## Python Commands:

python

ipython (better functions, prefix something with ! to run a bash command; functions are green, strings are orange, and module imports are blue; %quickref 🡪 show a quick reference sheet; %lsmagic 🡪 list magic functions; %paste or %cpaste 🡪 paste codes that otherwise dont work)

Python scripts have the .py file ending and the shebang line: #!/usr/bin/env python

Regular expressions: https://regexone.com/

quit(), exit() or Strg+D 🡪 to exit python

which python (for bash, where it is stored)  
python –version (for bash, see version)

import math 🡪 import math module  
math.sqrt() 🡪 or math. andfunction (for specific operations, for ipython use math.tab to see all operations)  
„“ or ‚‘ 🡪 for string data types  
True or False 🡪 Boolean  
None 🡪 Nonetype  
type () 🡪 see data type  
isinstance („abc“, str) 🡪 see if object is spec. Datatype e.g. is abc a string?  
 up/down 🡪 scroll command history  
Strg+A 🡪 beginning of line  
Strg+E 🡪 ending of line  
Strg+right 🡪 one word to right (see Strg+left)  
Strg+K🡪 delete rest of line after cursor  
Strg+R 🡪 command history

if:   
else:

"ACC" in "AAACCCACTG" 🡪 check if string a is in string b  
"ACT" not in "AAACCCACTG" 🡪 check if string a is not in string b

[number] 🡪 indexing  
[start:stop:step] 🡪 slicing  
x is y 🡪 compare identity (gets Boolean)  
print(id(x)) 🡪 print id of x  
list[2,4,1] 🡪 creates list  
list.append(7) 🡪 appends 7 to existing list  
list.pop() 🡪 removes last added object e.g. 7  
newlist = oldlist[:] 🡪 creates newlist with different id than oldlist but same values

Strg+Shift+L 🡪 cursor to all matches of the current selection (VSCode)  
Alt+Click 🡪 to add new cursor (VSCode)

<https://www.cyberciti.biz/faq/linux-unix-bsd-extract-targz-file/> 🡪 tar commands simply (for bash)

int(), float(), string() 🡪 explicit type casting  
range(start,stop,step)

simple for loop:

#!/usr/bin/env python

list1=list(range(10,100,10))

print(type(list1))

for var in list1:

print(var)

else:

print("The loop has completed")

simple conditional:

x=1

y=2

if x==0:

print("yes")

elif y==2:

print("y=2")

else:

print("No")

len(“string”) 🡪 outputs the length of the string

abs() 🡪 returns absolute value

min(), max()  
  
list.append(7) 🡪 append 7 to a

list.pop() 🡪 deletes the latest added thing

"acgattg".count("g") 🡪 count g in string, also possible; 1. a="acgattg"; 2. a.count("g")

String.upper() 🡪 string gets transformated to uppercase letters

String.lower() 🡪 transform to lowercase letters

String.isupper() 🡪 check if string is uppercase letters, gets boolean

help(object) 🡪 get help about a object, in ipython also object?; object?? Shows the source code if available (even more info)

dir(object) 🡪 list object attributes

string.split(“patternofsplitting”) 🡪 split the string  
string.rsplit(“patternofsplitting”, positionofsplitting) 🡪 if the pattern is in the string multiple times you can further define on which position to split the string  
string.replace(“tata”, “TATA”) 🡪 replace tata with TATA  
a.startswith("ACC") 🡪 check if string starts with pattern, gets Boolean

bin(object) 🡪 binary value

nums.sort() 🡪 sorts a list in place, so it is a void function (has to do with lists being mutable)

sorted(nums) 🡪 creates a copy of the list that is sorted

reversed(nums) 🡪 reverses the list, creates new copy

list.reverse() 🡪 reverses list in place, void function

defining a function:

def fahr\_to\_celsius(temp):

# ...more python statements that make up the function body...

return ((temp-32)\*(5/9))

%reset -f 🡪 remove user specified names, for ipython

You can give the function a possible error when something unintended happens (give description for certain cases):

def add\_or\_subtract\_two(some\_num, do="a"):

"""Add or subtract 2 to/from a number"""

if do == "a":

return some\_num + 2

elif do == "s":

return some\_num - 2

else:

**raise Exception("Sorry, no idea what to do")**