Python Basics Cheatsheet

Quick reference for core Python syntax: variables, data types, loops, functions and more.

Python & comments

- Python is a general-purpose language used for automation, data, web apps, scripts, and more.
- A single-line comment starts with # and everything after it on that line is ignored by Python.
- A longer explanation can be written between three quotes at the start and end, so it is treated as non-executed text.
- Use the ***print()***function to show text or values, which is helpful for checking what your code is doing.

Variables & data types

- A variable is a name that stores a value so you can reuse or change it later in your program.
- Python is dynamically typed, which means you do not declare types; the type comes from the value you assign.
- Integers represent whole numbers, often used for counts, indexes, and simple arithmetic.
- Floating-point numbers represent values with decimal points, such as measurements or percentages.
- Strings store text inside quotes, which can include letters, numbers, spaces, and punctuation.
- Boolean values are either true or false and are commonly used in conditions and logical checks.

Lists, tuples, dicts, sets

- Collections group related values together so they can be processed, searched, or looped over as a unit.
- A list is an ordered, changeable collection; you can add, remove, or update items and keep them in a specific sequence.
- A tuple is like a list but fixed once created, which makes it useful when you want a sequence that should not be modified.
- A dictionary stores key–value pairs so you look up information by a descriptive key instead of by a numeric position.
- A set stores unique values without duplicates and does not guarantee order, which is useful when you only care whether a value is present.

If / elif / else

- Conditionals let your code choose between different paths so certain blocks only run when a condition is met.
- The usual pattern is: start with ***if*** for the first test, add optional ***elif*** blocks for additional tests, and end with an optional ***else*** block for the fallback case.
- Python uses indentation to mark which statements belong to each branch, so lines at the same indent level are part of the same block.
- Typical comparisons include checking equality or difference, comparing greater or smaller values, and combining checks with logical “and” or “or”.

Loops

- Loops repeat a block of code several times so you do not have to write the same instructions repeatedly.
- A ***for*** loop steps through each item in a sequence, such as the elements of a list or the numbers produced by a range.
- A ***while*** loop continues to run as long as its condition stays true, which is useful when the number of repetitions is not fixed in advance.
- A break statement stops the current loop early, while a continue statement skips the rest of the current iteration and moves on to the next one.

Functions

- A function groups several instructions under a name so you can run them whenever you need without copying code.
- When defining a function you normally choose a clear name, list any parameters in brackets, and write the body on the indented lines below.
- Calling a function means using its name followed by brackets, optionally passing values to match the parameters.
- A return statement sends a result back to the caller so it can be stored in a variable, printed, or used in further calculations.

Handy built-ins

- ***len*** gives the number of items in a collection or characters in a string.
- ***range*** creates a sequence of numbers, often used when looping a fixed number of times.
- ***type*** shows the type of a value, such as integer, list, or dictionary.
- ***input*** reads text from the user so your program can react to their answers.