



# Neural Data Science with Python

## Cheat Sheet

Find all course information on <https://github.com/mgroupe/DataSciPy2022>

## > Variables and Data Types

### Variable Assignment

```
In[0]: x=5           #Comments are preceeded by '#'
In[1]: x
5
```

### Calculatoinis with Variables

```
In[2]: x+2           #Sum of two variables
7
In[3]: x-2           #Substraction of two variables
3
In[4]: x*2           #Multiplicaiton of two variables
10
In[5]: x**2          #Exponentiation of a variable
25
In[6]: x%2           #Remainder of a variable
1
In[7]: x/float(2)    #Division of a variable
2.5
```

### Types and Type Conversation

```
str()                #Variables to strings
"5", "3.7", "True", "False"
int()                #Variables to integers
5, 3, 1, 0
float()              #Variables to floats
5, 3.7, 1.0, 0.0
bool()               #Variables to booleans
True, True, True, False
```

## > Libraries



### Import Libraries

```
In[8]: import numpy           #Imports the numpy library
In[9]: import numpy as np     #Imports the numpy library and calls it 'np'
In[10]: np.mean(...)          #Using the numpy function 'mean()'
In[11]: import matplotlib.pyplot as plt #Imports the matplotlib plotting library and calls 'plt'
```

### Selective import

```
In[12]: from math import pi   #Imports the specific function 'pi' from the math library
```

## > Asking for Help

```
In[13]: help(str)
In[14]: help(mp.mean)
```

## > NumPy Arrays

```
In[15]: my_list = [2,4,6,8,10]
In[16]: my_array1 = np.array(my_list)           #creates 1-dimensional array
In[17]: my_array2 = np.arange(10)              #create 1d vector of 10 elements, from 0 to 9
In[18]: my_array3 = np.zeros(3)                 #creates array of length 3 containing zeros
In[19]: my_2darray = np.array([[1,2,3],[4,5,6]]) #2-dimensional array, 2 rows, 3 columns
```

### Selecting NumPy Array Elements

Index starts at 0

```
Subset
In[20]: my_array1[2]           #select item at index 2
6
Slice
In[21]: my_array1[1:4]         #select items at indices 1 to 3
array([4,6,8])
In[22]: my_array[2:]           #all items starting from index 2
array([6,8,10])
In[23]: my_array[:-1]          #all but the last element
array([2,4,6,8])
Subset 2D NumPy arrays
In[24]: my_2darray1[:,1]       #select all items (":") of row with index 1, my_2darray[rows, columns]
array([2,5])
```

### NumPy Array Operations

```
In[25]: my_array1 > 4
array([False,False,True,True],dtype=bool)
In[26]: my_array1 * 2
array([4,8,12,16,20])
In[27]: my_array1 + np.array([1,2,3,4,5])
array([3,6,9,12,15])
```

### NumPy Array Functions

```
In[28]: np.shape(my_2darray)   #Get the dimensions of the array
(2,3)
In[29]: len(my_array1)         #Get the length - number of rows for 2d - of an array
5
In[30]: np.mean(my_array1)     #Mean of the array
6.0
```

## > Lists

See also NumPy Arrays

```
In[31]: a = 5
In[32]: my_list = ['my', True, 2, 3.14, a]
In[33]: my_list2 = [[1,'a',True],[2,'b',False]]
```

### Selecting List Elements

Index starts at 0

Subse and Slice → see NumPy Arrays

```
Subset Lists of Lists
In[34]: my_list2[1][0]         #my_list[list][itemOfSublist]
In[35]: my_list2[1][:2]
```

### List Operations

```
In[36]: my_list + my_List
['my', True, 2, 3.14, 5, 'my', True, 2, 3.14, 5]
In[37]: my_list * 2
['my', True, 2, 3.14, 5, 'my', True, 2, 3.14, 5]
```

### List Methods

```
In[38]: my_list.index(a)        #Get the index of an item
In[39]: my_list.count(a)        #Count an item
In[40]: my_list.append('!')     #Append an item at a time
In[41]: my_list.remove('!')     #remove a specific item
In[42]: my_list.pop(-1)         #Remove an item at specific index
In[43]: my_list.reverse()       #Reverse the list
In[44]: my_list.inert(0,'!')    #Insert item at specified index
In[45]: my_list.sort()          #Sort the list
```

## > Control structures

### Compare expressions

```
In[46]: x == y                 #Returns True if x is equal to y
In[47]: x != y                 #True if x is NOT equal to y
In[48]: x > y                  #True if x is larger than y
In[49]: x <= y                 #True if x is smaller or equal to y
```

### Conditional statements

```
In[50]: if x>4:                #Condition
        print('x is larger than 4') #Indented line after the colon ':' is executed if condition is True
    elif x<0:                   #else condition
        print('x is smaller than 0')
    else:                       #if none of the above was True
        print('x is between 0 and 4')
```

### for and while Loops

```
In[51]: a = np.array([2,6,8,10])
In[52]: for i in range(len(a)): #Iterates over sequence; use index of arrays,lists in the loop
        print(i,a[i])          #Indented line!
                                #Prints index and array element

0  2
...

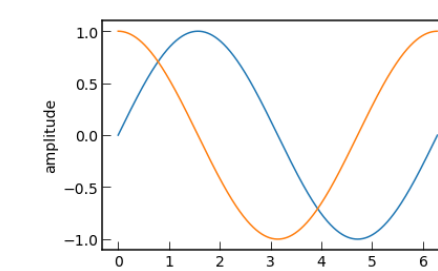
In[53]: x = 0
In[54]: while < 3:             #while loop is executed as long as condition is True
        print(x)               #Indented line!
        x=x+1                  #Multiple indented line are all executed if condition is True

0
...
```

## > Plotting

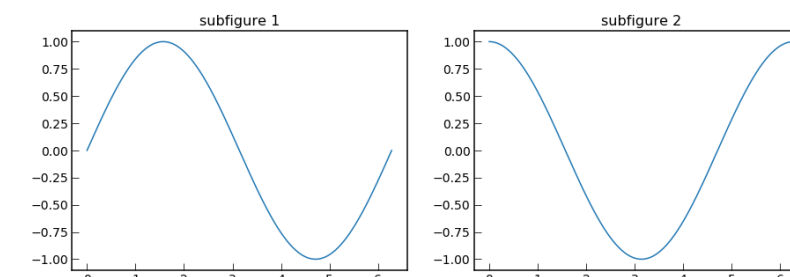
### Simple plot

```
In[55]: import matplotlib.pyplot as plt #Import matplotlib plotting library
In[56]: x = np.linspace(0,2*np.pi,1000) #Create uniformly spaced array between 0 and 2*Pi
In[57]: y1 = np.sin(x); y2 = np.cos(x)   #Calculate sin and cos function of each array element
In[58]: plt.plot(x,y1)                  #Plot y1 over x, the sin function
In[59]: plt.plot(x,y2)                  #Plot y2 over x, the cos function
In[60]: plt.xlabel('time',size=14)       #Add x-axis label
In[61]: plt.ylabel('amplitude',size=14)  #Add y-axis label
In[62]: plt.show()                      #Show figure
```



### Multipanel figure

```
In[63]: fig = plt.figure(figsize=(15,5)) #Creates figure of specified size in inches
In[64]: ax0 = fig.add_subplot(121)       #Adds subplot, format: (rows,columns,number of subplot)
In[65]: ax0.set_title('subfigure 1',size=16) #Add subplot title
In[66]: ax0.plot(x,y1)                   #Plot y1 over x, the sin function
In[67]: ax1 = fig.add_subplot(122)       #Adds subplot, index is here 2 for 2nd subplot!
In[68]: ax1.set_title('subfigure 2',size=16) #Add subplot title
In[69]: ax1.plot(x,y2)                   #Plot y2 over x, the cos function
In[70]: plt.show()                      #Show figure
```



## > Python IDEs (Integrated Development Environment)

