

# Stage 1 Python Cheat Sheet

Dr Chris Graham ([christopher.graham@ncl.ac.uk](mailto:christopher.graham@ncl.ac.uk))

Good practice	
# A comment	Don't forget to comment your code
""" docstring """	docstring for module or function help
Have a sensible way of organising your files, such as a folder structure by week. Never use untitled.py as a filename (!)	

Importing modules	
import numpy as np	Import NumPy module
import matplotlib.pyplot as plt	Import PyPlot from Matplotlib

Creating lists and arrays	
[3,10,7,4]	Create a list with specific values
range(1,11)	A range of values 1,2,3...10 (not 11)
range(1,11,2)	A range of values 1,3,5...9 (not 11)
np.linspace(0,5,10)	Linearly spaced NumPy array of 10 values between 0 and 5
np.arange(1,11,2)	Create a NumPy array with values 1,3,5,7,9
np.zeros(10)	Create a NumPy array with ten elements, each with value 0

Querying List Values	
x[2]	Returns the <b>third</b> value in a list or array x
x[a:b]	Return values in x with index a through to (b-1)
x[a:]	Values in x with index a through to the array end
x[:b]	Return values in x with from array start to index b

Miscellaneous tips	
Be consistent with your indenting!	CTRL-C will stop execution of an infinite loop. Explore and have fun! No question is too silly - try to self-diagnose issues with your code, but if you need help, just ask!

Data types	
<ul style="list-style-type: none"><li>Text: str</li><li>Numeric: int, float, complex</li><li>Sequence: list, range, tuple, dict</li><li>Boolean: bool</li><li>NumPy: np.ndarray (NumPy array)</li></ul>	
type(x)	Query the type of `x`
str(x)	Convert `x` to type `str`

Some functions (that do pretty much what you'd expect)
sum(x)
min(x)
max(x)
len(x)
sorted(x)

Class Example	
<pre>class Rectangle:      def __init__(self, x, y):         self.x = x         self.y = y      def area(self):         return self.x * self.y  r = Rectangle(2,3) print(r.area())</pre>	

Variable management
Note these commands will work in Spyder & other "iPython" consoles.
%reset
Clear all variables
%whos
View all variables (or use variables tab)

Logical operators	
==	Equal to
!=	Not equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
and	And
or	Or

modulo
8 % 2 == 0

**Disclaimer:** not intended to be exhaustive (or exhausting)!

Printing and formatting numbers	
print(x)	Print a variable x
print("x = {}, y = {}".format(x,y))	Print a string with variables
str(x)	Convert x to a string
round(x,n)	Round x to n d.p.

Assignment operators				
x += 1	x -= 1	x *= 2	x /= 2	x **= 2

Random Numbers	
np.random.rand()	Random number in [0,1]
np.random.rand(2,2)	Random [0,1] in a 2 x 2 array
np.random.randint(0,10)	Random integer between 0 and 9
np.random.normal(0,1)	Random form norm. dist. mean 0 and s.d. 1

Plotting Example
<pre>import numpy as np import matplotlib.pyplot as plt  # Set up an array x = np.linspace(0,10,100)  # Create a figure (optional) plt.figure()  plt.plot(x,np.sin(x)) plt.xlabel("x") plt.ylabel("sin(x)") plt.title("sin(x) versus x")</pre>

List comprehension example
[x**2 for x in range(5)]

Exception handling
<pre>try:     add_one("2") except:     print("exception occurred")</pre>

Function Example
<pre>def add_one(x):     """     Add 1 to input     """     return x + 1  add_one(2)  help(add_one)</pre>

Control flow examples
<pre>for n in range(5):     print(n)  x = 0 while x &lt; 5:     x += 1  if x &gt; 0:     print("positive!")</pre>