Python Basics Cheat Sheet

Here you will find all the Python core concepts you need to know before learning any third-party library.

Data Types

```
Integers (int): 1
Float (float): 1.2
String (str): "Hello World"
Boolean: True/False
List: [value1, value2]
Dictionary: {key1:value1, key2:value2, ...}
```

Numeric Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Exponent
%	Modulus
//	Floor division

Comparison Operators

==	Equal to
!=	Different
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

String methods

```
string.upper(): converts to uppercase
string.lower(): converts to lowercase
string.title(): converts to title case
string.count('l'): counts how many times "l"
                   appears
string.find('h'): position of the "h" first
                  ocurrance
string.replace('o', 'u'): replaces "o" with "u"
```

Variables

Variable assignment:

```
message_1 = "I'm learning Python"
message_2 = "and it's fun!"
```

String concatenation (+ operator):

```
message_1 + ' ' + message_2
```

String concatenation (f-string):

```
f'{message_1} {message_2}'
```

List

Creating a list:

```
countries = ['United States', 'India',
             'China', 'Brazil']
```

Create an empty list:

```
my_list = []
```

Indexing:

```
>>> countries[0]
United States

>>> countries[3]
Brazil

>>> countries[-1]
Brazil
```

Slicing:

```
>>>countries[0:3]
['United States', 'India', 'China']

>>>countries[1:]
['India', 'China', 'Brazil']

>>>countries[:2]
['United States', 'India']
```

Adding elements to a list:

```
countries.append('Canada')
countries.insert(0,'Canada')
```

Nested list:

```
nested_list = [countries, countries_2]
```

Remove element:

```
countries.remove('United States')
countries.pop(0)#removes and returns value
del countries[0]
```

Creating a new list:

```
numbers = [4, 3, 10, 7, 1, 2]
```

Sorting a list:

```
>>> numbers.sort()
[1, 2, 3, 4, 7, 10]

>>> numbers.sort(reverse=True)
[10, 7, 4, 3, 2, 1]
```

Update value on a list:

```
>>> numbers[0] = 1000
>>> numbers
[1000, 7, 4, 3, 2, 1]
```

Copying a list:

```
new_list = countries[:]
new_list_2 = countries.copy()
```

Built-in Functions

Print an object:

```
print("Hello World")
```

Return the length of x:

```
len(x)
```

Return the minimum value:

```
min(x)
```

Return the maximum value:

```
max(x)
```

Returns a sequence of numbers:

```
range(x1,x2,n) # from x1 to x2
(increments by n)
```

Convert x to a string:

```
str(x)
```

Convert x to an integer/float:

```
int(x)
float(x)
```

Convert x to a list:

```
list(x)
```

Dictionary

Creating a dictionary:
```
my_data = {'name':'Frank', 'age':26}
```

Create an empty dictionary:
```
my_dict = {}
```

Get value of key "name":
```
>>> my_data["name"]
'Frank'
```

Get the keys:
```
>>> my_data.keys()
dict_keys(['name', 'age'])
```

Get the values:
```
>>> my_data.values()
dict_values(['Frank', 26])
```

Get the pair key-value:
```
>>> my_data.items()
dict_items([('name', 'Frank'), ('age', 26)])
```

Adding/updating items in a dictionary:
```
my_data['height']=1.7
my_data.update({'height':1.8,
                'languages':['English', 'Spanish']})
>>> my_data
{'name': 'Frank',
'age': 26,
'height': 1.8,
'languages': ['English', 'Spanish']}
```

Remove an item:
```
my_data.pop('height')
del my_data['languages']
my_data.clear()
```

Copying a dictionary:
```
new_dict = my_data.copy()
```

If Statement

Conditional test:
```
if <condition>:
    <code>
elif <condition>:
    <code>
...
else:
    <code>
```

Example:
```
if age>=18:
    print("You're an adult!")
```

Conditional test with list:
```
if <value> in <list>:
    <code>
```

Loops

For loop:
```
for <variable> in <list>:
    <code>
```

For loop and enumerate list elements:
```
for i, element in enumerate(<list>):
    <code>
```

For loop and obtain dictionary elements:
```
for key, value in my_dict.items():
    <code>
```

While loop:
```
while <condition>:
    <code>
```

Data Validation

Try-except:
```
try:
    <code>
except <error>:
    <code>
```

Loop control statement:
```
break: stops loop execution
continue: jumps to next iteration
pass: does nothing
```

Functions

Create a function:
```
def function(<params>):
    <code>
    return <data>
```

Modules

Import module:
```
import module
module.method()
```

OS module:
```
import os
os.getcwd()
os.listdir()
os.makedirs(<path>)
```

Special Characters

#	Comment
\n	New Line

Boolean Operators

and	logical AND
or	logical OR
not	logical NOT

Boolean Operators (Pandas)

&	logical AND
|	logical OR
~	logical NOT

Below there are my guides, tutorials and complete Data Science course:
- Medium Guides
- YouTube Tutorials
- Data Science Course (Udemy)

Made by Frank Andrade frank-andrade.medium.com