

Complete Python CheatSheet

- Developer Shaurya

Important Links

- **Python Cheatsheet :**

<https://developershaurya.com/python-cheatsheet/>

- **Website:** <https://developershaurya.com/>

- **All Cheatsheets :**

<https://developershaurya.com/cheatsheets/>

- **Youtube:**

<https://www.youtube.com/@DeveloperShaurya>

- **(Developer Shaurya) Community :**

https://t.me/developer_shaurya (Telegram)

CheatSheet

Python

Python CheatSheet – A Quick Guide for Beginners & Developers

by Developer Shaurya • September 9, 2025 • 0



PYTHON CHEATSHEET

BY DEVELOPER SHAURYA

This Cheatsheet covers these all topic from basic to advance concept of Python

Contents Table

- [1. Basics](#)
- [2. Data Types & Casting](#)
- [3. Collections](#)
- [4. Operators](#)
- [5. Conditionals & Loops](#)
- [6. Functions](#)
- [7. File Handling](#)
- [8. Exceptions](#)
- [9. OOP \(Classes\)](#)
- [10. Modules & Pip](#)
- [11. Decorators](#)
- [12. Generators & Iterators](#)
- [13. Context Managers](#)
- [14. Typing & Annotations](#)
- [15. Dataclasses](#)
- [16. Async / Await](#)
- [17. Regex](#)
- [18. Data Science Basics](#)
- [19. Database \(SQLite\)](#)
- [20. Testing](#)
- [21. Web \(Flask Example\)](#)

1. Basics

```
print("Hello, World!")

x, y = 10, 3.14
name = "Alice"
is_active = True

print(type(x))    # <class 'int'>
```

2. Data Types & Casting

```
int(), float(), str(), bool(), complex()

s = "Python"
print(s.lower(), s.upper(), s[::-1])  # reverse
```

3. Collections

```
# List
fruits = ["apple", "banana"]
fruits.append("mango")
fruits[0]

# Tuple
t = (1, 2, 3)

# Set
s = {1, 2, 2, 3}
s.add(4)

# Dictionary
person = {"name": "John", "age": 25}
person["city"] = "NY"
```

4. Operators

```
+, -, *, /, //, %, **    # arithmetic
==, !=, >, <, >=, <=    # comparison
and, or, not             # logical
in, not in               # membership
```

5. Conditionals & Loops

```
if x > 5:
    print("big")
elif x == 5:
    print("equal")
else:
    print("small")

for i in range(5):
    print(i)

while x > 0:
    x -= 1
```

6. Functions

```
def greet(name="Guest"):
    return f"Hello, {name}"

# Lambda
square = lambda n: n*n

# Comprehensions
evens = [x for x in range(10) if x % 2 == 0]
```

7. File Handling

```
with open("file.txt", "w") as f:
    f.write("Hello")

with open("file.txt", "r") as f:
    print(f.read())
```

8. Exceptions

```
try:
    x = 1/0
except ZeroDivisionError:
    print("Error!")
finally:
    print("Done")
```

9. OOP (Classes)

```
class Person:
    def __init__(self, name):
        self.name = name
    def greet(self):
        print(f"Hi, I'm {self.name}")

p = Person("Alice")
p.greet()
```

10. Modules & Pip

```
import math, random
print(math.sqrt(16), random.randint(1,10))

# Pip
# pip install requests
```

11. Decorators

```
def log(func):
    def wrapper(*args, **kwargs):
        print("Calling", func.__name__)
        return func(*args, **kwargs)
    return wrapper

@log
def hello():
    print("Hello")

hello()
```

12. Generators & Iterators

```
def countdown(n):  
    while n > 0:  
        yield n  
        n -= 1  
  
for num in countdown(5):  
    print(num)
```

13. Context Managers

```
class MyFile:  
    def __enter__(self): print("Open"); return self  
    def __exit__(self, *args): print("Close")  
  
with MyFile() as f:  
    print("Inside")
```

14. Typing & Annotations

```
def add(x: int, y: int) -> int:  
    return x + y
```

15. Dataclasses

```
from dataclasses import dataclass

@dataclass
class Point:
    x: int
    y: int

p = Point(1,2)
print(p)
```

16. Async / Await

```
import asyncio

async def say_hi():
    print("Hi")
    await asyncio.sleep(1)
    print("Bye")

asyncio.run(say_hi())
```

17. Regex

```
import re
txt = "Python 3.12"
match = re.search(r"\d+\.\d+", txt)
print(match.group())    # "3.12"
```

18. Data Science Basics

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

arr = np.array([1,2,3])
df = pd.DataFrame({"A": [1,2], "B": [3,4]})
plt.plot([1,2,3], [4,5,6]); plt.show()
```

19. Database (SQLite)

```
import sqlite3
con = sqlite3.connect("test.db")
cur = con.cursor()
cur.execute("CREATE TABLE IF NOT EXISTS users(id INT, name TEXT)")
cur.execute("INSERT INTO users VALUES(1, 'Alice')")
con.commit()
```

20. Testing

```
import unittest

def add(x,y): return x+y

class TestMath(unittest.TestCase):
    def test_add(self):
        self.assertEqual(add(2,3), 5)

unittest.main()
```

21. Web (Flask Example)

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def home():
    return "Hello Flask!"

if __name__ == "__main__":
    app.run()
```