

Python Data Science and Big Data tutorials (CWH)

Playlist Link: https://youtube.com/playlist?list=PLu0W_9lII9agK8pojo23OHINz3Jm6VQCH
(https://youtube.com/playlist?list=PLu0W_9lII9agK8pojo23OHINz3Jm6VQCH)

Jupyter Notebook demo

Video Link: <https://youtu.be/TjRXT8mkTvM> (<https://youtu.be/TjRXT8mkTvM>)

```
In [1]: 'shshs'; # the output is not shown
```

```
In [2]: import numpy as np
import pandas as pd
```

```
In [3]: !mkdir this
```

```
In [4]: %lsmagic
```

Out[4]: Available line magics:

```
%alias %alias_magic %autoawait %autocall %automagic %autosave %bookmark %cd
%clear %cls %colors %conda %config %connect_info %copy %ddir %debug %dhist
%dirs %doctest_mode %echo %ed %edit %env %gui %hist %history %killbgscript
%ls %ldir %less %load %load_ext %loadpy %logoff %logon %logstart %logstate
%logstop %ls %lsmagic %macro %magic %matplotlib %mkdir %more %notebook %pa
ge %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2 %pip %popd %pprint %
precision %prun %psearch %psource %pushd %pwd %pycat %pylab %qtconsole %qu
ickref %recall %rehashx %reload_ext %ren %rep %rerun %reset %reset_selectiv
e %rmdir %run %save %sc %set_env %store %sx %system %tb %time %timeit %
unalias %unload_ext %who %who_ls %whos %xdel %xmode
```

Available cell magics:

```
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%html %%javascript
%%js %%latex %%markdown %%perl %%prun %%pypy %%python %%python2 %%python3
%%ruby %%script %%sh %%svg %%sx %%system %%time %%timeit %%writefile
```

Automagic is ON, % prefix IS NOT needed for line magics.

```
In [5]: %ls
```

```
Volume in drive C is Windows  
Volume Serial Number is 9048-2C9E
```

```
Directory of C:\Users\suman\PYTHON_FOLDER\Tutorials_CodeWithHarry\Data_analysis_CodeWithHarry
```

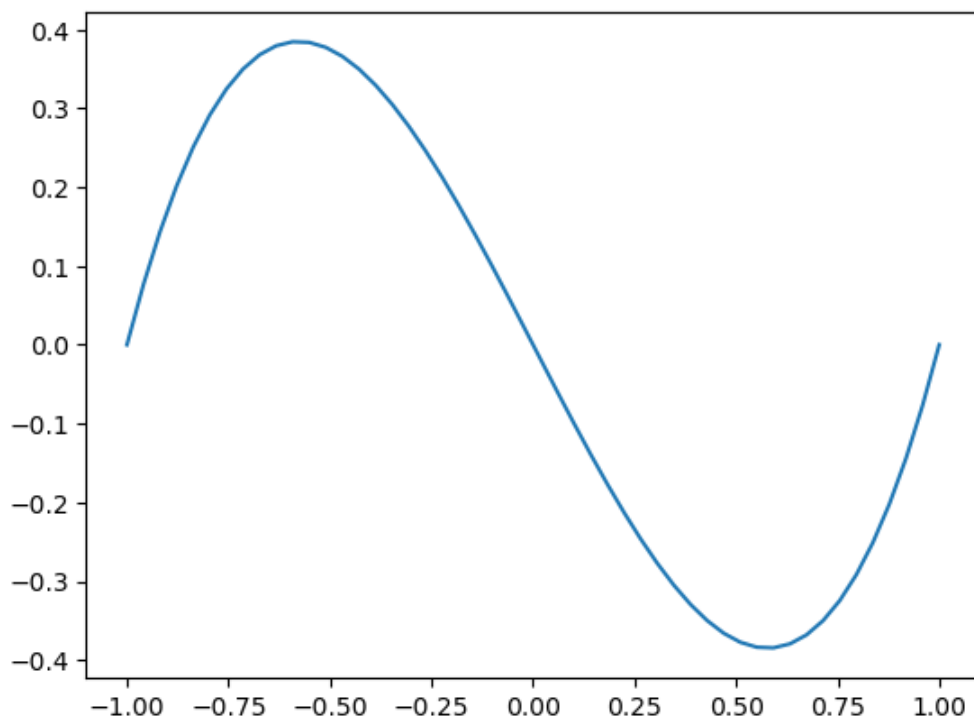
```
07-10-2023  09:09    <DIR>          .  
06-10-2023  08:49    <DIR>          ..  
07-10-2023  09:08    <DIR>          .ipynb_checkpoints  
23-05-2023  10:11             140,303 cc09 modern physics prac prog1.xlsx  
17-03-2023  13:57             29,789 NumPy Tutorials (CWH).ipynb  
16-07-2023  12:46            136,095 Pandas Tutorials (CWH).ipynb  
07-10-2023  09:08            47,031 Python Data Science and Big Data tutorials (CWH).ipynb  
16-07-2023  12:46                 318 sem 4 overview.csv  
22-05-2023  10:12                 237 single slit diffraction data.csv  
07-10-2023  09:09    <DIR>          this  
               6 File(s)          353,773 bytes  
               4 Dir(s)  250,335,387,648 bytes free
```

```
In [6]: %%HTML  
<h1> heading </h1>
```

heading

```
In [7]: %matplotlib inline  
import matplotlib.pyplot as plt  
xplt = np.linspace(-1,1,50)  
yplt = xplt**3 - xplt  
plt.plot(xplt, yplt)
```

```
Out[7]: [<matplotlib.lines.Line2D at 0x1af01b44760>]
```



```
In [8]: %timeit import sympy
```

The slowest run took 36.00 times longer than the fastest. This could mean that an intermediate result is being cached.
643 ns \pm 1.21 μ s per loop (mean \pm std. dev. of 7 runs, 1 loop each)

numpy tutorial

Video Link: https://youtu.be/HYoGR7kq_sA (https://youtu.be/HYoGR7kq_sA)

numpy is written in C (low level language). That's why it takes less space and works faster than python lists.

```
In [9]: import numpy as np
```

```
In [10]: arr1 = np.arange(100000)
list1 = list(range(100000))
%time for _ in range(10): [item*3 for item in list1]
%time for _ in range(10): arr1 = arr1*3
```

Wall time: 32.9 ms
Wall time: 0 ns

Numpy array functions and slicing

Video Link: <https://youtu.be/cg8P0NSe8Ew> (<https://youtu.be/cg8P0NSe8Ew>)

```
In [11]: import numpy as np
ar1 = np.array([[1,5,6],[7,2,4],[1,6,8]])
print(ar1)
print(type(ar1), ar1.dtype, ar1.shape)
ar2 = np.empty((4,5))
print('empty array', ar2, ar2.dtype)
```

```
[[1 5 6]
 [7 2 4]
 [1 6 8]]
<class 'numpy.ndarray'> int32 (3, 3)
empty array [[0.  0.  0.3 1.  0. ]
 [0.  1.  1.  1.  1. ]
 [1.  1.  1.  0.  0. ]
 [1.  0.5 0.  0.  1. ]] float64
```

```
In [12]: print('element multiplication', ar1*ar1)
print('matrix multiplication', ar1@ar1)
```

```
element multiplication [[ 1 25 36]
 [49  4 16]
 [ 1 36 64]]
matrix multiplication [[42 51 74]
 [25 63 82]
 [51 65 94]]
```

```
In [13]: print(ar1[0:2])
ar3 = np.array([1,5.1,3,4,0,4,5,5,7,3,2,4])
print(ar3)
ar3v = ar3[3:5] # view
ar3c = ar3[6:9].copy() # copy
ar3v[0] = np.pi
ar3c[1] = np.pi
print(ar3)
```

```
[[1 5 6]
 [7 2 4]]
[1.  5.1 3.  4.  0.  4.  5.  5.  7.  3.  2.  4. ]
[1.          5.1          3.          3.14159265 0.          4.
 5.          5.          7.          3.          2.          4.          ]
```

axis, sorting & useful numpy function

Video Link: <https://youtu.be/mcpphUjarHo> (<https://youtu.be/mcpphUjarHo>)

```
In [14]: print('ar1 =', ar1, '\nar2 =', ar2, '\nar3 =', ar3)
```

```
ar1 = [[1 5 6]
 [7 2 4]
 [1 6 8]]
ar2 = [[0.  0.  0.3 1.  0. ]
 [0.  1.  1.  1.  1. ]
 [1.  1.  1.  0.  0. ]
 [1.  0.5 0.  0.  1. ]]
ar3 = [1.          5.1          3.          3.14159265 0.          4.
 5.          5.          7.          3.          2.          4.          ]
```

```
In [15]: print('sum of columns', ar2.sum(axis=0))
print('sum of rows', ar2.sum(axis=1))
# resume from 4:40 min
```

```
sum of columns [2.  2.5 2.3 2.  2. ]
sum of rows [1.3 4.  3.  2.5]
```

In []:

numpy argsort, argmin, argmax and other tips

Video Link: <https://youtu.be/ZD1Qig8gmVA> (<https://youtu.be/ZD1Qig8gmVA>)

In []:

intro to pandas

Video Link: <https://youtu.be/iOX150jUnH8> (<https://youtu.be/iOX150jUnH8>)

In []:

creating pandas dataframe

Video Link: <https://youtu.be/Q5F9WLhtfRk> (<https://youtu.be/Q5F9WLhtfRk>)

In []:

slicing and adding columns to a dataframe

Video Link: <https://youtu.be/VZwYd4UaaeY> (<https://youtu.be/VZwYd4UaaeY>).

In []:

reading a csv file in pandas

Video Link: <https://youtu.be/TRJr5h6HI8Q> (<https://youtu.be/TRJr5h6HI8Q>).

In []:

writing a csv file in pandas

Video Link: <https://youtu.be/qtxrpHjG-oE> (<https://youtu.be/qtxrpHjG-oE>).

In []:

merging dataframes in python

Video Link: https://youtu.be/v_f__2IWfZI (https://youtu.be/v_f__2IWfZI).

In []:

writing a json in python

Video Link: https://youtu.be/Rolq7_E4ses (https://youtu.be/Rolq7_E4ses).

In []:

matplotlib tutorial

Video Link: <https://youtu.be/VFsRLjSc8GA> (<https://youtu.be/VFsRLjSc8GA>).

In []:

In []: