Cheatography

# Python Core Language Cheat Sheet

by Venkatesh Jeyakumaresan (codegutsy) via cheatography.com/122776/cs/23864/

## Lexical Analysis

Line Structure

Joining Lines

Blank Lines

Comments

Encoding Declaration

Indentation

## Input and Output

| | | |
|---|---|---|
| Printing output | print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False) | print("Hello world!", end=" ") |
| Getting input | input([prompt]) | input("Enter your age") |

## Indentation

Leading whitespace (spaces and tabs) at the beginning of a logical line is used to compute the indentation level of the line, which in turn is used to determine the grouping of statements.

## Encoding Declaration

| | |
|---|---|
| If a comment in the first or second line of the Python script matches the regular expression, this comment is processed as an encoding declaration; the first group of this expression names the encoding of the source code file. | #!/usr/bin/env python<br># -*- coding: utf-8 -*- |

## Comments

| | | |
|---|---|---|
| Hash | # | # some comment |
| DocString | """<br>""" | """Demonstrates docstrings and does nothing really.""" |

## Blank Lines

A logical line that contains only spaces, tabs, formfeeds and possibly a comment, is ignored.

During interactive input of statements, handling of a blank line may differ depending on the implementation of the read-eval-print loop.

In the standard interactive interpreter, an entirely blank logical line terminates a multi-line statement.

## Joining Lines

| | | |
|---|---|---|
| Explicit line joining | Two or more physical lines may be joined into logical lines using backslash characters (\), as follows: when a physical line ends in a backslash that is not part of a string literal or comment, it is joined with the following forming a single logical line, deleting the backslash and the following end-of-line character. | if 1900 < year < 2100 and 1 <= month <= 12 \ and 1 <= day <= 31 and 0 <= hour < 24 \ and 0 <= minute < 60 and 0 <= second < 60: # Looks like a valid date return 1 |

## Joining Lines (cont)

| | | |
|---|---|---|
| Implicit line joining | Expressions in parentheses, square brackets or curly braces can be split over more than one physical line without using backslashes. | month_names = ['Januari', 'Februari', 'Maart', # These are the 'April', 'Mei', 'Juni', # Dutch names 'Juli', 'Augustus', 'September', # for the months 'Oktober', 'November', 'December'] # of the year |

Remember:
-> A line ending in a backslash cannot carry a comment.
-> A backslash does not continue a comment.
-> A backslash does not continue a token except for string literals (i.e., tokens other than string literals cannot be split across physical lines using a backslash).
-> A backslash is illegal elsewhere on a line outside a string literal.

## Line Structure

| | | |
|---|---|---|
| Logical Lines | A logical line is what Python sees as a single statement | Here is one logical on a physical line: 1) my_list = [1, 2, 3, 4] 2) print("-Hello world!") |
| Physical Lines | A physical line is what you see when you write the program. Two or more physical lines may be joined into logical lines using backslash characters. | Here is one logical line on two physical lines: 1) my_list = [1, 2, 3, 4] 2) s = 'This is a string. \ This continues the string.' |

## Introduction

| | |
|---|---|
| Founder | **Guido van Rossum** |
| First release | February 20, 1991 |
| Developers | *Python Software Foundation* |
| Paradigm | Multi-paradigm: functional, imperative, object-oriented, structured, reflective |
| Type Discipline | Duck, dynamic, gradual |
| Language Type | Interpreted |

## Identifiers

| | | |
|---|---|---|
| Lu - uppercase letters | A..Z | UPPERCASENAME |
| Ll - lowercase letters | a...z | LOWERCASENAME |
| Lt - titlecase letters | a..zA..z | titleCaseName |
| Nl - letter numbers | a...zA...z-1...n | name2000 \| NAME999 |
| Pc - connector punctuations | a_zA_Z | name_connector \| NAME_CONNECTOR |

A Python identifier is a name used to identify a variable, function, class, module or other object.
Identifier naming standards are mentioned above.

## Keywords and Reserved words

| | | | |
|---|---|---|---|
| False | await | else | import |
| pass | None | break | except |
| in | raise | True | class |
| finally | is | return | and |
| continue | for | lambda | try |
| as | def | from | nonlocal |
| while | assert | del | global |
| not | with | async | elif |
| if | or | yield | |

Keywords cannot be used as ordinary identifiers.

Cheatography

# Python Core Language Cheat Sheet

by Venkatesh Jeyakumaresan (codegutsy) via cheatography.com/122776/cs/23864/

## Variables

| | |
|---|---|
| Definition | Variable is a reserved memory location to store values |
| Syntax | IDENTIFIER_NAME = VALUE(1) |
| Assignment Operator | = |
| Walrus Operator/Assignment Expressions | := |
| Example | number = 10 \| name = "Some name" \| print(walrus := True) |
| Multiple Assignment | a = b = c = 1 \| a,b,c = 1,2,"john" |
| Deleting variable | del VARIABLE_NAME |

(1) VALUE => any built-in type or user-defined class or function

## Built-in Types

| | | |
|---|---|---|
| Boolean types | Truthy Falsy values | Scalar |
| Numeric types | Integer, Float, Complex | Scalar |
| Sequence types | List, Tuple, Range | Compound |
| Text Sequence types | String | Scalar |
| Mapping types | dictionary | Compound |
| Binary Sequence types | bytes, bytesarray, memoryview | Compound |
| Set types | set, frozenset | Compound |
| Other types | functions, modules, class, methods, code objects, type objects, elipsis object, null object, notImplemented objects | |

Special attributes

## Boolean Type

| | |
|---|---|
| Reference | Link |

## Numeric Type

| | |
|---|---|
| Integer | Link |
| Float | Link |
| Complex | Link |

## Sequence Types

| | |
|---|---|
| Common sequence operations | Link |
| List | Link |
| Tuple | Link |
| Range | LInk |
| String | Link |

## Mapping type

| | |
|---|---|
| Dictionary | Link |

## Binary Sequence Types

| | |
|---|---|
| Bytes | Link |
| Bytesarray | Link |
| Memoryview | Link |

## Sets ypes

| | |
|---|---|
| Set | Link |
| Frozenset | Link |

## Other Types

| | |
|---|---|
| Functions | Link |
| Class | Link |
| Modules | Link |
| Elipsis | Link |
| Null | Link |
| NotImplemented | Link |
| Special Attributes | Link |

By **Venkatesh Jeyakumaresan** (codegutsy)
cheatography.com/codegutsy/

Not published yet.
Last updated 29th July, 2020.