Basic Syntax

Comments

Single line: Use `#` symbol

Multi-line: Use triple quotes `" " "` or `'''`

Inline comments: Add `#` at end of line

Print Function

Basic output: `print()` function displays text

Multiple values: Separate with commas

String formatting: Use `f-strings, format(), or %` formatting.

Indentation

Critical in Python - Defines code blocks

Use 4 spaces - Standard convention

Consistent indentation - All lines in same block must have same indentation

Data Types

int	Whole numbers
float	Decimal Numbers
complex	Real + imaginary
str	Character strings
Immutable	Can't change chars
Unicode	International support
bool	True/False Values
list	Ordered, mutable
tuple	Ordered, immutable
dict	Key-value pairs
set	Unique elements only

Data types include numeric, text, boolean, and collections.

Naming Rules

Start	Letter or underscore
Characters	alphanumeric + underscore
No reserved words	if, for, class
Convention	snake_case

Variable Declaration

No keyword needed	Just assign
Dynamic typing	Auto-determined
Case sensitive	myvar ≠ MyVar
Multiple assignment	One line

By **Muhammad Usman Asghar** (musmankkh)

cheatography.com/musmankkh/

Published 3rd August, 2025.

Last updated 3rd August, 2025.

Formatting

f-strings	`f"Hello {name} "`
format()	`" Hello {}".f or mat (name)`
% style	`" Hello %s" % name`

String Methods

Case	upper(), lower(), title()
Search	find(), index(), count()
Check	startswith(), endswith()
Modify	replace(), strip(), split()

String Operations

Concatenation	+ operator
Repetition	* operator
Length	`en() function
Indexing	[0] zero-based
Slicing	[start:end:step]

Strings (Creation Methods)

Single quotes	'Hello'
Double quotes	"Hello"
Triple quotes	Multi Line
Raw strings	r 'literal'

Conditionals

if	Basic condition check
elif	Additional conditions
else	Default
Operators	==, !=, <, >
Logic	and, or, not

Loops

For	Iterate sequences
While	Condition-based
range()	Number Sequence
enumerate()	Index + Value
zip()	Multiple sequence

By **Muhammad Usman Asghar** (musmankkh)

cheatography.com/musmankkh/

Published 3rd August, 2025.
Last updated 3rd August, 2025.

Loop Control

break	Exit loop
continue	Skip iteration
else	Normal completion
pass	Empty placeholder

Functions

Definition

- def keyword: Define function
- Parameters: Input values
- Return: Output value
- Docstrings: Documentation

Arguments

- Positional: Order matters
- Keyword: Use parameter names
- Default: Provide default values
- Variable: *args,* *kwargs

Types

- Built-in: print(), len(), type()
- User-defined: Custom functions
- Lambda: Anonymous functions
- Generator: Uses yield

Scope

- Local: Inside function
- Global: Outside functions
- global: Modify global vars
- nonlocal: Enclosing scope

Built-in Functions

Essential

- print(): Output to console
- input(): Get user input
- len(): Sequence length
- type(): Object type
- isinstance(): Check type

Conversion

- int(), float(), str()
- list(), tuple(), set()
- ord(), chr(): Char conversion

Mathematical

Built-in Functions (cont)

• abs(): Absolute value
• round(): Round numbers
• min(), max(): Extremes
• sum(): Sequence sum
• pow(): Power calculation

Sequence

• range(): Number sequences
• enumerate(): Add indices
• zip(): Combine iterables
• sorted(): Sort sequence
• all(), any(): Boolean ops

File Handling

Operations

• open(): Open file for I/O
• Modes: 'r', 'w', 'a', 'x'
• Text/Binary: 't', 'b' modes
• close(): Close file

Reading

• read(): Entire file
• readline(): Single line
• readlines(): All lines
• Iteration: Loop through lines

Writing

write(): Write string
writelines(): Write list
Context manager: with statement

```
</d iv>
<div style= " flex: 1;">
```

Error Handling

try	Code that may fail
except	Handle exceptions
else	No exception occured
finally	Always excute
rise	Trigger exception

Handling

Error Handling	
SyntaxError	Invalid syntax
NameError	Undefined variable
TypeError	Wrong data type
ValueError	Wrong Value
IndexError	List index error
KeyError	Dict key missing

Exception Types

Object-Oriented Programming

Classes and Objects

- class keyword – Define class
- Object instantiation – Create instances
- Instance variables – Data unique to each object
- Class variables – Data shared by all instances
- self parameter – Reference to current instance

Methods

- Instance methods – Access instance data
- Class methods – Work with class data (@classmethod)
- Static methods – Independent functions (@staticmethod)
- Special methods – init, str, repr, etc.

Inheritance

- Parent class – Base class being inherited
- Child class – A class that inherits
- super() function – Access parent class methods
- Method overriding – Redefine parent methods
- Multiple inheritance – Inherit from multiple classes

Encapsulation

- Private attributes – Prefix with underscore(s)
- Property decorator – Create getter/setter methods
- Name mangling – Double underscore prefix
- Public interface – Methods meant for external use

Polymorphism

- Same interface, different implementations
- Duck typing – "If it looks like a duck..."
- Method overloading – Not directly supported
- Operator overloading – Define behavior for operators

Data Structures

Lists

• Ordered, mutable, duplicates OK
• Methods: `append(), insert(), remove()`
• Access: indexing [0], slicing [1:3]
• Sort: `sort(), reverse()`

Tuples

• Ordered, immutable, duplicates OK
• Methods: `count(), index()`
• Packing/Unpacking: Multiple assignment
• Faster than lists

Dictionaries

• Key-value pairs, mutable
• Methods: `get(), keys(), values()`
• Access: dict[key] or dict.get(key)
• Update: update(), pop()

Sets

• Unordered, unique elements
• Methods: `add(), remove(), union()`
• Operations: intersection, difference
• Membership: fast in testing

Industry Applications

Web Development

• Backend APIs – RESTful services, GraphQL
• Full-stack frameworks – Django, FastAPI
• Microservices – Flask, FastAPI
• Content management – Django CMS, Wagtail

Data Science & Analytics

• Data analysis – Pandas, NumPy
• Machine learning – Scikit-learn, TensorFlow, PyTorch
• Data visualisation – Matplotlib, Seaborn, Plotly
• Big data – PySpark, Dask

Automation & Scripting

• System administration – Automate server tasks
• Web scraping – BeautifulSoup, Scrapy
• Task automation – Schedule, monitor processes
• DevOps tools – Ansible, Fabric

Scientific Computing

• Research computing - SciPy ecosystem

Industry Applications (cont)

- Bioinformatics – BioPython
- Astronomy – AstroPy
- Physics simulations – NumPy, SciPy

Game Development

- Pygame – 2D game development
- Panda3D – 3D game engine
- Game scripting – Embed Python in games
- Game tools – Level editors, asset pipelines

Learning Resources

Official Documentation

- python.org – Official Python documentation
- PEP documents – Python Enhancement Proposals
- Library reference – Standard library documentation
- Language reference – Complete language specification

Online Tutorials

- W3Schools
- Python.org tutorial
- Real Python
- Codecademy

Practice Platforms

- LeetCode – Coding interview preparation
- HackerRank – Programming challenges
- Codewars – Coding kata and challenges
- Project Euler – Mathematical programming problems

By **Muhammad Usman Asghar** (musmankkh)

cheatography.com/musmankkh/

Published 3rd August, 2025.
Last updated 3rd August, 2025.