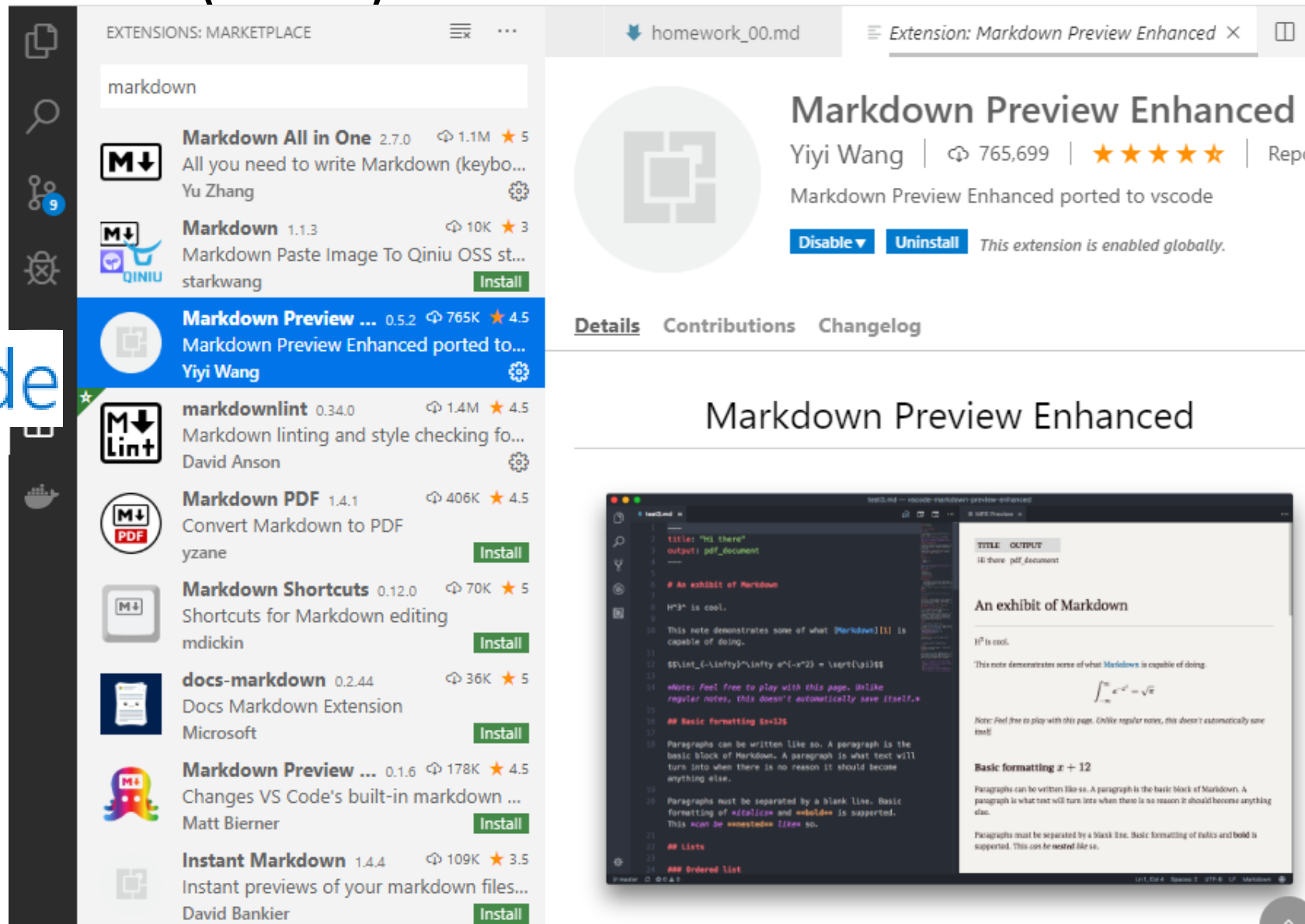
Software tutorial

VSCode

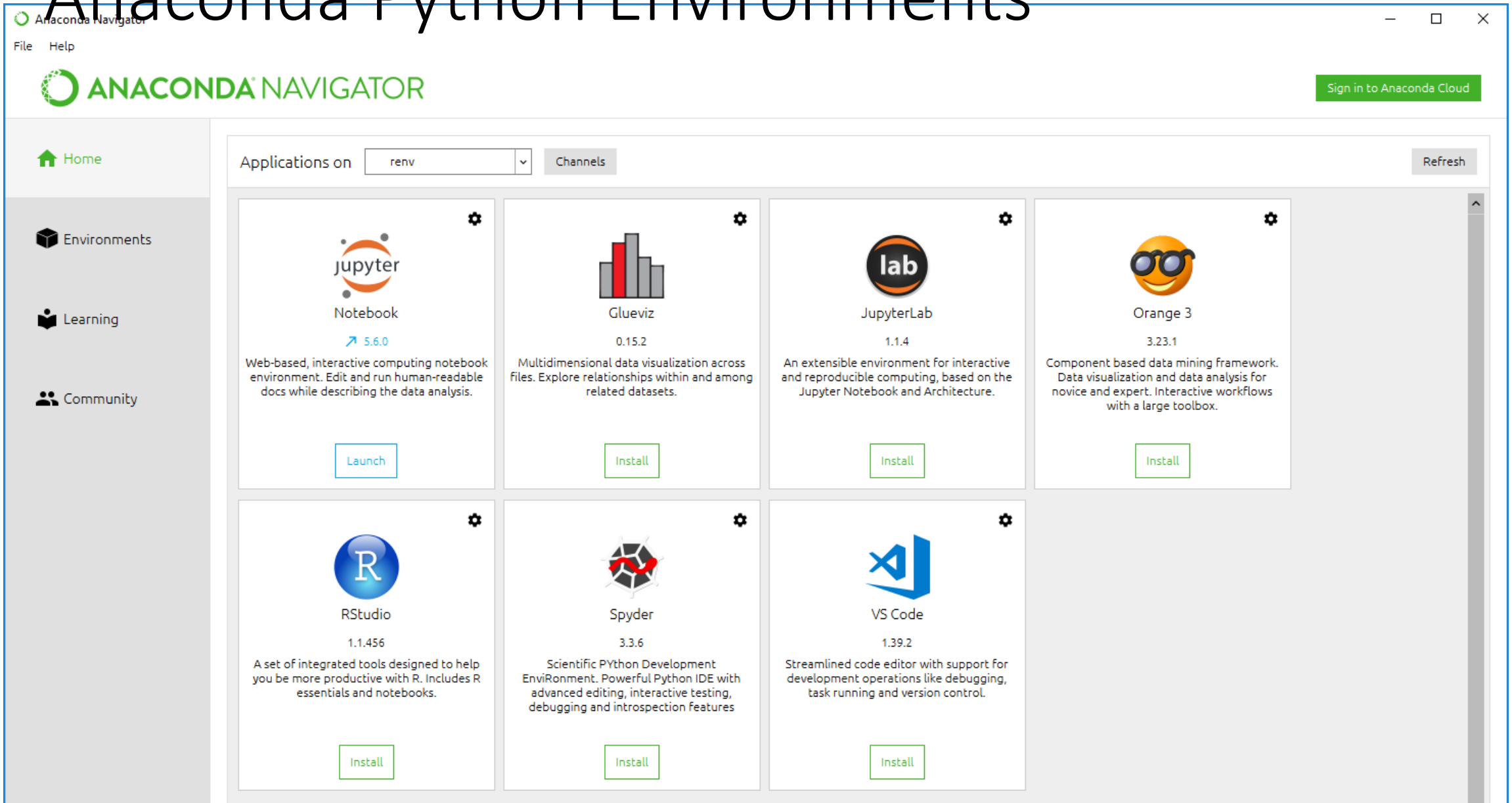
- Install Visual Studio Code
  - Explain the GUI of VSCode
  - Show example of MarkDown visualization
  - Show example of installing VSCode extensions

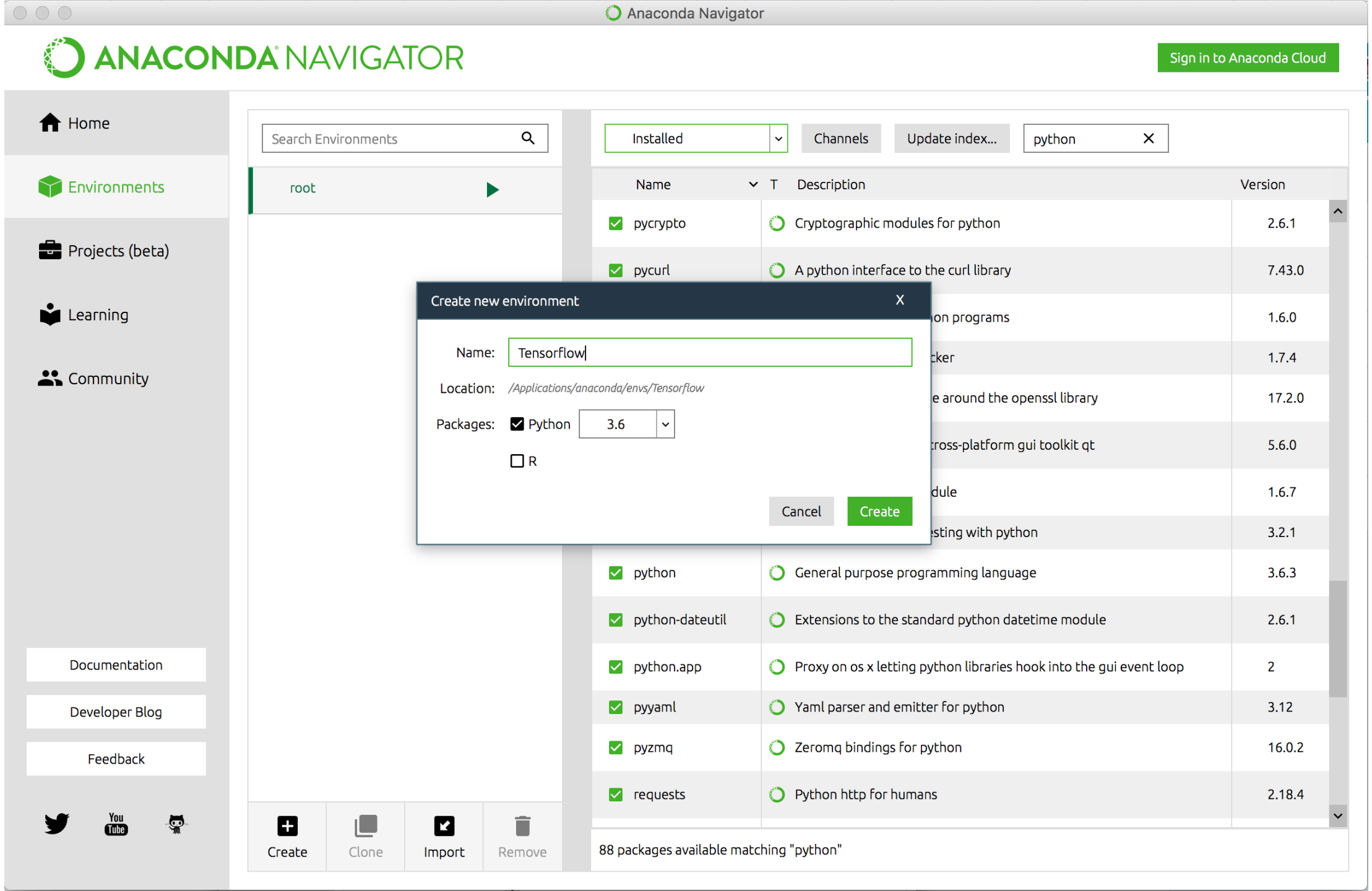
VSCode MarkDown (.md)

Visual Studio Code



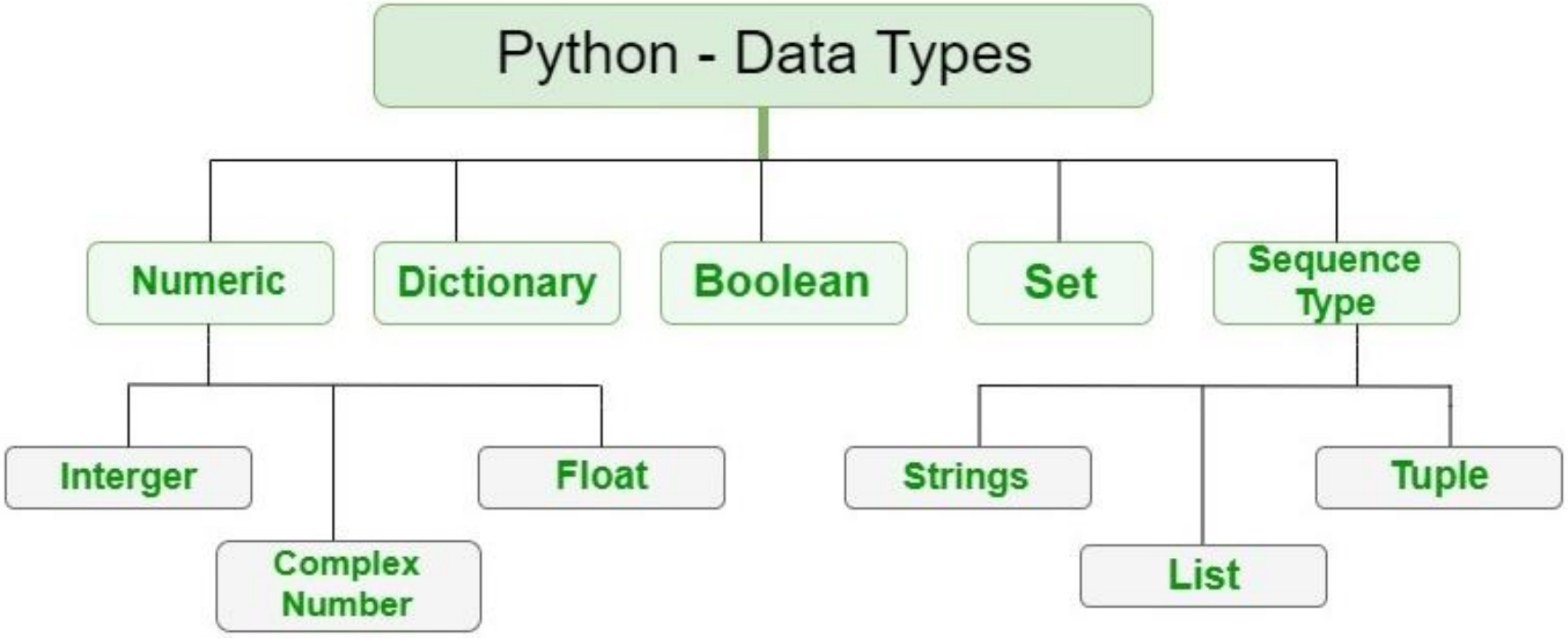
Anaconda Python Environments




Anaconda Navigator
ANACONDA NAVIGATOR
Sign in to Anaconda Cloud
Home
Environments
Projects (beta)
Learning
Community
Documentation
Developer Blog
Feedback
Search Environments
root
Create
Clone
Import
Remove
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
dule
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
88 packages available matching "python"
Create new environment
Name:
Tensorflow
Location:
/Applications/anaconda/envs/Tensorflow
Packages:
Python
3.6
R
Cancel
Create

Python

Python



https://www.geeksforgeeks.org/python-data-types/

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 (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]
```

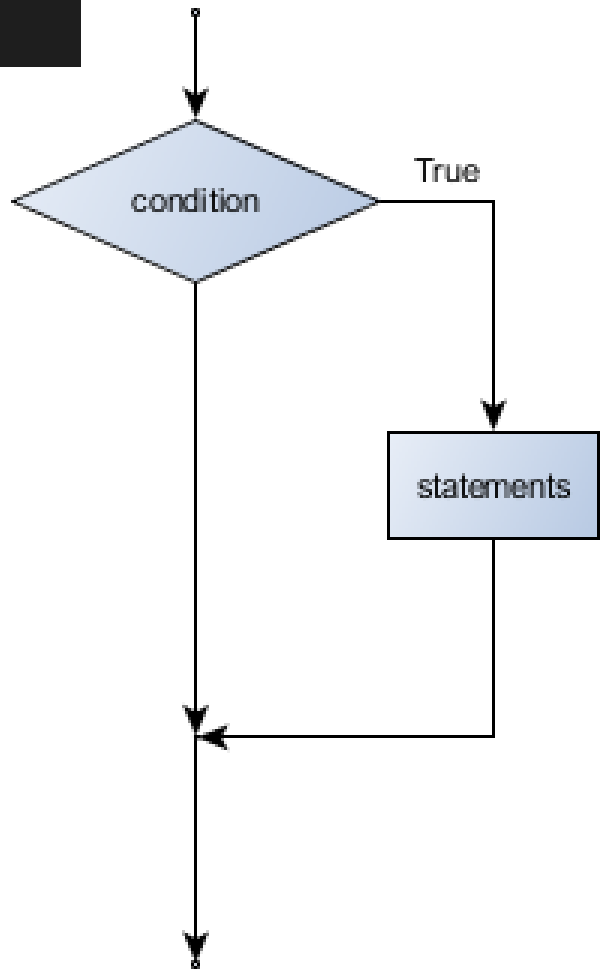
M	y		2	n	d		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12
-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Python (Data Types/Structures)

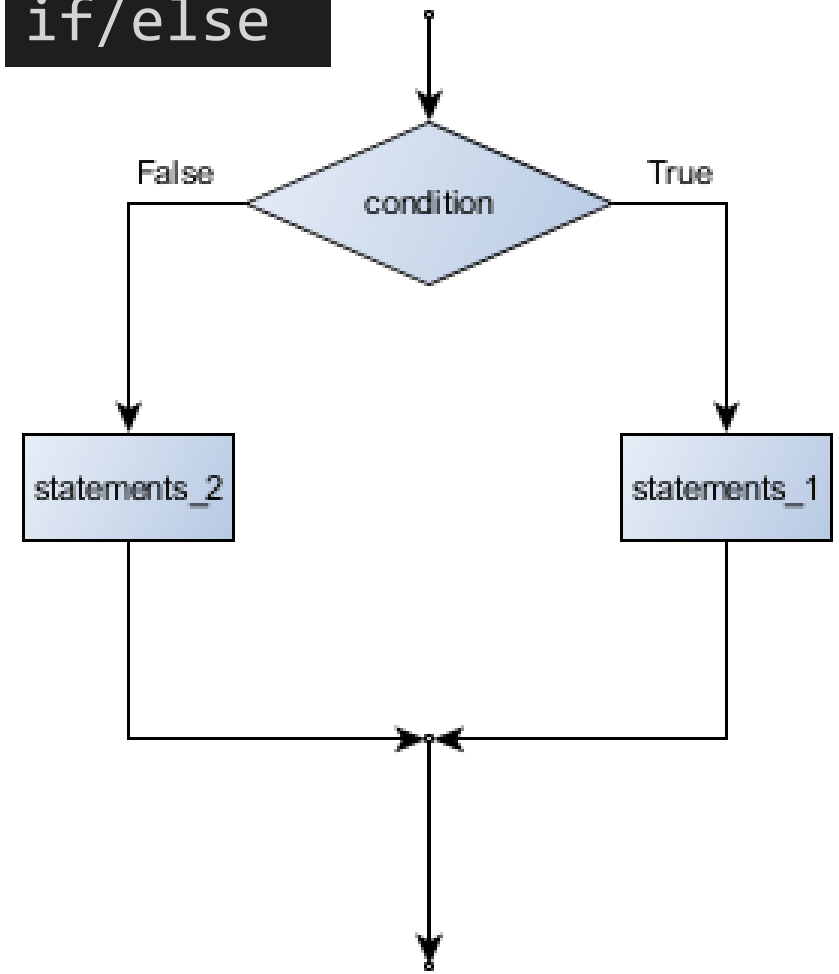

```
('a', 5) # tuple
['a', 5, True] # list
{2, 'a'} # set
mydict={'a':['Professor','Students'],'x':[111,555]}# Dictionary
mydict.keys()
```

Python (Coonditionals and loops)

`if`



`if/else`



`for`

Python Exceptions

```
x=6,denominator=4
try:
    y=x/denominator
    # print('d2i0 removed')
except Exception as e:
    print("Error. Division by 0")
    print(e)
```

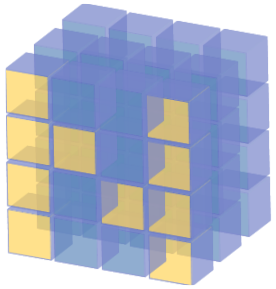
Python function **def**inition

```
def hw(x, a="aaaa", b=4, mylist=[5, 'KKK', 'PPP']):
    y=5*b
    return(a)

hw(x=3)
```

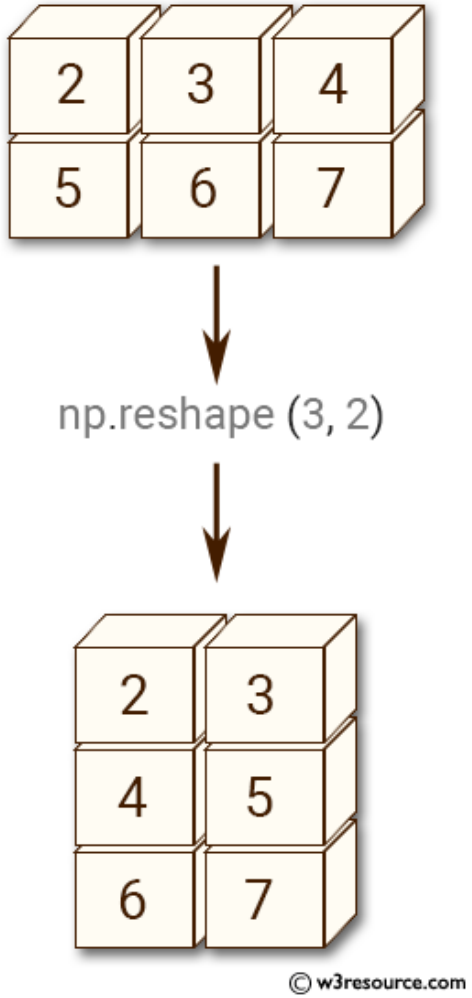
Exercise

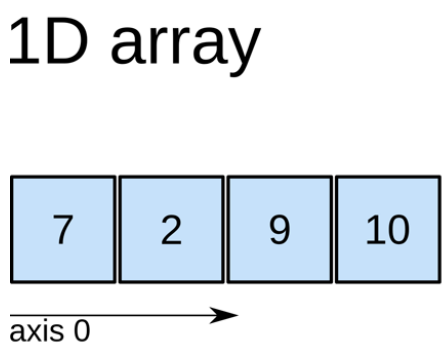
- Create a function that return the sum of 3 numbers
- Create a function `is_even()` that checks if a number is divisible by 2
- Create a for loop that prints only even numbers (5%2)
- Create a function that prints only the last 3 characters of a string, one-by-one

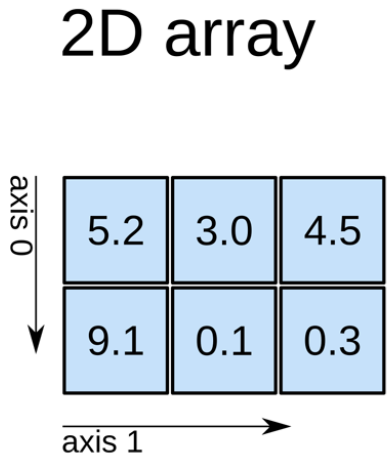


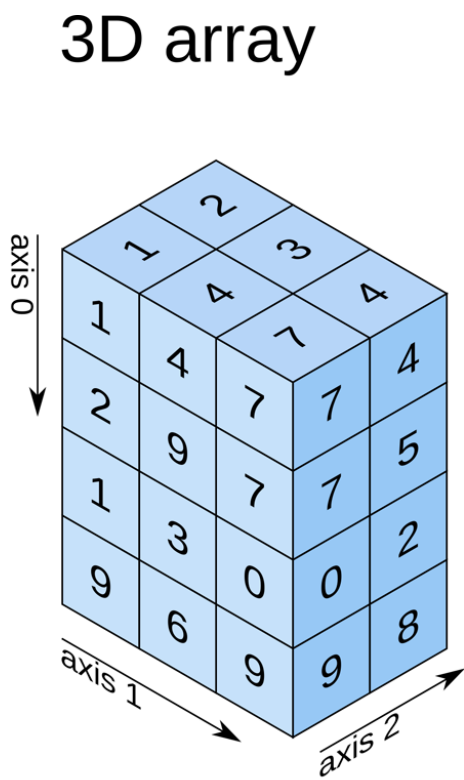
NumPy Arrays

```
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[,-1:]
x.reshape(3,2)
x.reshape(3,-1)
```









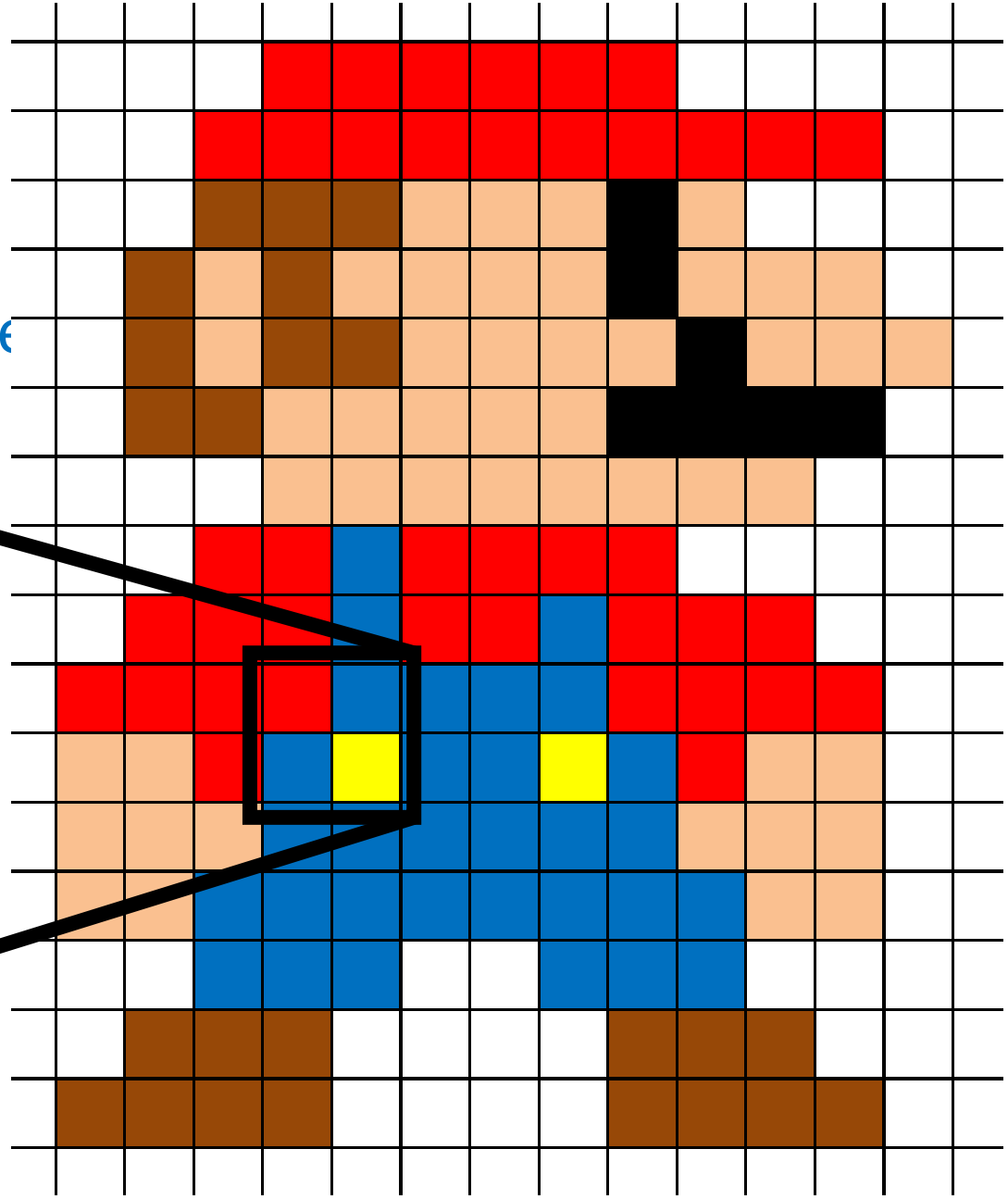
matplotlib

```
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()
```

3D Array
RGB colors

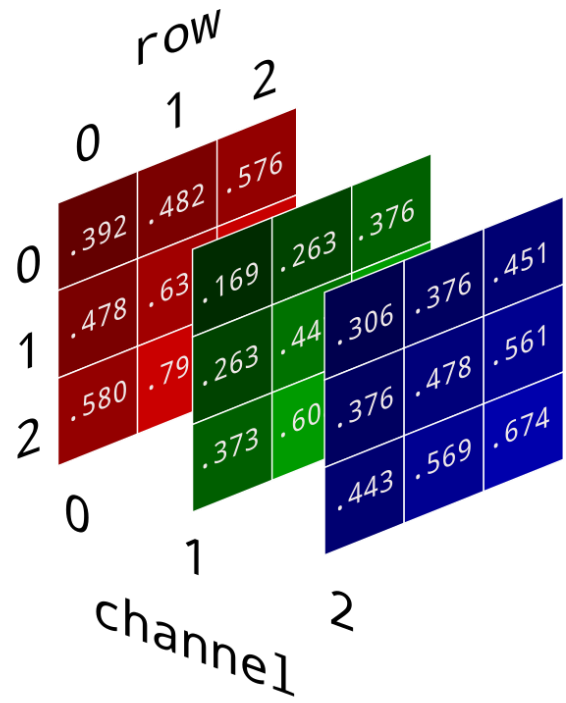
- Red, Green, Blue



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

>_ Code

Numpy Array:

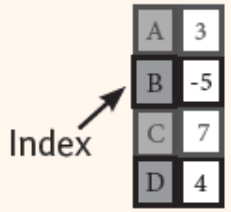


Python (Pandas)

Pandas Data Structures

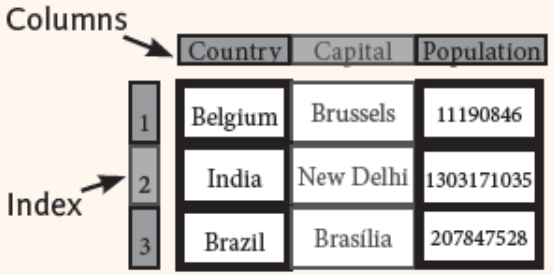
Series

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




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

DataFrame



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

```
>>> 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']]
```

python™ pandas

pandas.pydata.org/Pandas_Cheat_Sheet.pdf

Data Wrangling

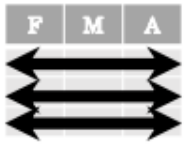
with pandas Cheat Sheet

http://pandas.pydata.org

Tidy Data – A foundation for wrangling in pandas

In a tidy data set:

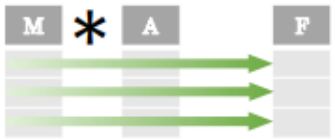
&



Each **variable** is saved in its own **column**

Each **observation** is saved in its own **row**

Tidy data complements pandas's vectorized operations. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas.



M * A

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])
```

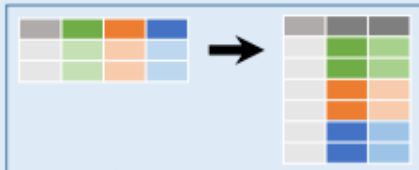
Specify values for each column.

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

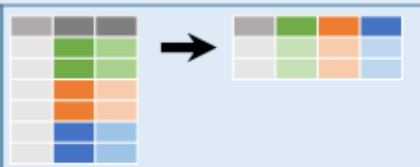
Specify values for each row.

		a	b	c
n	v			
d	1	4	7	10
	2	5	8	11
e	2	6	9	12

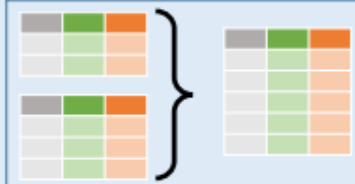
Reshaping Data – Change the layout of a data set



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 pandas as pd
import numpy as np
import matplotlib.pyplot as plt
timestamp=pd.date_range("2020-02-01","2020-02-
25",freq="d")
n=len(timestamp)
x=np.random.randn(n)
y=np.random.randn(n)
a = np.random.choice(['N', 'Y'],size=n)
df=pd.DataFrame(data={'x':x,'y':y,'a':a},index=tim
estamp)

df['y'].plot(kind='hist')
plt.show()
df['y'].plot()
plt.show()
df.plot(x='x',y='y',kind='scatter')
plt.show()
```

Jupyter Lab Notebooks

VSCode Extensions

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

Homework assignment 01