

# Software tutorial

# VSCode

- Install Visual Studio Code
  - Explain the GUI of VSCode
  - Show example of Markdown visualization
  - Show example of installing VSCode extensions

# VSCode MarkDown (.md)





Visual Studio Code





EXTENSIONS: MARKETPLACE


markdown


 **Markdown All in One** 2.7.0 1.1M ★ 5  
All you need to write Markdown (keybo...  
Yu Zhang


 **Markdown** 1.1.3 10K ★ 3  
Markdown Paste Image To Qiniu OSS st...  
starkwang


 **Markdown Preview ...** 0.5.2 765K ★ 4.5  
Markdown Preview Enhanced ported to...  
Yiyi Wang


 **markdownlint** 0.34.0 1.4M ★ 4.5  
Markdown linting and style checking fo...  
David Anson

 **Markdown PDF** 1.4.1 406K ★ 4.5  
Convert Markdown to PDF  
yzane

 **Markdown Shortcuts** 0.12.0 70K ★ 5  
Shortcuts for Markdown editing  
mdickin


 **docs-markdown** 0.2.44 36K ★ 5  
Docs Markdown Extension  
Microsoft

 **Markdown Preview ...** 0.1.6 178K ★ 4.5  
Changes VS Code's built-in markdown ...  
Matt Bierner

 **Instant Markdown** 1.4.4 109K ★ 3.5  
Instant previews of your markdown files...  
David Bankier

homework\_00.md

Extension: Markdown Preview Enhanced ×



## Markdown Preview Enhanced

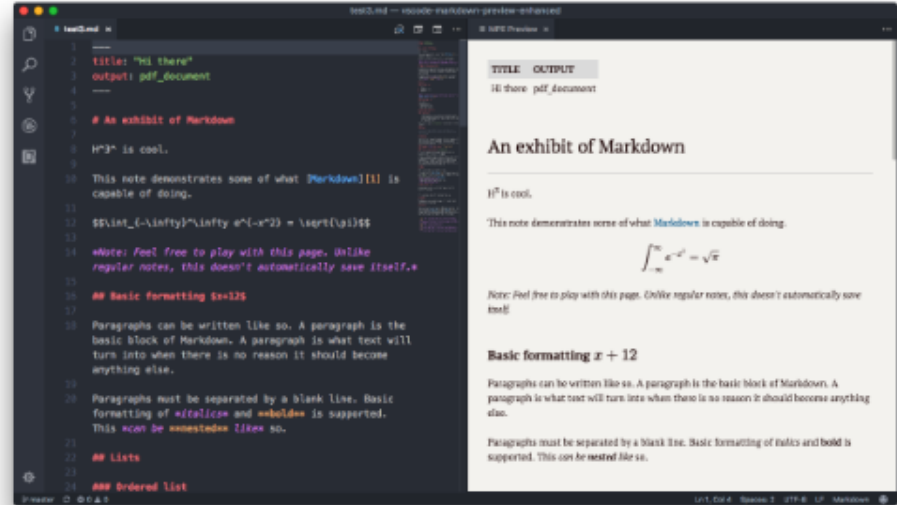
Yiyi Wang | 765,699 | ★★★★★ | Rep...

Markdown Preview Enhanced ported to vscode

[Disable](#) [Uninstall](#) This extension is enabled globally.

[Details](#) [Contributions](#) [Changelog](#)

## Markdown Preview Enhanced



# Anaconda Python Environments

The screenshot displays the Anaconda Navigator desktop application. The interface includes a top menu bar with 'File' and 'Help', a central header with the 'ANACONDA NAVIGATOR' logo and a 'Sign in to Anaconda Cloud' button, and a left-hand sidebar with navigation options: 'Home', 'Environments', 'Learning', and 'Community'. The main workspace is titled 'Applications on' with a dropdown menu set to 'renv' and a 'Channels' button. A 'Refresh' button is located in the top right of the application grid. The grid contains eight application tiles, each with an icon, name, version number, description, and an action button (either 'Launch' or 'Install').

Application	Version	Description	Action
Jupyter Notebook	5.6.0	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	Launch
Glueviz	0.15.2	Multidimensional data visualization across files. Explore relationships within and among related datasets.	Install
JupyterLab	1.1.4	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	Install
Orange 3	3.23.1	Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.	Install
RStudio	1.1.456	A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.	Install
Spyder	3.3.6	Scientific PYTHON Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features	Install
VS Code	1.39.2	Streamlined code editor with support for development operations like debugging, task running and version control.	Install

Home

Environments

Projects (beta)

Learning

Community

Documentation

Developer Blog

Feedback



Search Environments

root

Installed

Channels

Update index...

python

Name	T	Description	Version
✓ pycrypto		Cryptographic modules for python	2.6.1
✓ pycurl		A python interface to the curl library	7.43.0
		on programs	1.6.0
		cker	1.7.4
		e around the openssl library	17.2.0
		cross-platform gui toolkit qt	5.6.0
		odule	1.6.7
		esting with python	3.2.1
✓ python		General purpose programming language	3.6.3
✓ python-dateutil		Extensions to the standard python datetime module	2.6.1
✓ python.app		Proxy on os x letting python libraries hook into the gui event loop	2
✓ pyyaml		Yaml parser and emitter for python	3.12
✓ pyzmq		Zeromq bindings for python	16.0.2
✓ requests		Python http for humans	2.18.4

Create new environment

Name: Tensorflow

Location: /Applications/anaconda/envs/Tensorflow

Packages: ☒ Python

3.6

☐ R

Cancel

Create



Create

Clone

Import

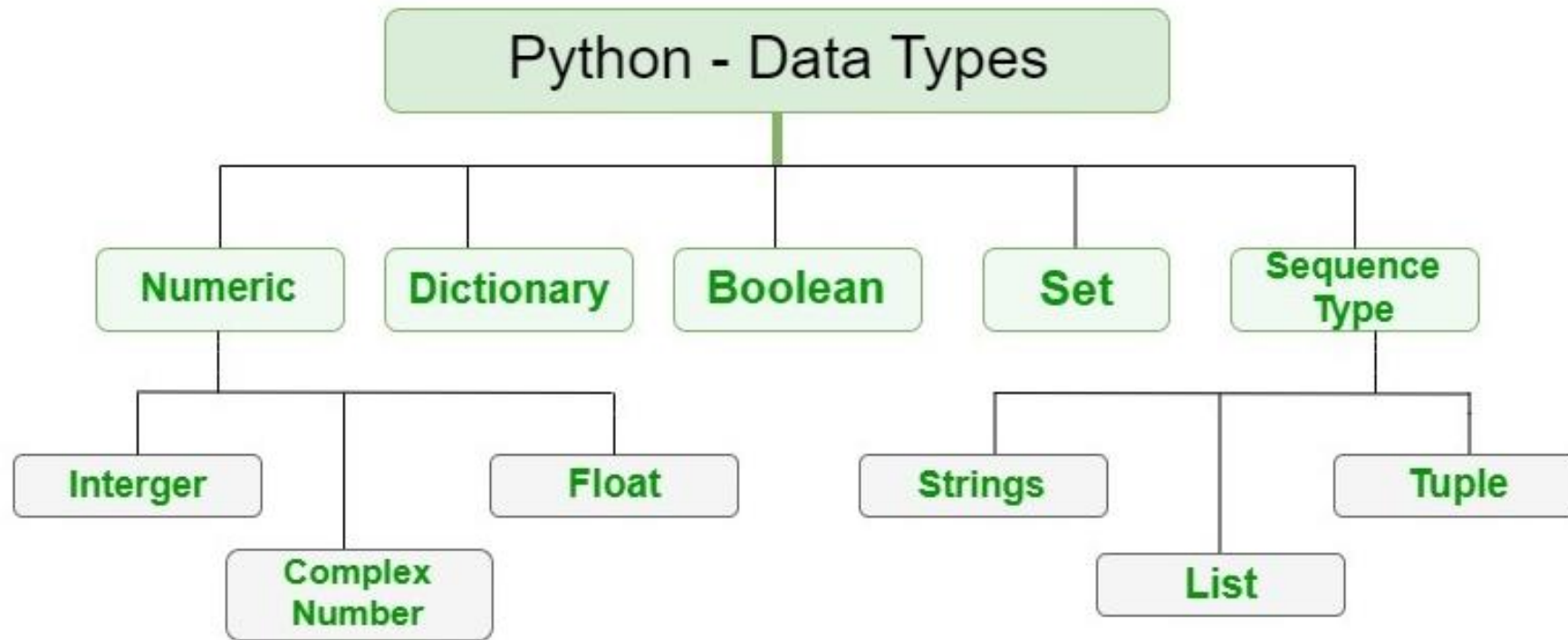
Remove

88 packages available matching "python"

# Python



# Python



# Python (Data Types/Structures)

```
# THIS IS A COMMENT

# NUMBERS
a = 5 # Integers
print("Type of a: ", type(a))
b = 5.0 # Numbers on the real line
(float)
print("\nType of b: ", type(b))
```

>\_ Code





python™

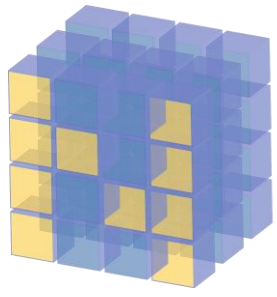
# Python (Data Types/Structures)

```
# TEXT
my_string = "My 1st string"
print(my_string)
my_string = 'My 2nd string'
print(my_string)
my_string[0]
my_string[-3]
my_string[4:]
my_string[:4]
my_string[-3:]
my_string[:-3]
```

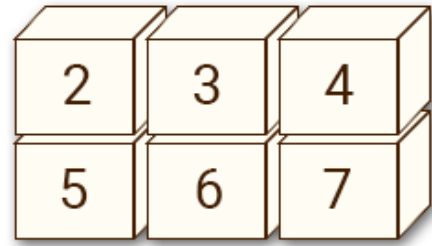
M	y		2	n	d		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12
-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

# Python (Data Types/Structures)

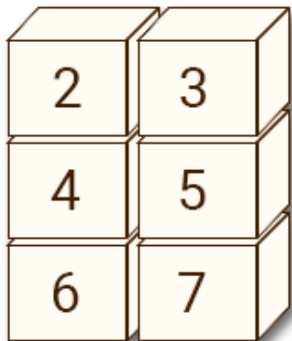
```
('a', 5) # tuple  
['a', 5, True] # list  
{2, 'a'} # set  
mydict =  
{ 'a': ['Professor', 'Students'], 'x': [111, 555] } # Dictionary  
mydict.keys()
```



# NumPy Arrays

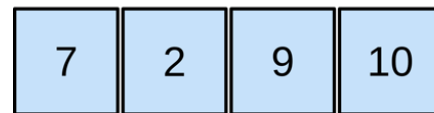


np.reshape(3, 2)



```
import numpy as np
a=np.array([7.,2.,9.,10.])
x=np.array([[2,3,4],[5,6,7]])
x.shape
X[,-1:]
x.reshape(3,2)
x.reshape(3,-1)
```

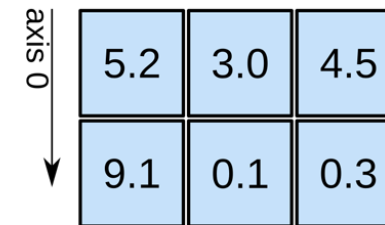
1D array



axis 0 →

shape: (4,)

2D array

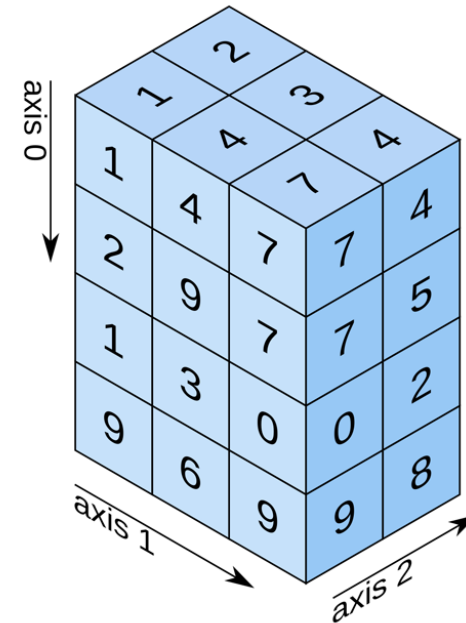


axis 0 ↓

axis 1 →

shape: (2, 3)

3D array



axis 0 ↓

axis 1 ↘

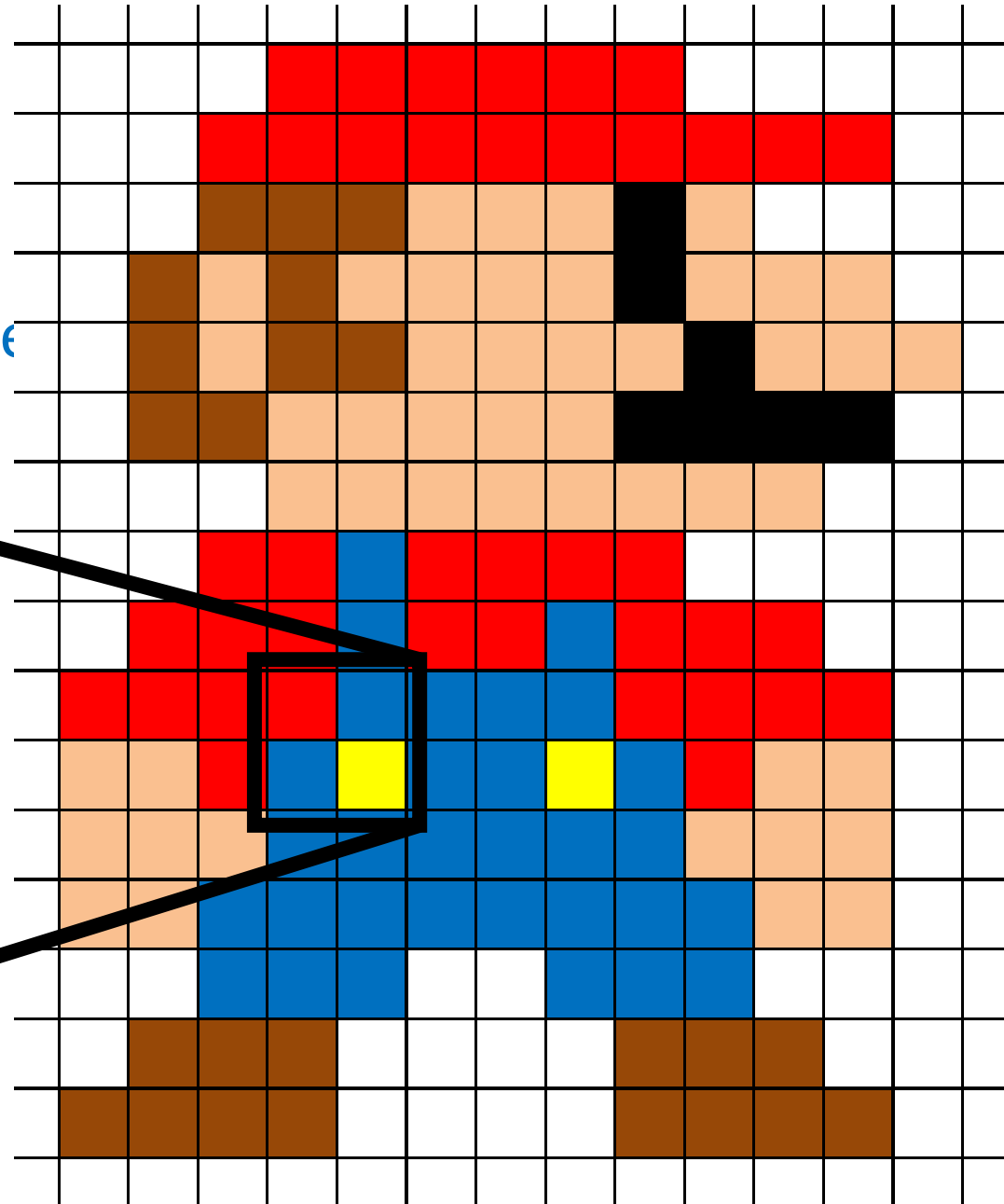
axis 2 ↗

shape: (4, 3, 2)

# RGB colors

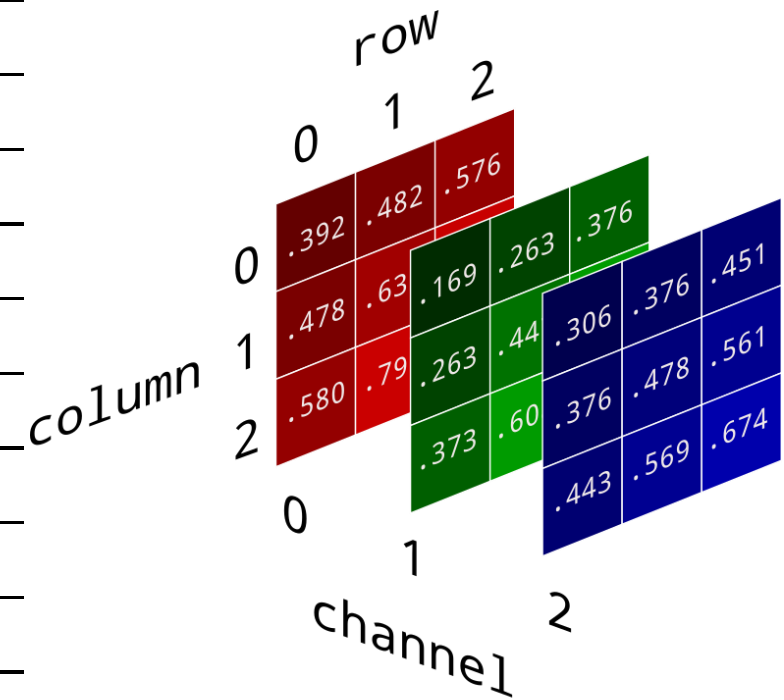
- Red, Green, Blue

R = 255	R = 0
G = 0	G = 112
B = 0	B = 192
R = 0	R = 255
G = 112	G = 255
B = 192	B = 0



>\_ Code

Numpy Array:



# Python (Pandas)



python™



pandas

## Pandas Data Structures

### Series

A one-dimensional labeled array capable of holding any data type

	A	3
	B	-5
	C	7
	D	4

Index

```
>>> s = pd.Series([3, -5, 7, 4], index=['a', 'b', 'c', 'd'])
```

### DataFrame

Columns

	Country	Capital	Population
1	Belgium	Brussels	11190846
2	India	New Delhi	1303171035
3	Brazil	Brasília	207847528

A two-dimensional labeled data structure with columns of potentially different types

Index

```
>>> data = {'Country': ['Belgium', 'India', 'Brazil'],  
           'Capital': ['Brussels', 'New Delhi', 'Brasília'],  
           'Population': [11190846, 1303171035, 207847528]}
```

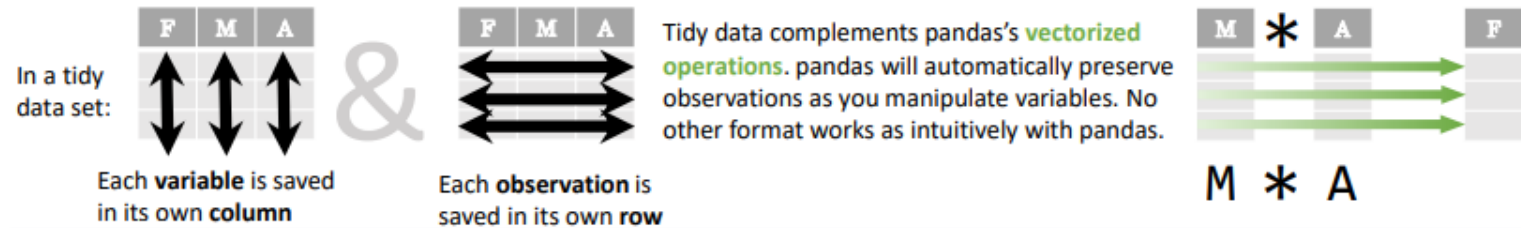
```
>>> df = pd.DataFrame(data,  
                      columns=['Country', 'Capital', 'Population'])
```

```
import pandas as pd  
x=pd.DataFrame({'a':['Professor', 'Students'], 'b':[111,555]},index=[2,15])  
print(x)  
print(type(x))  
#Slicing  
x.iloc[15,]  
x.loc[15,]#index-based  
x.y  
x['y']  
x[['y']]
```

# Data Wrangling with pandas Cheat Sheet

<http://pandas.pydata.org>

## Tidy Data – A foundation for wrangling in pandas



## Syntax – Creating DataFrames

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
    {"a" : [4 ,5, 6],
     "b" : [7, 8, 9],
     "c" : [10, 11, 12]},
    index = [1, 2, 3])
```

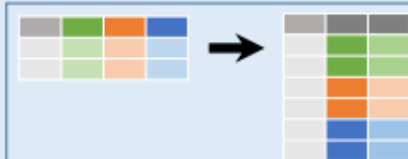
Specify values for each column.

```
df = pd.DataFrame(
    [[4, 7, 10],
     [5, 8, 11],
     [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
```

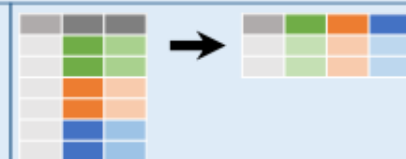
Specify values for each row.

	a	b	c
n			
d	1	4	7
e	2	5	8
	3	6	9

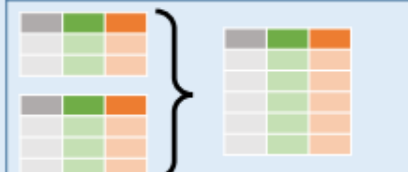
## Reshaping Data – Change the layout of a data set



`pd.melt(df)`  
Gather columns into rows.



`df.pivot(columns='var', values='val')`  
Spread rows into columns.



`pd.concat([df1, df2])`  
Append rows of DataFrames



`pd.concat([df1, df2], axis=1)`  
Append columns of DataFrames

`df.sort_values('mpg')`  
Order rows by values of a column (low to high).

`df.sort_values('mpg', ascending=False)`  
Order rows by values of a column (high to low).

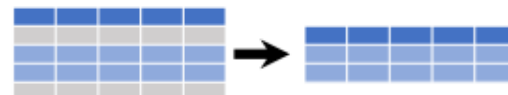
`df.rename(columns = {'y':'year'})`  
Rename the columns of a DataFrame

`df.sort_index()`  
Sort the index of a DataFrame

`df.reset_index()`  
Reset index of DataFrame to row numbers, moving index to columns.

`df.drop(columns=['Length', 'Height'])`  
Drop columns from DataFrame

## Subset Observations (Rows)



## Subset Variables (Columns)





```
import numpy as np
import matplotlib.pyplot as plt
x=np.random.rand(100)
y=np.random.normal(0,2, (100,1))
graph = plt.plot(x, y, 'b.')
plt.axhline(0.5)
plt.axvline(0.5,c='r')
plt.axis('equal')
plt.show()

plt.scatter(x=x, y=y, c='g')
plt.show()
```

# Jupyter Lab Notebooks





# VSCode Extensions

- Markdown
- Python
- Dash/Zeal
  - Ctrl+H

# Homework assignment 01