Python

{ programming language

- **Python** is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

- Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

- Python was conceived in the late 1980s, and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to the ABC language (itself inspired by SETL)capable of exception handling and interfacing with the Amoeba operating system. Van Rossum is Python's principal author.

The core philosophy of the language is summarized by the document "PEP 20 (The Zen of Python)", which includes aphorisms such as:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

- The if statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).
- The for statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
- The while statement, which executes a block of code as long as its condition is true.

- **Methods**
- Methods on objects are functions attached to the object's class; the syntax instance.method(argument) is, for normal methods and functions, syntactic sugar for Class.method(instance, argument). Python methods have an explicit self parameter to access instance data, in contrast to the implicit self (or this) in some other object-oriented programming languages (e.g. C++, Java, Objective-C, or Ruby).

Summary of Python 3's built-in types

Type	Mutable	Description	Syntax example

`str`	Immutable	A character string: Sequence of Unicode codepoints.	`'Wikipedia'`
			`"Wikipedia"`
			`"""Spanning multiple lines"""`

`bytearray`	Mutable	Sequence of bytes.	`bytearray(b'Some ASCII')`
			`bytearray(b"Some ASCII")`
			`bytearray([119, 105, 107, 105])`

`bytes`	Immutable	Sequence of bytes.	`b'Some ASCII'`
			`b"Some ASCII"`
			`bytes([119, 105, 107, 105])`

`list`	Mutable	List, can contain mixed types.	`[4.0, 'string', True]`
`tuple`	Immutable	Can contain mixed types.	`(4.0, 'string', True)`
`set`	Mutable	Unordered set, contains no duplicates. Can contain mixed types as long as they are hashable.	`{4.0, 'string', True}`
`frozenset`	Immutable	Unordered set, contains no duplicates. Can contain mixed types as long as they are hashable.	`frozenset([4.0, 'string', True])`
`dict`	Mutable	Associative array (or dictionary) of key and value pairs. Can contain mixed types (keys and values). Keys must be a hashable type.	`{'key1': 1.0, 3: False}`
`int`	Immutable	Integer of unlimited magnitude.[65]	`42`
`float`	Immutable	Floating point number (system-defined precision).	`3.1415927`
`complex`	Immutable	Complex number with real and imaginary parts.	`3+2.7j`
`bool`	Immutable	Boolean value.	`True` `False`
`ellipsis`		An ellipsis placeholder to be used as an index in NumPy arrays.	`...`

- Since 2003, Python has consistently ranked in the top ten most popular programming languages as measured by the TIOBE Programming Community Index.