

Introduction to Python

NIELIT Chandigarh



Python was conceived in the late 1980s, & implementation began in December 1989 by **Guido van Rossum**



Agenda

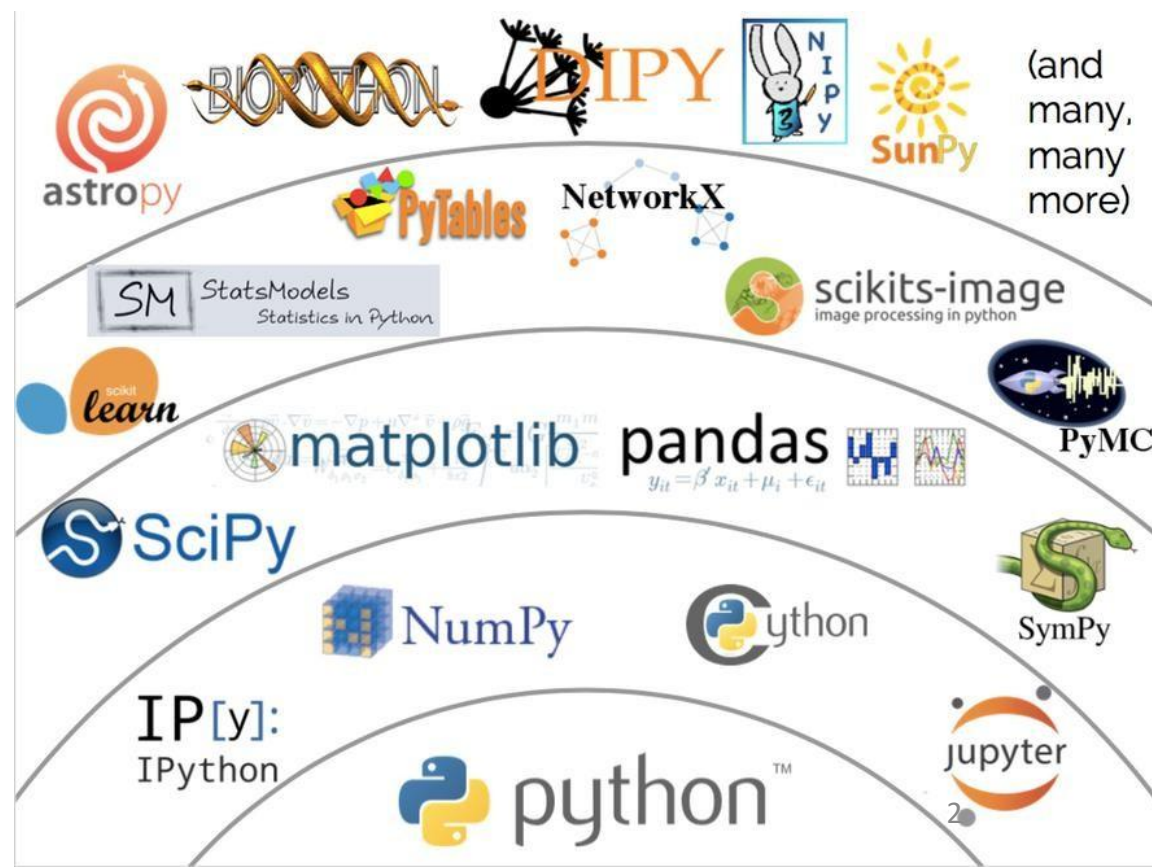
- Introduction
- History and usage
- Comparison
- companies using python
- Variables, Datatypes
- Keywords
- Looping constructs



Artificial Intelligence

Machine Learning

Deep Learning





Python ... *at first glance*

- Python is an **interpreted, object-oriented, high-level** programming language with dynamic semantics.
 - **Interpreted:** Python does not need to be converted into machine code (like C or C++). Instead, it is executed line by line by an interpreter, which makes debugging easier.
 - **Object-Oriented:** Python is based on the concept of "objects," which means it groups related data and functions together. This helps in organizing code better and reusing it easily.
 - **High-Level:** Python is closer to human language than to machine language, making it easier to read and write. You don't have to worry about things like memory management.
 - **Dynamic Semantics:** This means Python figures out the type of data (like a number or a word) on its own when you assign a value to a variable, so you don't have to define it manually.
- Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.
- *Python is portable: it runs on many Unix variants, on the Mac, and on PCs under MS-DOS, Windows, Windows NT, and OS/2.*
- *not to mention* around **80 percent** of the top computer science programs around the world teach Python as the introduction to the program.

The languages of the 4th millennium

Language	Frequency
Python	26 %
JavaScript	23 %
C#	20 %
Java	15 %
C++	13 %
Ruby	3 %
Perl	0.2%
LISP	< 0.1 %
BASIC	< 0.1 %
Cobol	< 0.1 %
Fortran	< 0.1 %

Source: masterbaboon



History of Python

- A scientist once said

“I have used a combination of Perl, Fortran, NCL, Matlab, R and others for routine research, but found out this general-purpose language, Python, can handle almost all in an efficient way from requesting data from remote online sites to statistics, and graphics.”

- The programming language Python was conceived in the late 1980s, and its implementation was started in December 1989 by Guido van Rossum at CWI in at the National Research Institute for Mathematics and Computer Science in the Netherlands.
- Guido van Rossum was also reading the published scripts from “Monty **Python's** Flying Circus”, a BBC comedy series from the 1970s.
- Van Rossum thought he needed a name that was short, unique, and slightly mysterious, so he decided to **call** the language **Python**.



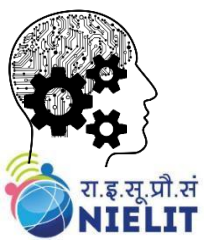
About Python

- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.
- Python source code is now available under the [GNU General Public License](#) (GPL).
- Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language.
- Python is a [scripting language](#) like PHP, Perl, Ruby and so much more.
- It can be used for [web programming](#) (Flask, **Django**, Zope, Google App Engine, and much more).
- Can be used for [desktop applications](#) (Blender 3D, or even for games pygame).
- Python can also be translated into [binary code](#) like java.
- Python can be used for both [artificial intelligence](#) and [statistical analysis](#).
- Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.



More about Python

- **Python** is developed under an OSI-approved open-source **license**, making it freely usable and distributable, even for **commercial use**. **Python's license** is administered by the **Python** Software Foundation.
- **Python** is not an exception - its most popular implementation is called **CPython** and is **written in C**
- The biggest difference between Java and Python is that **Java** is statically typed, and **Python** is dynamically typed. *This makes **Python** very easy to write and not too bad to read, but difficult to analyze.*
- **IronPython** is the **Python** running on .NET
- **Jython** is the **Python** running on the Java Virtual Machine



Python Implementations..

MicroPython – A lean and efficient Python implementation designed to run on microcontrollers and embedded systems.

PyPy – A Python implementation with a Just-In-Time (JIT) compiler, significantly improving performance over CPython for certain workloads.

Stackless Python – A modified version of CPython that removes reliance on the C call stack, enabling lightweight concurrency and micro-threading.

Brython (Browser Python) – A Python implementation that runs in web browsers, allowing developers to write client-side web applications in Python instead of JavaScript.

CircuitPython – A fork of MicroPython, optimized for educational purposes and designed for Adafruit's microcontroller hardware.

RPython (Restricted Python) – A subset of Python used to implement PyPy, focused on writing high-performance interpreters.

Pyston – A performance-oriented Python implementation with Just-In-Time (JIT) compilation and various optimizations.

Transcrypt – A Python-to-JavaScript transpiler that allows developers to write Python code and execute it in the browser as JavaScript.

Nuitka – A Python compiler that converts Python code into C/C++ executables, improving performance and simplifying deployment.

Cython – A superset of Python that enables performance improvements by compiling Python-like code into C, often used for writing Python extensions.

PyExecJS – A wrapper that enables Python to execute JavaScript using various JavaScript runtime engines.

Skulpt – A JavaScript-based Python implementation that runs entirely in the browser, enabling interactive web-based Python coding environments.

Vyper – A Pythonic programming language specifically designed for writing Ethereum smart contracts, featuring strict syntax and security-focused features.


Grumpy – A Python-to-Go transcompiler that allows Python code to run as native Go programs, focusing on performance improvements.

Numba – A Just-In-Time (JIT) compiler for Python that translates numerical Python code into optimized machine code using LLVM, improving performance for scientific computing.

PicoPy – A minimalistic Python implementation tailored for extremely constrained environments, such as embedded systems and IoT devices.

Boa – A Python implementation for WebAssembly, enabling Python code execution efficiently in WebAssembly (WASM) environments.

Programming *with* Python



Google Facebook, Inc. Dropbox Yahoo! NASA IBM Mozilla

Some companies I know that use python are:

- Google (Youtube)
- Facebook (Tornado)
- Dropbox.
- Yahoo.
- NASA.
- IBM.
- Mozilla.
- Quora :D.

More items...

What top tier companies use Python? - Quora
<https://www.quora.com/What-top-tier-companies-use-Python>



Python Identifiers

- A Python identifier is a name used to identify a variable, function, class, module or other object.
- An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).
- Python does not allow punctuation characters such as @, \$, and % within identifiers.
- Python is a **case sensitive programming** language.



Naming conventions for Python identifiers :

- Class names start with an uppercase letter. All other identifiers start with a lowercase letter.
- Starting an identifier with a single leading underscore indicates that the identifier is private.
- Starting an identifier with two leading underscores indicates a strongly private identifier.
- If the identifier also ends with two trailing underscores, the identifier is a language-defined special name.



Building Blocks

- Values and Variables
 - numeric values
 - variables
 - assignment
 - identifiers
 - reserved words
 - comments



Standard Data Types

- Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.
- Python has five standard data types
 - Numbers
 - String
 - List
 - Tuple
 - Dictionary



Python Numbers

- Number data types store numeric values.
- Number objects are created when you assign a value to them. For example –
 - `var1 = 1`
 - `var2 = 10`
- You can also delete the reference to a number object by using the del statement. The syntax of the del statement is –
 - `del var1[,var2[,var3[....,varN]]]`
- You can delete a single object or multiple objects by using the del statement. For example –
 - `del var`
 - `del var_a, var_b`



Checking the type of datatype type()

```
In [1]: type('hello')
```

```
Out[1]: str
```

```
In [2]: type(10)
```

```
Out[2]: int
```

```
In [3]: type(20.5)
```

```
Out[3]: float
```

```
In [4]: a=10  
a
```

```
Out[4]: 10
```




VARIABLE

Python Variable is containers that store values. Python is not “statically typed”.

Rules for Python variables

- A Python variable name must start with a letter or the underscore character.
- A Python variable name cannot start with a number.
- A Python variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _).
- Variable in Python names are case-sensitive (name, Name, and NAME are three different variables).
- The RESERVED WORD in Python cannot be used to name the variable in Python.



Assigning Values to Variables

- Python variables **do not need explicit declaration** to reserve memory space.
- The declaration happens automatically when you assign a value to a variable.
- The equal sign (=) is used to assign values to variables.

- For example –

```
counter = 100      # An integer assignment
```

```
miles  = 1000.0    # A floating point
```

```
name   = "Nielit"  # A string
```

```
print(counter)
```

```
print(miles)
```

```
print(name)
```

- Here, 100, 1000.0 and "Nielit" are the values assigned to *counter*, *miles*, and *name* variables, respectively.

Multiline/single line

In [7]: `a=10;b=20;c=30;
print(a,b,c)`

10 20 30

In [8]: `a\
=
50
print (a)`

50

Reserved words

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield



Data type

Python supports four different numerical types –

- int (signed integers)
- long (long integers, they can also be represented in octal and hexadecimal)
- float (floating point real values)
- complex (complex numbers)

```
x = 10      # int
y = 3.14    # float
z = "Python" # str
a = [1, 2, 3] # list
b = (4, 5, 6) # tuple
c = {7, 8, 9} # set
d = {"name": "Lovnish", "age": 25} # dict
e = True     # bool
```



Basic operator

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
(+= , -= , *= , /= , %=)
- Logical Operators
(and, or, not)
- Bitwise Operators

Arithmetic Operator

+ Addition
- Subtraction
* Multiplication
/ Division
% Modulus
** Exponent
// Floor division

9//2 -> 4

Comparison Operator

==

!=

<>

>

<

>=

<=

Bitwise Operator

& Binary AND

| Binary OR

^ Binary XOR

~ Binary Ones Complement

<< Binary Left Shift

>> Binary Right Shift



Decision making

```
x = int( input('enter marks'))  
if (x>50) : print('pass')  
else : print('fail')
```

Or

```
x = int( input('enter marks'))  
if (x>50) :  
    print('pass')  
else :  
    print('fail')
```

```
if expression1:  
    statement(s)  
    if expression2:  
        statement(s)  
    elif expression3:  
        statement(s)  
    elif expression4:  
        statement(s)  
    else:  
        statement(s)  
  
else:  
    statement(s)
```




loops

while expression :
 statements()

```
i=0  
while (i<5) :  
    print (i, 'Jai Ho')  
    i=i+1
```

```
0 Jai Ho  
1 Jai Ho  
2 Jai Ho  
3 Jai Ho  
4 Jai Ho
```

while expression :
 statements()
else :
 statements()

```
i=0  
while (i<5) :  
    print (i, 'Jai Ho')  
    i=i+1  
else :  
    print (i, ' Its over now')
```

```
0 Jai Ho  
1 Jai Ho  
2 Jai Ho  
3 Jai Ho  
4 Jai Ho  
5  Its over now
```



for loop

for iterating Variable in sequence
statement/s

```
In [3]: states=['J&K', 'HimachalPradesh', 'Punjab', 'Delhi']  
for st in states:  
    print (st)
```

```
J&K  
HimachalPradesh  
Punjab  
Delhi
```

```
In [4]: for st in range(len(states)):  
        print (states[st])
```

```
J&K  
HimachalPradesh  
Punjab  
Delhi
```

```
In [5]: for alpha in 'India':  
        print(alpha)
```

```
I  
n  
d  
i  
a
```

for iterating Variable in sequence
statement/s

else:

statement/s

```
In [7]: for st in range(len(states)):  
        print (states[st])  
else :  
        print('-----Its over ----- ')
```

```
J&K  
HimachalPradesh  
Punjab  
Delhi  
-----Its over -----
```

Loop Control Statements

Break : Terminates loop statement

```
for alpha in 'Greatness':  
    if alpha == 'n':  
        break  
    print ('letter ', alpha)
```

```
letter G  
letter r  
letter e  
letter a  
letter t
```

continue : returns the control to the beginning of the while/for loop

```
for alpha in 'Greatness':  
    if alpha == 'n':  
        continue  
    print ('letter ', alpha)
```

```
letter G  
letter r  
letter e  
letter a  
letter t  
letter e  
letter s  
letter s
```

pass : is used when a statement is required syntactically but you do not want any command or code to execute

```
for alpha in 'Greatness':  
    if alpha == 'n':  
        pass  
    print ('Pass block')  
    print ('letter ', alpha)
```

```
letter G  
letter r  
letter e  
letter a  
letter t  
Pass block  
letter n  
letter e  
letter s  
letter s
```



PYTHON INSTALLATION (OFFLINE MODE)

Step 1: Select Version to Install Python

Visit the official page for Python <https://www.python.org/downloads/> on the Windows operating system. Locate a reliable version of Python 3, preferably version 3.10.11, which was used in testing this tutorial. Choose the correct link for your device from the options provided: either Windows installer (64-bit) or Windows installer (32-bit) and proceed .

STEP 2:Download the executable file.

Once you have downloaded the installer, open the .exe file, such as python-3.10.11-amd64.exe, by double-clicking it to launch the Python installer. Choose the option to Install the launcher for all users by checking the corresponding checkbox, so that all users of the computer can access the Python launcher application. Enable users to run Python from the command line by checking the Add python.exe to PATH checkbox and at the end Click on Disable path length limit.

Step 3: Running the Executable Installer

After completing the setup. Python will be installed on your Windows system. You will see a successful message.

Step 4: Verify the Python Installation in Windows Close the window after successful installation of Python. You can check if the installation of Python was successful by using either the command line or the Integrated Development Environment (IDLE), which you may have installed. To access the command line, click on the Start menu and type “cmd” in the search bar. Then click on Command Prompt. `python --version`



How to install Python on Windows? (OFFLINE MODE)

Python >>> Downloads >>> Windows

Python Releases for Windows

[Latest Python 3 Release - Python 3.11.3](#)

Stable Releases

- Python 3.10.11 - April 5, 2023
Note that Python 3.10.11 cannot be used on Windows 7 or earlier.
 - Download [Windows embeddable package \(32-bit\)](#)
 - Download [Windows embeddable package \(64-bit\)](#)
 - Download [Windows help file](#)
 - Download [Windows installer \(32-bit\)](#)

Python 3.10.11 (64-bit) Setup

Install Python 3.10.11 (64-bit)

Select **Install Now** to install Python with default settings, or choose **Customize** to enable or disable features.

Install Now
C:\Users\ashub\AppData\Local\Programs\Python\Python310

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ **Customize installation**
Choose location and features

☒ Use admin privileges when installing py.exe
☒ Add python.exe to PATH

Cancel

Python 3.10.11 (64-bit) Setup

Setup Progress

Installing:

Python 3.10.11 Standard Library (64-bit)

python for windows

Python 3.10.11 (64-bit) Setup

Setup was successful

New to Python? Start with the [online tutorial](#) and [documentation](#). At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See [what's new](#) in this release, or find more info about [using Python on Windows](#).

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

python for windows

Close



PYTHON IDE ONLINE MODE STEP 1 & STEP2

About 1,100,000 results



Python

Programming language

Updates

Download

Courses

Frameworks



Programiz

<https://www.programiz.com/python-programming/online-compiler>

[Online Python Compiler \(Interpreter\) - Programiz](https://www.programiz.com/python-programming/online-compiler)

Write and run your Python code using our online compiler. Enjoy additional features like code sharing, dark mode, and support for multiple programming languages.

Online R Compiler

Enjoy additional features like code sharing, dark mode, and support for multiple programming languages. {} Online Python Compiler Online R Compiler Online SQL Editor Online HTML/CSS ...

[See results only from programiz.com](#)

People also search for

python mode of list

types of mode in python

https://www.programiz.com/python-programming/online-compiler/

BLACK FRIDAY SALE Get Programiz PRO for LIFE at 60% off! [Claim My Discount](#)

Programiz
Python Online Compiler

Premium Coding Courses by Programiz

main.py

Run Share Dark Mode

```
1 print("this is my online ide for Python")
```

Output

this is my online ide for Python

=== Code Execution Successful ===




Google Collab STEP 1 & STEP2

Microsoft Bing

google colab

SEARCH COPILOT IMAGES VIDEOS

About 984,000 results

 Google Colab
<https://colab.research.google.com>


Google Colab

What is Colab? Colab, or "Colaboratory", allows you to run code in the cloud with. Zero configuration required. Access to GPUs

Notebook


Colab notebooks allow you to combine executable code and rich text in a single ...


<https://colab.research.google.com/drive/1H66pA1eXv1b3TDC...>

 Untitled0.ipynb ☆

File Edit View Insert Runtime Tools Help

+ Code + Text

 `print("hello worls***")`

 hello worls***



PRACTICAL CALCULATION OF SIMPLE INTEREST

Input values

```
principal = float(input("Enter principal amount: "))  
rate = float(input("Enter rate of interest (in %): "))  
time = float(input("Enter the time (in years): "))
```

Calculation of Simple Interest

```
simple_interest = (principal * rate * time) / 100
```

Display the result

```
print(f"The Simple Interest is: {simple_interest}")
```

```
# Input values  
principal = float(input("Enter principal amount: "))  
rate = float(input("Enter rate of interest (in %): "))  
time = float(input("Enter the time (in years): "))  
|  
# Calculation of Simple Interest  
simple_interest = (principal * rate * time) / 100  
  
# Display the result  
print(f"The Simple Interest is: {simple_interest}")
```

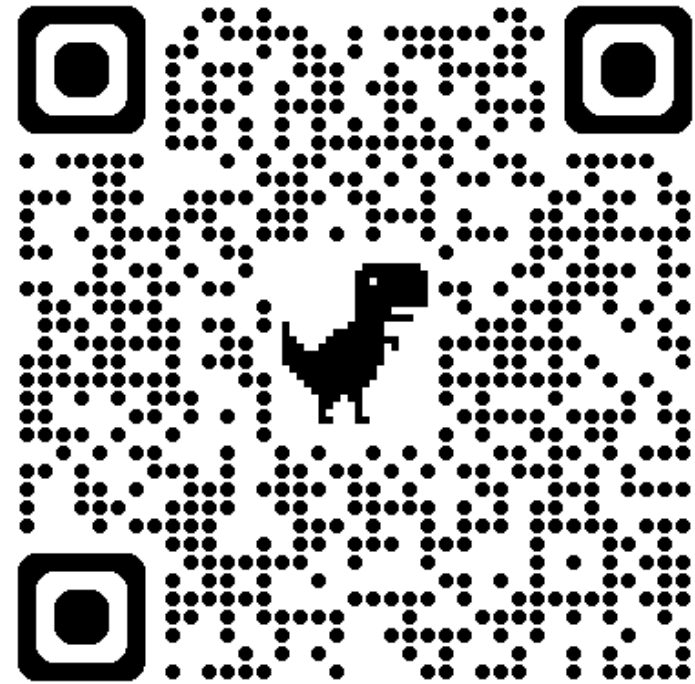
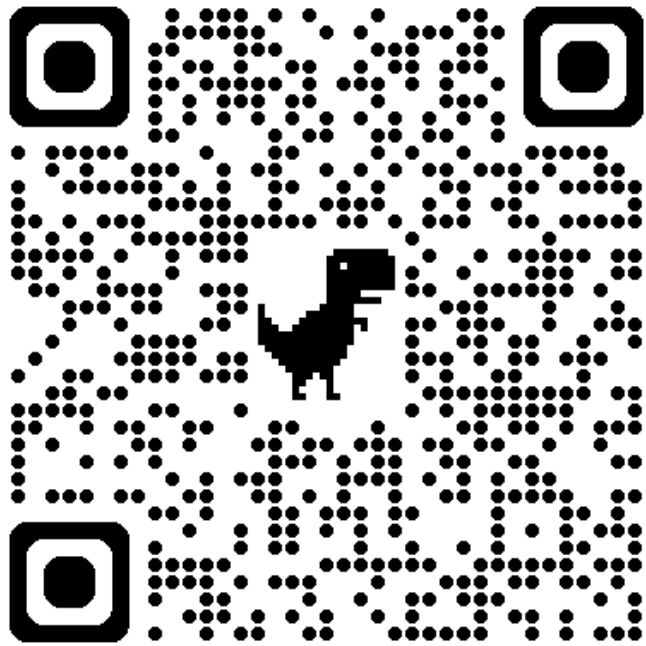
Output:-

```
Enter principal amount: 5  
Enter rate of interest (in %): 2  
Enter the time (in years): 5  
The Simple Interest is: 0.5
```

$$\text{Simple Interest (SI)} = \frac{P \times R \times T}{100}$$



Python Getting Started



<https://github.com/lovnishverma/Python-Getting-Started>

https://www.w3schools.com/python/python_getstarted.asp