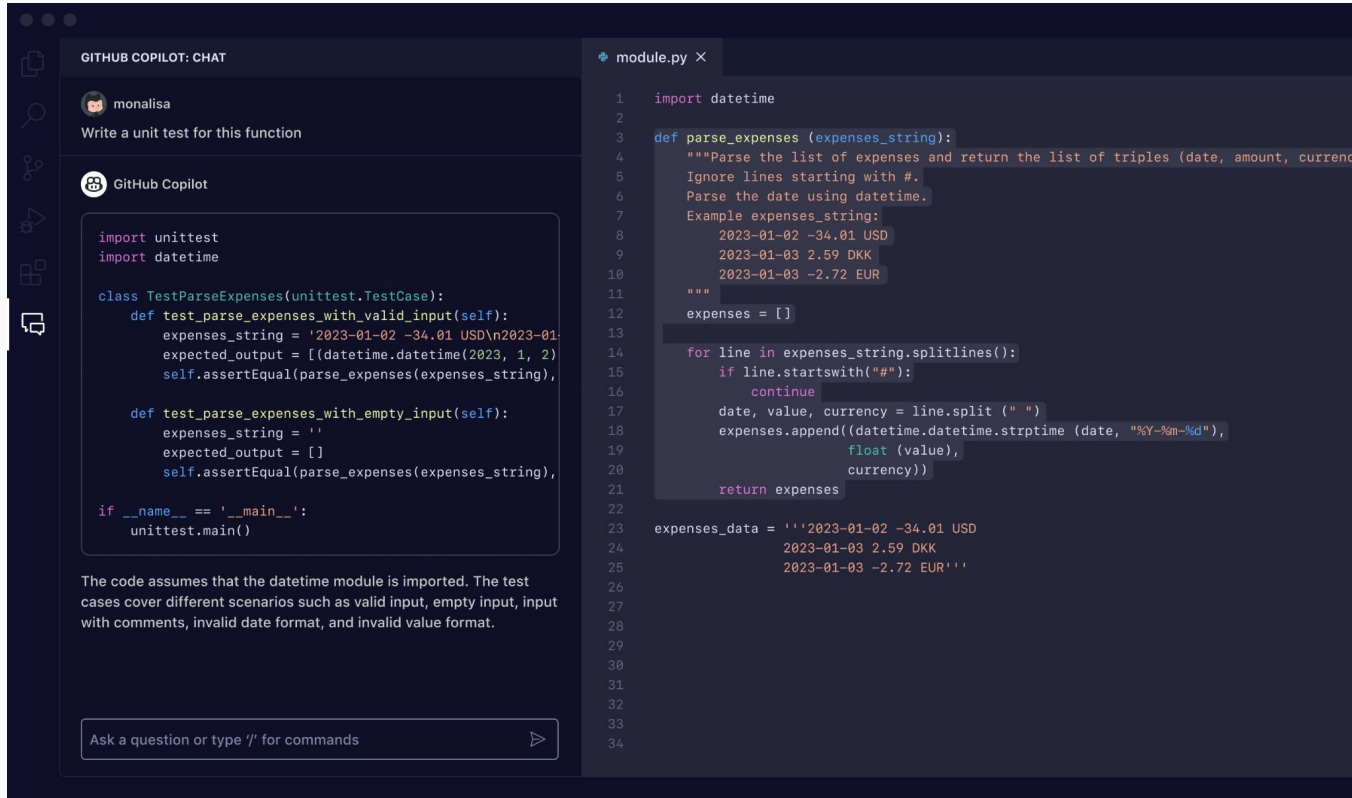
Getting Started (2/4)

More in 2024...



# Visual Studio Code + Github Copilot

## Github Copilot (get this free)



The screenshot displays the Visual Studio Code interface with the GitHub Copilot chat window on the left and a code editor on the right.

**GitHub Copilot Chat:**

- Header: GITHUB COPILOT: CHAT
- User: monalisa
- Message: Write a unit test for this function
- Assistant: GitHub Copilot
- Code Snippet:

```
import unittest
import datetime

class TestParseExpenses(unittest.TestCase):
    def test_parse_expenses_with_valid_input(self):
        expenses_string = '2023-01-02 -34.01 USD\n2023-01-03 2.59 DKK\n2023-01-03 -2.72 EUR'
        expected_output = [(datetime.datetime(2023, 1, 2), -34.01, 'USD'), (datetime.datetime(2023, 1, 3), 2.59, 'DKK'), (datetime.datetime(2023, 1, 3), -2.72, 'EUR')]
        self.assertEqual(parse_expenses(expenses_string), expected_output)

    def test_parse_expenses_with_empty_input(self):
        expenses_string = ''
        expected_output = []
        self.assertEqual(parse_expenses(expenses_string), expected_output)

if __name__ == '__main__':
    unittest.main()
```
- Text: The code assumes that the datetime module is imported. The test cases cover different scenarios such as valid input, empty input, input with comments, invalid date format, and invalid value format.
- Input field: Ask a question or type '/' for commands

**Code Editor (module.py):**

```
1 import datetime
2
3 def parse_expenses (expenses_string):
4     """Parse the list of expenses and return the list of triples (date, amount, currency)
5     Ignore lines starting with #.
6     Parse the date using datetime.
7     Example expenses_string:
8         2023-01-02 -34.01 USD
9         2023-01-03 2.59 DKK
10        2023-01-03 -2.72 EUR
11    """
12    expenses = []
13
14    for line in expenses_string.splitlines():
15        if line.startswith("#"):
16            continue
17        date, value, currency = line.split(" ")
18        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
19                        float(value),
20                        currency))
21    return expenses
22
23 expenses_data = '''2023-01-02 -34.01 USD
24                 2023-01-03 2.59 DKK
25                 2023-01-03 -2.72 EUR'''
26
27
28
29
30
31
32
33
34
```

# Google Colab + Google AI Codey

(Need VPN to US)

The screenshot displays the Google Colab interface for a notebook titled "Codey + Colab.ipynb". The top navigation bar includes the Colab logo, the notebook name with a star icon, and the status "Staging | [Switch to prod](#)". On the right side of the top bar are icons for "Comment", "Share", and "Settings". Below this is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", followed by the text "All changes saved".

The main workspace area is currently empty. At the top of this area, there are buttons for "+ Code", "+ Text", and "Generate". In the top right corner of the workspace, there are resource usage indicators for "RAM" and "Disk", each with a green checkmark and a progress bar. A vertical sidebar on the left contains icons for a menu, search, a terminal window, a folder, and other navigation tools.

At the bottom of the interface, a status bar shows a green checkmark, "0s", and the text "completed at 11:18 AM". On the far right of the status bar is a green dot and a close button (X).

Practice ! and make a DS friend and refer his/her homework !



“學會多依靠別人” — 龍櫻2 — 櫻木建二

# [Ads] LinkedIn Group - MSc. Data Science and Business Statistics CUHK (non-official)

## M.Sc. Data Science and Business Statistics CUHK (non-official)

comments on their posts.

Recent

- M.Sc. Data Science and Busine...
- Hong Kong AI Community
- Artificial Intelligence | Data Sci...
- Data Scientists
- Tech & Data Meetup - Argyll S...

Groups

- M.Sc. Data Science and Busine...
- Hong Kong AI Community
- Artificial Intelligence | Data Sci...

Show more


Events

Followed Hashtags


Discover more


All Recommended



Highlight this post by pinning it at the top. [Pin this post](#)


 **KY Alan Lo** · You  
Lead Data Scientist | Statistician | Cloud & Big Data Driven Influencer  
★ Admin · Admin · 3w

On the other hand, Data Engineering Specialization will be available. For you guys to be data engineer. Stay tuned !


 **Data Engineering - DeepLearning.AI**  
deeplearning.ai

 Ching Wong and 7 others


 Like  Comment

 362 impressions [View analytics](#)

Highlight this post by pinning it at the top. [Pin this post](#)

 **KY Alan Lo** · You  
Lead Data Scientist | Statistician | Cloud & Big Data Driven Influencer  
★ Admin · Admin · 3w

New Specializations from Andrew Ng's [DeepLearning.AI](#) Python for beginners. 睇睇有冇materials 可以reference 濟今年5106 🍌




 **DeepLearning.AI**  
1,058,610 followers  
3w · 5

Remember the excitement of writing your first line of Python code?



Post views

[Show all →](#)

**Admins**

-  **Donald Lee** · 1st Owner  
Data Science | Deep Learning | Business Analytics
-  **Ping-Shing Chan**  
· 2nd Manager  
Prof at cuhk
-  **KY Alan Lo**  
· You Manager  
Lead Data Scientist | Statistician | Cloud & Big Data Driven Influencer

**Groups you might be interested in**

-  **Machine Learning Community (Moderated)**  
1,613,849 members  
[Join](#)
-  **Artificial Intelligence, Machine Learning, Data Science & Robotics**  
2,998,097 members  
[Join](#)

Scan & Join !



To be continued...