# Advanced Python



This tutorial series explains the advanced Python concepts and helps you understand how and why things work in Python under the hood.

To learn advanced Python, you need to have basic Python (https://www.pythontutorial.net/python-basics/) knowledge and some practical experience in Python programming.

## Section 1. Variables & Memory Management

- References (https://www.pythontutorial.net/advanced-python/python-references/) – learn about references and how reference counting works in Python.
- Garbage collection (https://www.pythontutorial.net/advanced-python/python-garbage-collection/) – understand the garbage collection and how to interact with Python Garbage collector via the `gc` module.
- Dynamic typing (https://www.pythontutorial.net/advanced-python/dynamic-typing-in-python/) – explain to you how dynamic typing works and understand the differences between static types and dynamic types.

- Mutable & Immutable objects (https://www.pythontutorial.net/advanced-python/python-mutable-and-immutable/) – introduce you to mutable and immutable objects in Python.
- is operator (https://www.pythontutorial.net/advanced-python/python-is-operator/) – help you understand object identity and equality, and how to use the `is` operator to check if two variables reference the same object.
- None (https://www.pythontutorial.net/advanced-python/python-none/) – learn about the None object and how to use it properly.

## Section 2. Integer types

- Integers (https://www.pythontutorial.net/advanced-python/python-integers/) – learn about the integer and how Python stores the integers in the memory.
- Floor division operator (//) (https://www.pythontutorial.net/advanced-python/python-floor-division/) – introduce you to the floor division operator (//) and how to use it effectively.
- Modulo operator (%) (https://www.pythontutorial.net/advanced-python/python-modulo/) – explain how the module operator (%) works in Python.
- `bool` (https://www.pythontutorial.net/advanced-python/python-bool/) – explain how Python boolean works under the hood.
- The `and` operator (https://www.pythontutorial.net/advanced-python/python-and/) – learn how to use the `and` operator effectively.
- The `or` operator (https://www.pythontutorial.net/advanced-python/python-or/) – show you how to use the `or` operator.

## Section 3. Float

- Float (https://www.pythontutorial.net/advanced-python/python-float/) – explain how Python represents floating-point numbers internally and how to test two floats for equality.

- Converting float to int (https://www.pythontutorial.net/advanced-python/python-float-to-int/) – show you how to convert float to int.
- Rounding (https://www.pythontutorial.net/advanced-python/python-rounding/) – learn how to round a floating-point number to a specified number of digits after the decimal point.

## Section 4. Decimal

- Decimal (https://www.pythontutorial.net/advanced-python/python-decimal/) – learn about the `decimal` module that provides support for fast correctly-rounded decimal floating-point arithmetic.

## Section 5. Variable scopes

- Variable scopes (https://www.pythontutorial.net/advanced-python/python-variable-scopes/) – explain to you the variable scopes and help understand the built-in, local, and global variables.
- Nonlocal scopes and nonlocal variables (https://www.pythontutorial.net/advanced-python/python-nonlocal/) – understand the nonlocal scopes and how to change variables of the nonlocal scopes using the nonlocal keyword.

## Section 6. Closures

- Closures (https://www.pythontutorial.net/advanced-python/python-closures/) – help you understand the closures in Python and how to define closures.

## Section 7. Decorators

- Decorators (https://www.pythontutorial.net/advanced-python/python-decorators/) – explain to you the decorator and show you how to develop a simple decorator in Python.

- Decorators with arguments (https://www.pythontutorial.net/advanced-python/python-decorator-arguments/) – show you how to define a decorator that accepts one or more arguments.
- Class Decorators (https://www.pythontutorial.net/advanced-python/python-class-decorators/) – illustrate how to define a class as a decorator.

## Section 8. Named Tuples

- Named Tuples (https://www.pythontutorial.net/advanced-python/python-namedtuple/) – learn how to use named tuples.

## Section 9. Sequence Types

- Sequence types (https://www.pythontutorial.net/advanced-python/python-sequences/) – learn about sequences and their basic operations
- Lists vs. Tuples (https://www.pythontutorial.net/advanced-python/python-tuple-vs-list/) – explain the main differences between the tuple and list.
- Slicing (https://www.pythontutorial.net/advanced-python/python-slicing/) – show you how to use slicing to extract data from or assign data to a sequence.
- Custom Sequence Type (https://www.pythontutorial.net/advanced-python/python-fibonacci-sequence/) – learn about the custom sequence type and show you how to use a custom sequence type to define the Fibonacci sequence.

## Section 10. Iterators and Iterables

- Iterators (https://www.pythontutorial.net/advanced-python/python-iterators/) – learn about the iterator protocol and how to define a custom iterator.
- Iterators vs. Iterables (https://www.pythontutorial.net/advanced-python/python-iterator-vs-iterable/) – understand iterators and iterables, and the differences between them

- iter() (https://www.pythontutorial.net/advanced-python/python-iter/) – explain to you how the iter() function works and how to use it effectively.

## Section 11. Generators

- Generator functions (https://www.pythontutorial.net/advanced-python/python-generators/) – introduce you to the generator functions and how to use generators to create iterators.
- Generator expressions (https://www.pythontutorial.net/advanced-python/python-generator-expressions/) – show you an alternative syntax for creating a generator object.

## Section 12. Context Managers

- Context Managers (https://www.pythontutorial.net/advanced-python/python-context-managers/) – learn about context managers and how to use them effectively.