

The Python Language Reference

This reference manual describes the syntax and “core semantics” of the language. It is terse, but attempts to be exact and complete. The semantics of non-essential built-in object types and of the built-in functions and modules are described in [The Python Standard Library](#). For an informal introduction to the language, see [The Python Tutorial](#). For C or C++ programmers, two additional manuals exist: [Extending and Embedding the Python Interpreter](#) describes the high-level picture of how to write a Python extension module, and the [Python/C API Reference Manual](#) describes the interfaces available to C/C++ programmers in detail.

- 1. Introduction
 - [1.1. Alternate Implementations](#)
 - [1.2. Notation](#)
- 2. Lexical analysis
 - [2.1. Line structure](#)
 - [2.2. Other tokens](#)
 - [2.3. Identifiers and keywords](#)
 - [2.4. Literals](#)
 - [2.5. Operators](#)
 - [2.6. Delimiters](#)
- 3. Data model
 - [3.1. Objects, values and types](#)
 - [3.2. The standard type hierarchy](#)
 - [3.3. Special method names](#)
 - [3.4. Coroutines](#)
- 4. Execution model
 - [4.1. Structure of a program](#)
 - [4.2. Naming and binding](#)
 - [4.3. Exceptions](#)
- 5. The import system
 - [5.1. `importlib`](#)
 - [5.2. Packages](#)
 - [5.3. Searching](#)
 - [5.4. Loading](#)
 - [5.5. The Path Based Finder](#)
 - [5.6. Replacing the standard import system](#)
 - [5.7. Package Relative Imports](#)
 - [5.8. Special considerations for `__main__`](#)
 - [5.9. Open issues](#)
 - [5.10. References](#)
- 6. Expressions
 - [6.1. Arithmetic conversions](#)
 - [6.2. Atoms](#)
 - [6.3. Primaries](#)
 - [6.4. Await expression](#)

- 6.5. The power operator
- 6.6. Unary arithmetic and bitwise operations
- 6.7. Binary arithmetic operations
- 6.8. Shifting operations
- 6.9. Binary bitwise operations
- 6.10. Comparisons
- 6.11. Boolean operations
- 6.12. Assignment expressions
- 6.13. Conditional expressions
- 6.14. Lambdas
- 6.15. Expression lists
- 6.16. Evaluation order
- 6.17. Operator precedence
- 7. Simple statements
 - 7.1. Expression statements
 - 7.2. Assignment statements
 - 7.3. The `assert` statement
 - 7.4. The `pass` statement
 - 7.5. The `del` statement
 - 7.6. The `return` statement
 - 7.7. The `yield` statement
 - 7.8. The `raise` statement
 - 7.9. The `break` statement
 - 7.10. The `continue` statement
 - 7.11. The `import` statement
 - 7.12. The `global` statement
 - 7.13. The `nonlocal` statement
- 8. Compound statements
 - 8.1. The `if` statement
 - 8.2. The `while` statement
 - 8.3. The `for` statement
 - 8.4. The `try` statement
 - 8.5. The `with` statement
 - 8.6. Function definitions
 - 8.7. Class definitions
 - 8.8. Coroutines
- 9. Top-level components
 - 9.1. Complete Python programs
 - 9.2. File input
 - 9.3. Interactive input
 - 9.4. Expression input
- 10. Full Grammar specification