ANNU YADAV

FYCS

ROLL NO 43

Practical – 07
Implementing coding practices in Python using PEP8

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.[30]

Python

Paradigm, Multi-paradigm: object-oriented,[1] procedural (imperative), functional, structured, reflective

Designed by, Guido van Rossum

Developer, Python Software Foundation

First appeared, 20 February 1991; 31 years ago[2]

Stable release, 3.10.3[3] / 16 March 2022; 2 days ago

Preview release, 3.11.0a6[4] / 7 March 2022; 11 days ago

Typing discipline, Duck, dynamic, strong typing;[5] gradual (since 3.5, but ignored in CPython)[6]

OS, Windows, Linux/UNIX, macOS and more[7]

License, Python Software Foundation License

Filename extensions, .py, .pyi, .pyc, .pyd, .pyo (prior to 3.5),[8] .pyw, .pyz (since 3.5)[9]

Website, www.python.org

Major implementations,

CPython, PyPy, Stackless Python, MicroPython, CircuitPython, IronPython, Jython,

Dialects,

Cython, RPython, Starlark[10],

Influenced by,

ABC,[11] Ada,[12] ALGOL 68,[13] APL,[14] C,[15] C++,[16] CLU,[17] Dylan,[18] Haskell,[19] Icon,[20] Lisp,[21] Modula-3,[16] Perl, Standard ML[14],

Influenced,

Apache Groovy, Boo, Cobra, CoffeeScript,[22] D, F#, Genie,[23] Go, JavaScript,[24][25] Julia,[26] Nim, Ring,[27] Ruby,[28] Swift[29],

ANNU YADAV

FYCS

ROLL NO 43 Python Programming at Wikibooks,

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.[31][32]

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0.[33] Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.[3

Python consistently ranks as one of the most popular programming languages.

History

Python was conceived in the late 1980s[39] by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL,[40] capable of exception handling and interfacing with the Amoeba operating system.[11] Its implementation began in December 1989.[41] Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker.[42] In January 2019, active Python core developers elected a five-member Steering Council to lead the project.[43][44]

Python 2.0 was released on 16 October 2000, with many major new features.[45] Python 3.0, released on 3 December 2008, with many of its major features backported to Python 2.6.x[46] and 2.7.x. Releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.[47]

Python 2.7's end-of-life was initially set for 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.[48][49] No further security patches or other improvements will be released for it.[50][51] With Python 2's end-of-life, only Python 3.6.x[52] and later are supported.

Python 3.9.2 and 3.8.8 were expedited[53] as all versions of Python (including 2.7[54]) had security issues leading to possible remote code execution[55] and web cache poisoning.[5

Design philosophy and features

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming[57] and metaobjects [magic methods]).[58] Many other paradigms are supported via extensions, including design by contract[59][60] and logic programming.[61]

ANNU YADAV

FYCS

ROLL NO 43

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management.[62] It uses dynamic name resolution (late binding), which binds method and variable names during program execution.

Its design offers some support for functional programming in the Lisp tradition. It has filter,mapandreduce functions; list comprehensions, dictionaries, sets, and generator expressions.[63] The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.[64]

Its core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:[65]

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Readability counts.

Rather than building all of its functionality into its core, Python was designed to be highly extensible via modules. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.[39]

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" philosophy.[65] Alex Martelli, a Fellow at the Python Software Foundation and Python book author, wrote: "To describe something as 'clever' is not considered a compliment in the Python culture."[66]

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity.[67] When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C; or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

Python's developers aim for it to be fun to use. This is reflected in its name—a tribute to the British comedy group Monty Python[68]—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (a reference to a Monty Python sketch) instead of the standard foo and bar.[69][70]

A common neologism in the Python community is pythonic, which has a wide range of meanings related to program style. "Pythonic" code may use Python idioms well, be natural or show fluency in the language, or conform with Python's minimalist philosophy and emphasis on readability. Code that is difficult to understand or reads like a rough transcription from another programming language is called unpythonic.[71][72]

Python users and admirers, especially those considered knowledgeable or experienced, are often referred to as Pythonistas.[73][74]

Syntax and semantics

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but rarely used. It has fewer syntactic exceptions and special cases than C or Pascal.[7]

Expression

Some Python expressions are similar to those in languages such as C and Java, while some are not:

Addition, subtraction and multiplication are the same, but the behavior of division differs. There are two types of divisions in Python: floor division (or integer division) // and floating-point/division.[84] Python also uses the ** operator for exponentiation.

The @ infix operator was introduced in Python 3.5. It is intended to be used by libraries such as NumPy for matrix multiplication.[85][86]

The syntax :=, called the "walrus operator", was introduced in Python 3.8. It assigns values to variables as part of a larger expression.[87]

In Python, == compares by value, versus Java, which compares numerics by value[88] and objects by reference.[89] Python's is operator may be used to compare object identities (comparison by reference), and comparisons may be chained—for example, a <= b <= c.

Python uses and, or, and not as boolean operators rather than the symbolic &&, ||, ! in Java and C.

Python has a type of expression called a list comprehension, as well as a more general expression called a generator expression.[63]

Anonymous functions are implemented using lambda expressions; however, there may be only one expression in each body.

Conditional expressions are written as x if c else y[90] (different in order of operands from the c ? x : y operator common to many other languages).

Python makes a distinction between lists and tuples. Lists are written as [1, 2, 3], are mutable, and cannot be used as the keys of dictionaries (dictionary keys must be immutable in Python). Tuples, written as (1, 2, 3), are immutable and thus can be used as keys of dictionaries, provided all of the tuple's elements are immutable. The + operator can be used to concatenate two tuples, which does not directly modify their contents, but produces a new tuple containing the elements of both. Thus, given the variable t initially equal to (1, 2, 3), executing t = t + (4, 5) first evaluates t + (4, 5), which yields (1, 2, 3, 4, 5), which is then assigned back to t—thereby effectively "modifying the contents" of t while conforming to the immutable nature of tuple objects. Parentheses are optional for tuples in unambiguous contexts.[91]

Python features sequence unpacking where multiple expressions, each evaluating to anything that can be assigned (to a variable, writable property, etc.) are associated in an identical manner to that forming tuple literals—and, as a whole, are put on the left-hand side of the equal sign in an assignment statement. The statement expects an iterable object on the right-hand side of the equal sign that produces the same number of values as the provided writable expressions; when iterated through them, it assigns each of the produced values to the corresponding expression on the left.[92]

Python has a "string format" operator % that functions analogously to printf format strings in C—e.g. "spam=%s eggs=%d" % ("blah", 2) evaluates to "spam=blah eggs=2". In Python 2.6+ and 3+, this was supplemented by the format() method of the str class, e.g. "spam={0} eggs={1}".format("blah", 2). Python 3.6 added "f-strings": blah = "blah"; eggs = 2; f'spam={blah} eggs={eggs}'.[93]

Strings in Python can be concatenated by "adding" them (with the same operator as for adding integers and floats), e.g. "spam" + "eggs" returns "spameggs". If strings contain numbers, they are added as strings rather than integers, e.g. "2" + "2" returns "22".

Python has various string literals:

Delimited by single or double quote marks. Unlike in Unix shells, Perl and Perl-influenced languages, single and double quote marks function identically. Both use the backslash (\) as an escape character. String interpolation became available in Python 3.6 as "formatted string literals".[93]

Triple-quoted (beginning and ending with three single or double quote marks), which may span multiple lines and function like here documents in shells, Perl and Ruby.

Raw string varieties, denoted by prefixing the string literal with r. Escape sequences are not interpreted; hence raw strings are useful where literal backslashes are common, such as regular expressions and Windows-style paths. (Compare "@-quoting" in C#.)

Python has array index and array slicing expressions in lists, denoted as a[key], a[start:stop] or a[start:stop:step]. Indexes are zero-based, and negative indexes are relative to the end. Slices take elements from the start index up to, but not including, the stop index. The third slice parameter, called step or stride, allows elements to be skipped and reversed. Slice indexes may be omitted—for example a[:] returns a copy of the entire list. Each element of a slice is a shallow copy.

In Python, a distinction between expressions and statements is rigidly enforced, in contrast to languages such as Common Lisp, Scheme, or Ruby. This leads to duplicating some functionality. For example:

List comprehensions vs. for-loops

Conditional expressions vs. if blocks

ANNU YADAV

FYCS

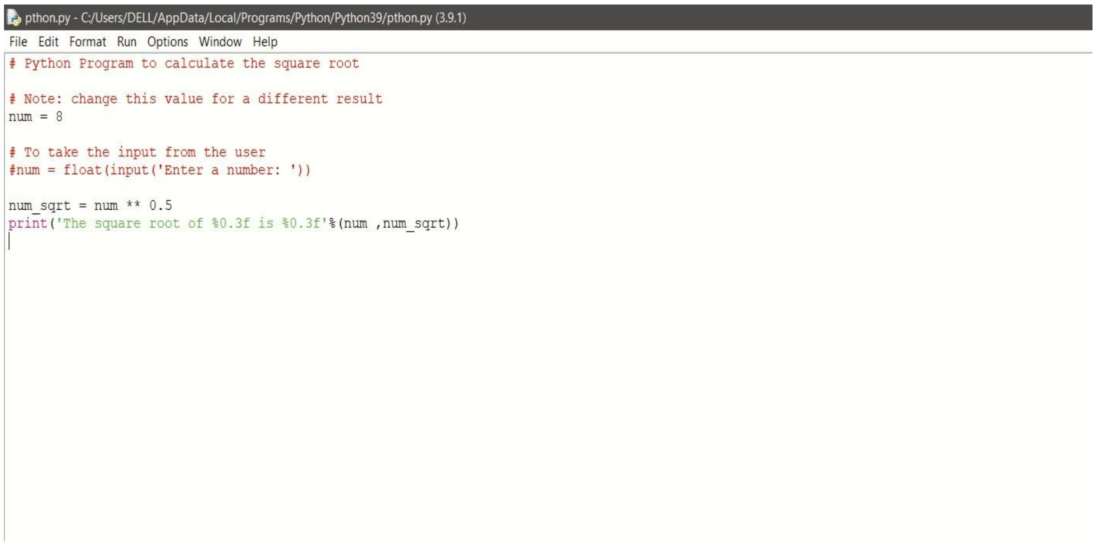
ROLL NO 43

Practical - 07

The eval() vs. exec() built-in functions (in Python 2, exec is a statement); the former is for expressions, the latter is for statements

Statements cannot be a part of an expression—so list and other comprehensions or lambda expressions, all being expressions, cannot contain statements. A particular case is that an assignment statement such as a = 1 cannot form part of the conditional expression of a conditional statement. This has the advantage of avoiding a classic C error of mistaking an assignment operator = for an equality operator == in conditions: if (c = 1) { ... } is syntactically valid (but probably unintended) C code, but if c = 1: ... causes a syntax error in Python

Code

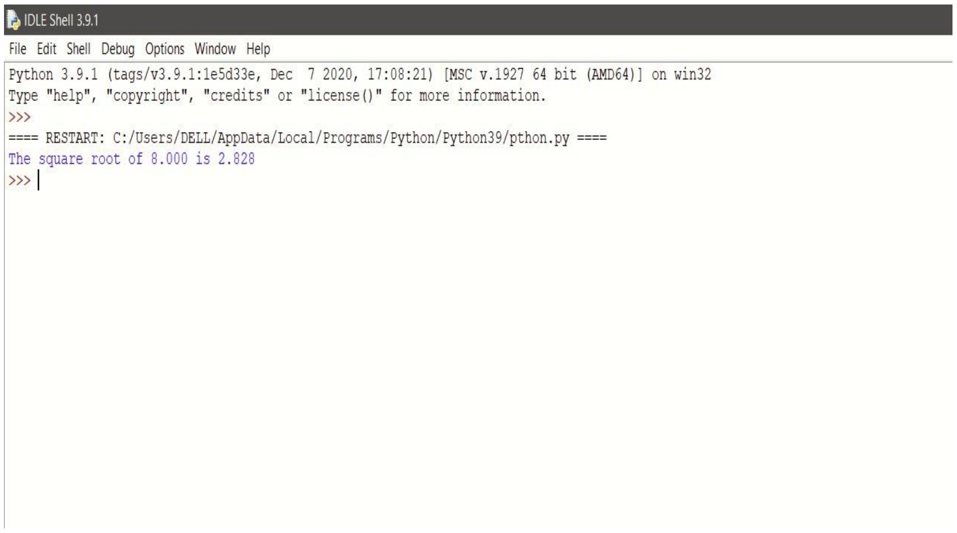

pthon.py - C:/Users/DELL/AppData/Local/Programs/Python/Python39/pthon.py (3.9.1)
File Edit Format Run Options Window Help

```
# Python Program to calculate the square root

# Note: change this value for a different result
num = 8

# To take the input from the user
#num = float(input('Enter a number: '))

num_sqrt = num ** 0.5
print('The square root of %0.3f is %0.3f'%(num ,num_sqrt))
```



```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python39/pthon.py ====
The square root of 8.000 is 2.828
>>>
```

Thank you

ANNU YADAV

FYCS

ROLL NO 43