

Python Cheatsheet

By Dhandapani Yedappalli Krishnamurthi Sep 2, 2025

Python

Python Programming Cheatsheet (Table Format)

1. Basics

Concept	Syntax / Example	Notes
Print	<code>print("Hello")</code>	Output to console
Comments	<code># single line</code> <code>""" multi-line """</code>	Documentation/comments
Variables	<code>x = 10</code> <code>name = "Dani"</code>	Dynamically typed
Data Types	<code>int</code> , <code>float</code> , <code>str</code> , <code>bool</code> , <code>list</code> , <code>tuple</code> , <code>dict</code> , <code>set</code>	Built-in types
Type Check	<code>type(x)</code>	Returns type
Type Casting	<code>int("5")</code> , <code>str(10)</code>	Convert between types

2. Operators

Type	Examples	Notes
Arithmetic	<code>+</code> <code>-</code> <code>*</code> <code>/</code> <code>%</code> <code>//</code> <code>**</code>	Floor division <code>//</code> , power <code>**</code>
Comparison	<code>==</code> <code>!=</code> <code>></code> <code><</code> <code>>=</code> <code><=</code>	Returns True/False
Logical	<code>and</code> <code>or</code> <code>not</code>	Combine conditions
Assignment	<code>=</code> <code>+=</code> <code>-=</code> <code>*=</code> <code>/=</code>	Shortcut assignments
Membership	<code>in</code> , <code>not in</code>	"a" in "cat" → True
Identity	<code>is</code> , <code>is not</code>	Checks memory address

3. Strings

Operation	Example	Notes
Concatenate	<code>"Hi" + " Dani"</code>	Combine
Repeat	<code>"ha" * 3</code> → "hahaha"	Repeat string
Indexing	<code>s[0]</code> , <code>s[-1]</code>	First/last char
Slicing	<code>s[1:4]</code>	Substring

Methods	s.lower(), s.upper(), s.strip(), s.replace("a","b"), s.split(", "), ".join(list)	Common ops
f-Strings	f"Name: {name}"	String interpolation

4. Lists

Operation	Example	Notes
Create	lst = [1, 2, 3]	Ordered, mutable
Access	lst[0], lst[-1]	Indexing
Slicing	lst[1:3]	Sublist
Add	lst.append(4), lst.insert(1, "x")	Insert values
Remove	lst.remove(2), lst.pop(), del lst[0]	Remove elements
Other	len(lst), sorted(lst), lst.sort(), lst.reverse()	Common ops
Comprehension	[x**2 for x in range(5)]	Powerful shorthand

5. Tuples & Sets

Type	Example	Notes
Tuple	t = (1, 2, 3)	Ordered, immutable
Set	s = {1, 2, 3}	Unique, unordered
Set Ops	s1	s2, s1 & s2, s1 - s2

6. Dictionaries

Operation	Example	Notes
Create	d = {"a":1, "b":2}	Key-value pairs
Access	d["a"], d.get("a")	Retrieve
Add/Update	d["c"] = 3	Add key
Delete	del d["a"], d.pop("b")	Remove
Keys/Values	d.keys(), d.values(), d.items()	Iteration

7. Control Flow

Type	Syntax	Example
If	if cond: ... elif cond: ... else: ...	Branching
For Loop	for i in range(5): print(i)	Iteration
While Loop	while cond: ...	Repeat
Break/Continue	Exit or skip loop	Used inside loops

8. Functions

Concept	Syntax	Notes
Define	def add(a, b): return a+b	Function definition
Default Arg	def f(x=10)	Optional args
*args	def f(*args)	Variable arguments (tuple)
kwargs	def f(kwargs)	Keyword args (dict)

Lambda	lambda x: x**2	Anonymous function
Map/Filter	map(f, lst), filter(f, lst)	Functional tools

9. Classes & OOP

Concept	Syntax	Notes
Class	class Car: ...	Blueprint
Init	def __init__(self, brand): self.brand = brand	Constructor
Object	c = Car("BMW")	Instance
Methods	def drive(self): ...	Functions in class
Inheritance	class ElectricCar(Car): ...	Reuse code
Magic Methods	__str__, __len__, __add__	Operator overloading

10. Exceptions

Concept	Syntax	Notes
Try-Except	try: ... except Exception as e: ...	Handle errors
Finally	finally: ...	Always runs
Raise	raise ValueError("Error!")	Throw exception

11. File Handling

Operation	Syntax	Notes
Open	f = open("file.txt", "r")	Modes: r,w,a,b
Read	f.read(), f.readline()	Get data
Write	f.write("Hello")	Write text
Close	f.close()	Free resources
With	with open("f.txt", "r") as f:	Auto-close

12. Modules & Packages

Concept	Syntax	Notes
Import	import math, from math import sqrt	Load module
Install	pip install package	External libs
Check Path	import sys; print(sys.path)	Module search path

13. Useful Built-ins

Function	Example
len()	len([1,2,3])
sum()	sum([1,2,3])
min(), max()	min([3,1,4])
sorted()	sorted([3,1,2])
zip()	list(zip([1,2],[3,4]))
enumerate()	for i,v in enumerate(lst): ...
any(), all()	any([True,False]), all([True,True])

14. Libraries (Quick Picks)

Library	Use
math	Math functions
datetime	Dates & times
random	Random numbers
os, sys	System interaction
json	JSON parsing
re	Regex
pandas, numpy	Data analysis
requests	HTTP requests