Cheatsheet Day 1

Introduction to Data Analysis with Python

Basics

`x = 1`
`y = "abc"`
	Assign a value to a *variable*

`print("hello")`
`print(x, 2, 3)`
	Print one or multiple values or variables

`list(y)`
`len([0, 1, 2])`
	A *function* is called with (optional) arguments

`y.count("a")`
`"abc".upper()`
	A *method* is called on a value or on a variable

`# a comment`	A # starts a *comment* that will not be evaluated
`type(x)`	Get the *type* of a value or of a variable

Data types

`"hello"`, `'abc'`, `"0.9"`, `str(123)`	A *string* is a sequence of characters in quoation marks
`12`, `-4`, `int("5")`	An *integer* is a number without a decimal part
`0.9`, `-3.1415`, `float("-0.1")`	A *float* is a number with a decimal part
`True`, `False`, `bool(0)`, `x < 1`	A *boolean* can only take the values `True` or `False`

`[0, "abc", 0.1]`
`list("hello")`
	A *list* is a mutable, sorted collection of values

`{"a": 1, "b": 2}`
`dict(a=1, b=2)`
	A *dictionary* is a mutable collection of key-value pairs

`(0, "0.9", True)`
`tuple([0, 1, 2])`
	A *tuple* is an immutable, sorted collection of values

Strings

`s = "hello"`	Initialize a string
`len(s)`	Get the length of a string
`s[2]`, `s[1:-1]`	Get characters from a string
`s[::-1]`	Get a string in reverse order
`"he" in s`	Check if a string contains a substring
`s + "abc"`	Concatenate/append strings
`s * 10`	Repeat a string N times

`s.isalpha()`
`s.isnumeric()`
	Check properties of the characters in a string

Lists

`x = [1, 2, 3]`	Initialize a list with values
`len(x)`	Get the length of a list
`x[2]`, `x[0:-1]`	Get values from a list
`x[::-1]`	Get a list in reverse order
`x[0] = 3`	Change a value in a list
`x.append(4)`	Append a value to a list
`x.extend([4, 5])`	Extend a list by another list
`y = x + [4, 5]`	Add lists to create a new list

Dictionaries

`d = {"a": 1, "b": 2}`
`d = dict(a=1, b=2)`
	Create a dictionary with key-value pairs

`d["a"]`	Get the value of a key from a dictionary

`d["a"] = 0`
`d["c"] = 3`
	Update/add a key-value pair in/to a dictionary

`d.keys()`
`d.values()`
`d.items()`
	Get all keys, all values or all key-value pairs in a dictionary

`for` loops

`range(n)`
`range(m, n)`
	Create a sequence from `0`/`m` to `n-1` in steps of `1`

`for i in range(n):`
`    print(i)`
	Iterate over the integer numbers in a sequence

`for c in "hello":`
`    print(c)`
	Iterate over the characters in a string

`for v in [0, -1, 2]:`
`    print(v)`
	Iterate over the values in a list

`for k in d.keys():`
`    print(k)`
	Iterate over the keys in a dictionary

`x = [1, -5, 3, 0]`
`y = [v+1 for v in x]`
	Create a new list with a list comprehension

Conditions and `if` - `elif` - `else` statements

`a == b`, `a != b`
`a > b`, `a >= b`
`a < b`, `a <= b`
	Compare two values, the result will be `True` or `False`

`if a < 0:`
`    a = -a`
`elif a < 10:`
`    a = a - 10`
`else:`
`    a = a * 5`
	Check a condition with an `if` statement. Check multiple conditions with `if` and `elif` (as many as you want). The (optional) `else` block handles all other cases.

Jupyter shortcuts

Enter / Esc	Start/exit the edit mode
Shift + Enter	Run cell(s) and select next
X, C, V	Cut, copy or paste cell(s)
D + D	Delete cell(s)