

# Python cheat sheet

(COMMONLY USED CODE SNIPPETS)

## 1. Basic Python Syntax:

| Task                       | Code  |
|----------------------------|---|
| <b>Print to Console</b>    | <code>print("Hello, World!")</code>               |
| <b>Variable Assignment</b> | <code>x = 10</code>                               |
| <b>Commenting</b>          | <code># This is a comment</code>                  |
| <b>Multi-line Comment</b>  | <code>''' This is a multi-line comment '''</code> |
| <b>Input from User</b>     | <code>name = input("Enter your name: ")</code>    |
| <b>Check Data Type</b>     | <code>type(x)</code>                              |
| <b>Type Casting</b>        | <code>int("10"), float("10.5"), str(100)</code>   |

## 2. Data Structures:

| Task                           | Code  |
|--------------------------------|---|
| <b>List (Array)</b>            | <code>my_list = [1, 2, 3, 4, 5]</code>                      |
| <b>Access List Item</b>        | <code>my_list[0]</code>                                     |
| <b>List Slicing</b>            | <code>my_list[1:4]</code>                                   |
| <b>Add Item to List</b>        | <code>my_list.append(6)</code>                              |
| <b>Remove Item from List</b>   | <code>my_list.remove(3)</code>                              |
| <b>Tuple</b>                   | <code>my_tuple = (1, 2, 3, 4)</code>                        |
| <b>Set</b>                     | <code>my_set = {1, 2, 3, 4}</code>                          |
| <b>Dictionary (HashMap)</b>    | <code>my_dict = {"key1": "value1", "key2": "value2"}</code> |
| <b>Access Dictionary Value</b> | <code>my_dict["key1"]</code>                                |
| <b>Add Key-Value Pair</b>      | <code>my_dict["key3"] = "value3"</code>                     |

## 3. Control Flow:

| Task              | Code  |
|-------------------|---|
| If Statement      | if x > 10: print("x is greater than 10")  |
| If-Else Statement | if x > 10: print("x is greater than 10") else: print("x is less than or equal to 10")       |
| Elif Statement    | if x > 10: print("x is greater") elif x == 10: print("x is 10") else: print("x is smaller") |
| For Loop          | for i in range(5): print(i)   |
| While Loop        | while x < 10: x += 1  |
| Break             | for i in range(5): if i == 3: break   |
| Continue          | for i in range(5): if i == 3: continue  |

## 4. Functions:

| Task                       | Code   |
|----------------------------|--|
| Define Function            | def my_function(): print("Hello from function!") |
| Function with Parameters   | def greet(name): print(f"Hello, {name}!")        |
| Return Value from Function | def add(a, b): return a + b                      |
| Lambda Function            | add = lambda a, b: a + b                         |

## 5. String Manipulation:

| Task                  | Code                                       |
|-----------------------|--|
| Concatenate Strings   | full_name = "John" + " " + "Doe"           |
| String Length         | len("Hello")                               |
| Convert to Upper Case | "hello".upper()                            |
| Convert to Lower Case | "HELLO".lower()                            |
| Substring             | "Hello, World!"[7:12]                      |
| Find Substring        | "Hello, World!".find("World")              |
| Replace Substring     | "Hello, World!".replace("World", "Python") |
| Split String          | "Hello, World!".split(",")                 |

## 6. File Handling:

| Task                     | Code  |
|--------------------------|---|
| <b>Open a File</b>       | <code>file = open("example.txt", "r")</code>                              |
| <b>Read File</b>         | <code>content = file.read()</code>  |
| <b>Read Line by Line</b> | <code>lines = file.readlines()</code>                                     |
| <b>Write to a File</b>   | <code>file = open("example.txt", "w"); file.write("Hello, World!")</code> |
| <b>Close a File</b>      | <code>file.close()</code>   |

## 7. List Comprehension:

| Task                                     | Code  |
|--|---|
| <b>Basic List Comprehension</b>          | <code>[x**2 for x in range(5)]</code>             |
| <b>List Comprehension with Condition</b> | <code>[x for x in range(10) if x % 2 == 0]</code> |

## 8. Error Handling:

| Task                    | Code  |
|-------------------------|---|
| <b>Try-Except Block</b> | <code>try: x = 10 / 0 except ZeroDivisionError: print("Cannot divide by zero")</code>                     |
| <b>Finally Block</b>    | <code>try: x = 10 / 0 except ZeroDivisionError: print("Error!") finally: print("This runs always")</code> |

## 9. Working with Libraries:

| Task                                 | Code                               |
|--------------------------------------|------------------------------------|
| <b>Importing a Library</b>           | <code>import math</code>           |
| <b>Using a Library Function</b>      | <code>math.sqrt(16)</code>         |
| <b>Install a Library (using pip)</b> | <code>pip install pandas</code>    |
| <b>Import Specific Function</b>      | <code>from math import sqrt</code> |

## **10. NumPy for Numerical Operations:**

| Task                      | Code                                      |
|---------------------------|---|
| <b>Import NumPy</b>       | import numpy as np                        |
| <b>Create NumPy Array</b> | arr = np.array([1, 2, 3, 4, 5])           |
| <b>Array Reshaping</b>    | arr.reshape(5, 1)                         |
| <b>Array Operations</b>   | arr + 10, arr * 2                         |
| <b>Array Slicing</b>      | arr[1:4]                                  |
| <b>Array Statistics</b>   | np.mean(arr), np.median(arr), np.std(arr) |

## **11. Pandas for Data Handling:**

| Task                    | Code   |
|-------------------------|--|
| <b>Import Pandas</b>    | import pandas as pd  |
| <b>Create DataFrame</b> | df = pd.DataFrame({"Name": ["Alice", "Bob"], "Age": [25, 30]}) |
| <b>Read CSV File</b>    | df = pd.read_csv("data.csv")                                   |
| <b>View Data</b>        | df.head()  |
| <b>Basic Statistics</b> | df.describe()  |
| <b>Filter Data</b>      | df[df["Age"] > 25]   |
| <b>Group By</b>         | df.groupby("Age").mean()                                       |

## **12. Matplotlib for plotting:**

| Task                     | Code  |
|--------------------------|---|
| <b>Import Matplotlib</b> | import matplotlib.pyplot as plt               |
| <b>Simple Plot</b>       | plt.plot([1, 2, 3], [4, 5, 6]); plt.show()    |
| <b>Bar Plot</b>          | plt.bar([1, 2, 3], [4, 5, 6]); plt.show()     |
| <b>Histogram</b>         | plt.hist([1, 2, 2, 3, 4, 5]); plt.show()      |
| <b>Scatter Plot</b>      | plt.scatter([1, 2, 3], [4, 5, 6]); plt.show() |