

# Section 0 Software Preparation

## 1. (Optional) Basics of Git

[Github](#) is perhaps the world largest open source repository to store, share and collaborate on codes (now purchased by Microsoft), and is based on [Git](#)-- a version control tool. For experienced programmers, of course command line is the most popular tool, while for beginners, [Github Desktop](#), is a good choice to start with and get familiar with the concepts.

Basic concepts of git:

- **Clone**: get a local copy of the folder (repository) from the cloud
- **Fetch and Pull**: download the updated repository from cloud and merge with the current local copy
- **Commit and Push**: upload the local changes to the cloud and merge with the repository there

## 2. Jupyter Notebook

The minimal element of notebook file (or building blocks) are the **cells**. For each cell, it has two **types**:

- **Markdown**: write the text, equations ...
- **Code**: write and excute the python code

For both cells, you need to run the cell to evaluate (either display or run the code)

This is the markdown type: **Hello World!** *Hello World!*

### This is another section (header level 3)

inline equations with latex:  $a^2 + b^2 = c^2$

separate-line equation:

$$a^2 + b^2 = c^2$$

In [3]:

```
# this is the code type, I am the comment in Python followed by ## signs in the code cel

print('Hello World!') # in Python, both "" and '' are for strings
print("hello")
# to run this cell, press control+ enter (cursor in current cell after running)
# or shift+ enter (cursor in next cell after running)
```

```
Hello World!
hello
```

For each cell, it also has two **modes**:

- Command mode: The edge color is blue. To enter this mode, press `Esc` on keyboard. This mode is for changing the property or global control of the cell (for example, insert more cells, change cell type from markdown to code)
- Edit mode: The edge color is green, and there is a pencil sign in the up-right corner of the menu bar. To enter this mode, press `Enter` or double-click (for markdown cells). This mode is for locally edit the content of the cell.

Useful keyboard shortcuts (of course you can always click the menu bar in the top) in **command mode** (Press `Esc` to enter):

- change from markdown to code: `Y`
- change from code to markdown: `M`
- insert cell above: `A`
- insert cell below: `B`
- delete cell: type `D` consecutively for two times, i.e. `DD`
- undo delete cell: `Z`
- move up and down: up and down in keyboard

In [ ]: `# this cell is for you to practice`

In [14]: `5+3`

Out[14]: 8

In [5]: `5-3`

Out[5]: 2

In [6]: `5*3 # multiplication`

Out[6]: 15

In [7]: `5/3 # float division`

Out[7]: 1.6666666666666667

In [8]: `5//3 # integer division`

Out[8]: 1

In [9]: `5%3 # modulo`

Out[9]: 2

```
In [10]: 5**3 # important ! this is exponential in python -- double star **
```

```
Out[10]: 125
```

```
In [11]: 5^3 # this is the bitwise operator in Python -- first difference you come across with m
```

```
Out[11]: 6
```

```
In [15]: abs(-0.8) # built-in function -- absolute value
```

```
Out[15]: 0.8
```

```
In [16]: help(abs) # help to see the function documentations
```

Help on built-in function abs in module builtins:

```
abs(x, /)
    Return the absolute value of the argument.
```

```
In [17]: min(3,2) # built-in function -- minimal number
```

```
Out[17]: 2
```

```
In [18]: sqrt(5) # what?
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-18-3d21b68d484f> in <module>
----> 1 sqrt(5) # what?

NameError: name 'sqrt' is not defined
```

```
In [19]: cos(0) # what???
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-19-6b1b2f32141b> in <module>
----> 1 cos(0) # what???

NameError: name 'cos' is not defined
```

```
In [20]: import math # the functions are in the math module of python! will introduce in later l
math.sqrt(5)
```

```
Out[20]: 2.23606797749979
```

```
In [21]: math.cos(0)
```

```
Out[21]: 1.0
```

```
In [22]: 'hello'+' world' # concatenation of strings
```

```
Out[22]: 'hello world'
```

```
In [23]: "hello "*3 # repeat of strings
```

```
Out[23]: 'hello hello hello '
```

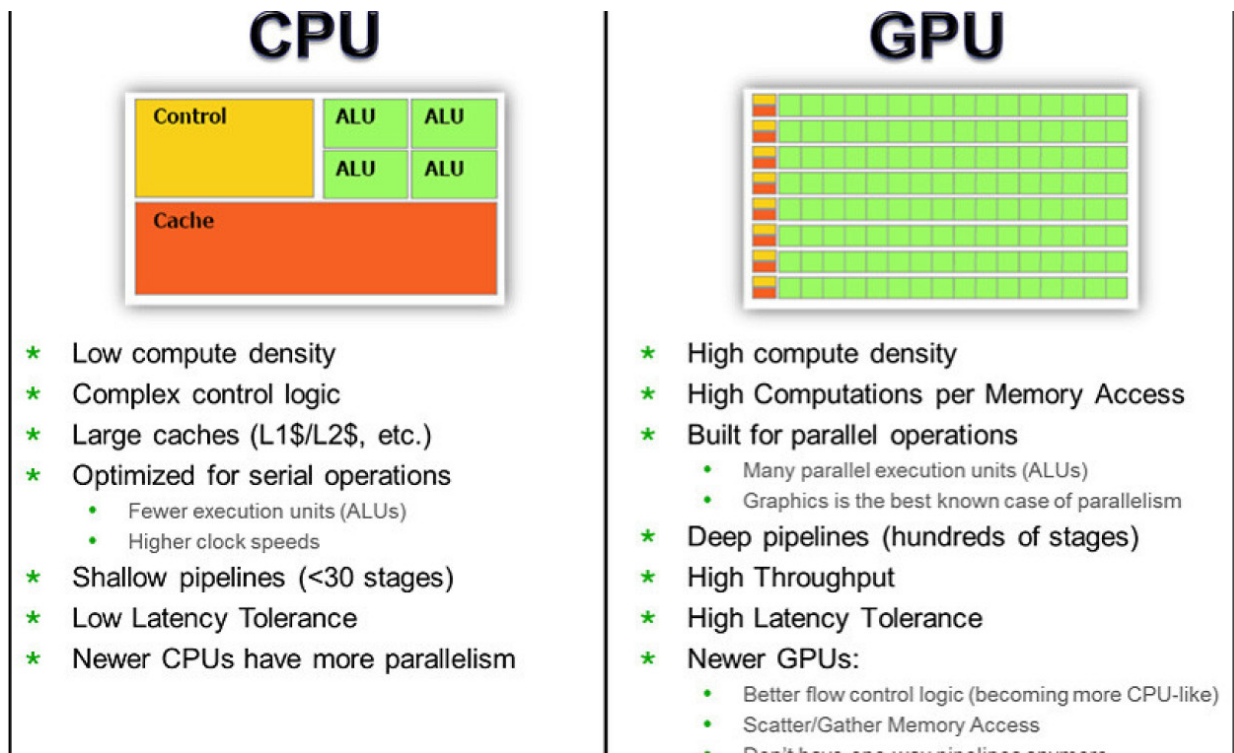
Further Readings :

- [Jupyter Notebook Shorcuts](#)
- [Markdown syntax](#)
- [Latex syntax](#)

### 3. Basics of CPU and GPU

- **CPU**: Central Processing Unit, good at complex operations, small number of processor cores.  
*Famous Manufacturer*: Intel and AMD
- **GPU**: Graphics Processing Unit, good at simple operations, huge amount of processor cores.  
*Famous Manufacturer*: Intel (integrated cards), Nvidia and AMD (discrete cards). In machine learning application, the most popular cards are from Nvidia, partly because of its support for **CUDA** (programming interface to write codes with GPU).

**Intuitive Understanding**: CPU is like one college student who can do calculus, while GPU is like 100 elementary school students who only know basic arithmetics.



It happens that GPU is very suitable for machine learning (especially deep learning) tasks (easily speed up calculation by 10-100 times than CPU), and the fast development of GPU is one of the key

contributor of machine learning popularity. See more introductions [here](#).

Nowadays, beyond CPU and GPU, AI [ASIC](#) (Application-specific integrated circuit) is becoming more and more popular. One of the most well-known example to general public is Google's [TPU](#) (Tensor Processing Unit), which is used in Alpha GO. Until today, it is still a "secret weapon" of Google, and you can use TPU from the google's cloud service, which can be [10 times more expensive than GPU](#). (Google Colab provides some out-dated TPU for free trial).

## 4. (Optional) [Google Colab](#) and [Kaggle Notebook](#)

Both provide the free CPU and GPU resources and the notebook enviroment in the cloud.