Python Basics

In this section, you'll learn basic Python. If you're completely new to Python programming, this Python basics section is perfect for you.

After completing the tutorials, you'll be confident in Python programming and be able to create simple programs in Python.

Section 1. Fundamentals

- Syntax (https://www.pythontutorial.net/python-basics/python-syntax/) – introduce you to the basic Python programming syntax.
- Variables (https://www.pythontutorial.net/python-basics/python-variables/) – explain to you what variables are and how to create concise and meaningful variables.
- Strings (https://www.pythontutorial.net/python-basics/python-string/) – learn about string data and some basic string operations.
- Numbers (https://www.pythontutorial.net/python-basics/python-numbers/) – introduce to you the commonly-used number types including integers and floating-point numbers.

- Booleans (https://www.pythontutorial.net/python-basics/python-boolean/) – explain the Boolean data type, falsy and truthy values in Python.
- Constants (https://www.pythontutorial.net/python-basics/python-constants/) – show you how to define constants in Python.
- Comments (https://www.pythontutorial.net/python-basics/python-comments/) – learn how to make notes in your code.
- Type conversion (https://www.pythontutorial.net/python-basics/python-type-conversion/) – learn how to convert a value of one type to another e.g., converting a string to a number.

Section 2. Operators

- Comparison operators (https://www.pythontutorial.net/python-basics/python-comparison-operators/) – introduce you to the comparison operators and how to use them to compare two values.
- Logical operators (https://www.pythontutorial.net/python-basics/python-logical-operators/) – show you how to use logical operators to combine multiple conditions.

Section 3. Control flow

- if...else statement (https://www.pythontutorial.net/python-basics/python-if/) – learn how to execute a code block based on a condition.
- Ternary operator (https://www.pythontutorial.net/python-basics/python-ternary-operator/) – introduce you to the Python ternary operator that makes your code more concise.
- for loop with range() (https://www.pythontutorial.net/python-basics/python-for-range/) – show you how to execute a code block for a fixed number of times by using the for loop with range() function.
- while (https://www.pythontutorial.net/python-basics/python-while/) – show you how to execute a code block as long as a condition is True.

- break (https://www.pythontutorial.net/python-basics/python-break/) – learn how to exit a loop prematurely.
- continue (https://www.pythontutorial.net/python-basics/python-continue/) – show you how to skip the current loop iteration and start the next one.
- pass (https://www.pythontutorial.net/python-pass/) – show you how to use the pass statement as a placeholder.

Section 4. Functions

- Python functions (https://www.pythontutorial.net/python-basics/python-functions/) – introduce you to functions in Python, and how to define functions, and reuse them in the program.
- Default parameters (https://www.pythontutorial.net/python-basics/python-default-parameters/) – show you how to specify the default values for function parameters.
- Keyword arguments (https://www.pythontutorial.net/python-basics/python-keyword-arguments/) – learn how to use the keyword arguments to make the function call more obvious.
- Recursive functions (https://www.pythontutorial.net/python-basics/python-recursive-functions/) – learn how to define recursive functions in Python.
- Lambda Expressions (https://www.pythontutorial.net/python-basics/python-lambda-expressions/) – show you how to define anonymous functions in Python using lambda expressions.
- Docstrings (https://www.pythontutorial.net/python-basics/python-function-docstrings/) – show you how to use docstrings to document a function.

Section 5. Lists

- List (https://www.pythontutorial.net/python-basics/python-list/) – introduce you to the list type and how to manipulate list elements effectively.
- Tuple (https://www.pythontutorial.net/python-basics/python-tuples/) – introduce you to the tuple which is a list that doesn't change throughout the program.

- Sort a list in place (https://www.pythontutorial.net/python-basics/python-sort-list/) – show you how to use the sort() method to sort a list in place.
- Sort a List (https://www.pythontutorial.net/python-basics/python-sorted/) – learn how to use the sorted() function to return a new sorted list from the original list.
- Slice a List (https://www.pythontutorial.net/python-basics/python-list-slice/) – show you how to use the list slicing technique to manipulate lists effectively.
- Unpack a list (https://www.pythontutorial.net/python-basics/python-unpack-list/) – show you how to assign list elements to multiple variables using list unpacking.
- Iterate over a List (https://www.pythontutorial.net/python-basics/python-for-loop-list/) – learn how to use a for loop to iterate over a list.
- Find the index of an element (https://www.pythontutorial.net/python-basics/python-find-index-of-element-in-list/) – show you how to find the index of the first occurrence of an element in a list.
- Iterables (https://www.pythontutorial.net/python-basics/python-iterables/) – explain to you iterables, and the difference between an iterable and an iterator.
- Transform list elements with map() (https://www.pythontutorial.net/python-basics/python-map-list/) – show you how to use the map() function to transform list elements.
- Filter list elements with filter() (https://www.pythontutorial.net/python-basics/python-filter-list/) – use the filter() function to filter list elements.
- Reduce list elements into a value with reduce() (https://www.pythontutorial.net/python-basics/python-reduce-list/) – use the reduce() function to reduce list elements into a single value.
- List comprehensions (https://www.pythontutorial.net/python-basics/python-list-comprehensions/) – show you how to create a new list based on an existing list.

Section 6. Dictionaries

- Dictionary (https://www.pythontutorial.net/python-basics/python-dictionary/) – introduce you to the dictionary type.

- Dictionary comprehension (https://www.pythontutorial.net/python-basics/python-dictionary-comprehension/) – show you how to use dictionary comprehension to create a new dictionary from an existing one.

Section 7. Sets

- Set (https://www.pythontutorial.net/python-basics/python-set/) – explain to you the Set type and show you how to manipulate set elements effectively.
- Set comprehension (https://www.pythontutorial.net/python-basics/python-set-comprehension/) – explain to you the set comprehension so that you can create a new set based on an existing set with a more concise and elegant syntax.
- Union of Sets (https://www.pythontutorial.net/python-basics/python-set-union/) – show you how to union two or more sets using the `union()` method or set union operator (`|`).
- Intersection of Sets (https://www.pythontutorial.net/python-basics/python-set-intersection/) – show you how to intersect two or more sets using the intersection() method or set intersection operator (`&`).
- Difference of sets (https://www.pythontutorial.net/python-basics/python-set-difference/) – learn how to find the difference between sets using the set `difference()` method or set difference operator (`-`)
- Symmetric Difference of sets (https://www.pythontutorial.net/python-basics/python-symmetric-difference/) – guide you on how to find the symmetric difference of sets using the `symmetric_difference()` method or the symmetric difference operator (`^`).
- Subset (https://www.pythontutorial.net/python-basics/python-issubset/) – check if a set is a subset of another set.
- Superset (https://www.pythontutorial.net/python-basics/python-issuperset/) – check if a set is a superset of another set.
- Disjoint sets (https://www.pythontutorial.net/python-basics/python-disjoint-sets/) – check if two sets are disjoint.

Section 8. Exception handling

- try...except (https://www.pythontutorial.net/python-basics/python-try-except/) – show you how to handle exceptions more gracefully using the try...except statement.
- try...except...finally (https://www.pythontutorial.net/python-basics/python-try-except-finally/) – learn how to execute a code block whether an exception occurs or not.
- try...except...else (https://www.pythontutorial.net/python-basics/python-try-except-else/) – explain to you how to use the try...except...else statement to control the follow of the program in case of exceptions.

Section 9. More on Python Loops

- for...else (https://www.pythontutorial.net/python-basics/python-for-else/) – explain to you the `for else` statement.
- while...else (https://www.pythontutorial.net/python-basics/python-while-else/) – discuss the `while else` statement.
- do...while (https://www.pythontutorial.net/python-basics/python-do-while/) loop emulation – show you how to emulate the do...while loop in Python by using the while loop statement.

Section 10. More on Python functions

- Unpacking tuples (https://www.pythontutorial.net/python-basics/python-unpacking-tuple/) – show you how to unpack a tuple that assigns individual elements of a tuple to multiple variables.
- `*args` Parameters (https://www.pythontutorial.net/python-basics/python-args/) – learn how to pass a variable number of arguments to a function.
- `**kwargs` Parameters (https://www.pythontutorial.net/python-basics/python-kwargs/) – show you how to pass a variable number of keyword arguments to a function.

- Partial functions (https://www.pythontutorial.net/python-basics/python-partial-functions/) – learn how to define partial functions.
- Type hints (https://www.pythontutorial.net/python-basics/python-type-hints/) – show you how to add type hints to the parameters of a function and how to use the static type checker (mypy) to check the type statically.

Section 11. Modules & Packages

- Modules (https://www.pythontutorial.net/python-basics/python-module/) – introduce you to the Python modules and show you how to write your own modules.
- Module search path (https://www.pythontutorial.net/python-basics/python-module-search-path/) – explain to you how the Python module search path works when you import a module.
- `__name__` variable (https://www.pythontutorial.net/python-basics/python-__name__/) – show you how to use the `__name__` variable to control the execution of a Python file as a script or as a module.
- Packages (https://www.pythontutorial.net/python-basics/python-packages/) – learn how to use packages to organize modules in more structured ways.

Section 12. Working with files

- Read from a text file (https://www.pythontutorial.net/python-basics/python-read-text-file/) – learn how to read from a text file.
- Write to a text file (https://www.pythontutorial.net/python-basics/python-write-text-file/) – show you how to write to a text file.
- Create a new text file (https://www.pythontutorial.net/python-basics/python-create-text-file/) – walk you through the steps of creating a new text file.
- Check if a file exists (https://www.pythontutorial.net/python-basics/python-check-if-file-exists/) – show you how to check if a file exists.

- Read CSV files (https://www.pythontutorial.net/python-basics/python-read-csv-file/) – show you how to read data from a CSV file using the csv module.
- Write CSV files (https://www.pythontutorial.net/python-basics/python-write-csv-file/) – learn how to write data to a CSV file using the csv module.
- Rename a file (https://www.pythontutorial.net/python-basics/python-rename-file/) – guide you on how to rename a file.
- Delete a file (https://www.pythontutorial.net/python-basics/python-delete-file/) – show you how to delete a file.

Section 13. Working Directories

- Working with directories (https://www.pythontutorial.net/python-basics/python-directory/) – show you commonly used functions to work with directories.
- List files in a Directory (https://www.pythontutorial.net/python-basics/python-list-files/) – list files in a directory.

Section 14. Third-party Packages, PIP, and Virtual Environments

- Python Package Index (PyPI) and pip (https://www.pythontutorial.net/python-basics/python-pip/) – introduce you to the Python package index and how to install third-party packages using pip.
- Virtual Environments (https://www.pythontutorial.net/python-basics/python-virtual-environments/) – understand Python virtual environments and more importantly, why you need them.
- Install pipenv on Windows (https://www.pythontutorial.net/python-basics/install-pipenv-windows/) – show you how to install the `pipenv` tool on Windows.

Section 15. Strings

- F-strings (https://www.pythontutorial.net/python-basics/python-f-strings/) – learn how to use the f-strings to format text strings in a clear syntax.
- Raw strings (https://www.pythontutorial.net/python-basics/python-raw-strings/) – use raw strings to handle strings that contain the backslashes.
- Backslash (https://www.pythontutorial.net/python-basics/python-backslash/) – explain how Python uses the backslashes (`\`) in string literals.