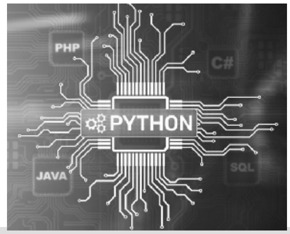


1 परिचय
(Introduction)

- Brief History
- Python Version
- Installing Python
- Environment Variable
- Executing Python from Command Line
- IDLE
- Editing Python Files
- Python Documentation
- Getting Help
- Dynamic Types
- Reserved Word
- Naming Conventions

■ Python

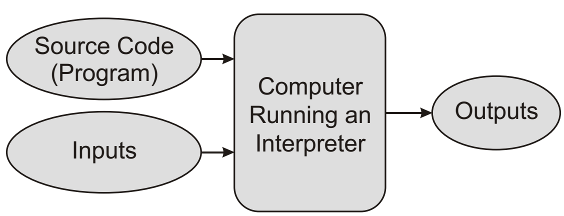
Python एक high level तथा interpreted programming language है; जोकि सन् 1989 में Guido Van Rossum द्वारा develop की गई तथा प्रथम मर सन् 1991 में release की गई थी। परिणामस्वरूप यह ऐसी programming language हुई जो software development तथा web development के लिए बहुत उपयोगी साबित हुई।

Python में कुल दो Version हैं—Python 2 एवं Python 3 तथा दोनों ही आपस में बिल्कुल अलग ह□।

सामान्यतया Python file का extension .py होता है तथा Command के माध्यम से भी हम कोई प्रोग्राम आसानी से run कर सकते हैं।

किसी भी Python Program को निम्नलिखित तीन प्रकार से run कर सकते हैं—

1. Interactive Interprefer के द्वारा
2. Script from command line के द्वारा
3. Integrated Development Environment (IDLE) के द्वारा



चित्र : Interpreting a High Level Language

Features of Python

1. Python एक high level language है; जो free तथा open source language है।

2. यह एक interpreted language है अर्थात् Python program हमेशा interpreter द्वारा execute किये जाते हैं।
3. Python program को आसानी से समझा जा सकता है। तथा इसका Syntax तथा Structure simple एवं clearly defined होता है।
4. Python case sensitive programming language है अर्थात् PROGRAM तथा program दोनों अलग-अलग हैं।
5. Python, portable तथा plateform independent है। अर्थात् यह विभिन्न आपरेटिंग सिस्टम एवं हार्डवेयर प्लेटफार्म पर आसानी से run कर सकती है।
6. Python में बहुत सारे Predefined library functions उपलब्ध हैं।
7. Python web development में बहुत उपयोगी है।
8. Python में कोड प्लान तथा Nested प्लान लिखने के लिए indentation का प्रयोग किया जाता है।
9. Python, कम्प्यूटर इंजीनियरिंग के नये तकनीकी जैसे Machine learning, 10T NLP आदि के implementation हेतु बहुत उपयोगी है।
10. Python हमें लगभग सभी Commercial database से interface प्रदान करता है।

Limitations/Disadvantage of Python

यह language विशेष रूप से Enterprise solution आदि के लिए बहुत उपयोगी नहीं है। इसके कुछ limitations निम्नवत् हैं—

1. **Difficulty in using other language :** Python को यूज करने वाले programmer जो इसकी library तथा features से familiar हो जाते हैं, उन्हें दूसरी programming languages पर कार्य करने में समस्या उत्पन्न होती है।
2. **Weak in Mobile Computing :** Python language किसी भी mobile computing application के लिए बहुत उपयोगी नहीं है।
3. **Gets Slow in Speed :** Python में प्रोग्राम compiler के बजाय interpreter के मदद से exicute होता है, जिससे program execution slow हो जाता है तथा कभी-कभी यह web application में fast भी हो जाता है।
4. **Run Time Error :** Python एक dynamically typed प्रोग्रामिंग लैंग्वेज है, जिससे यह run time error show करता है।
5. **Data Base Access :** अन्य प्रोग्रामिंग लैंग्वेज Java, JDBL, ODBC की तुलना में इसमें limited data base access होता है।
6. **Memory Consumption :** Python, memory intensive task के लिए बहुत अच्छा विकल्प नहीं होता है। Python में ज्यादा memory का consumption होता है।

Applications of Python

Python एक general purpose programming language है, जिसके बहुत सारे अनुप्रयोग हैं। कुछ महत्वपूर्ण Applications निम्नवत् हैं—

1. **Web Development :** हम Web application develop करने के लिए भी Python का प्रयोग कर रहे हैं। इसमें बहुत सारी predefined library उपलब्ध हैं। जिसकी मदद से HTML, XML, JSON, E-mail, Internet Protocol को आसानी से handle किया जा सकता है। Python में Python Wiki Engines, POCOO, Python Blog Software कुछ महत्त्वपूर्ण developments हैं।

2. **Data Science :** आज के दौर में जहाँ बहुत सारा data को process करना होता है, Python में उपलब्ध महत्वपूर्ण library के कारण Data Science में यह बहुत ही उपयोग सिद्ध हो रहा है। जिसमें Numpy, Pands एवं Matplotlib कुछ महत्वपूर्ण एवं उपयोगी library हैं। जिससे हम data cleaning, visualization आदि को आसानी से हैण्डल कर पाते है।
3. **Machine Learning :** Machine learning भी Python की एक बहुत ही महत्वपूर्ण application है। Python में उपलब्ध library जैसे Scikit-lean, Tensorflow एवं NLTK आदि का प्रयोग machine learning हेतु बहुतायत रूप से किया जाता है। जिसमें मुख्य रूप से Automation, AI, Medical diagnosis, Sales Analysis आदि प्रमुख अनुप्रयोग हैं।
4. **Image Processing :** Image processing के application, face recognition, activity tracking आदि में open cv library का प्रयोग किया जाता है। Python, Image processing के लिए बहुत popular है। Vpython, Gogh, imgseek प्रमुख application हैं।
5. **Game Development :** Game development के लिए भी Phyton का प्रयोग हो रहा है। Pygame library मुख्य रूप से इसके लिए प्रयोग की जाती है। Android mobile में games development के लिए Python का प्रयोग हो रहा है।
6. **Embedded System and IOT :** यह भी Python की मुख्य application है। इसके प्रयोग के द्वारा product automation, robotics, IOT आदि को develop करने में Python बहुत ही उपयोगी है। Micropython, Python का एक light version है, जोकि Microcontroller programing के लिए विशेष रूप से बनाया गया है।
7. **Android Apps :** Android Apps का development विशेष रूप से Android SDK के माध्यम से किया जाता है। लेकिन python के माध्यम से भी हम Android App को develop कर सकते हैं, जिसके लिए Pythoris Kivy library का प्रयोग किया जाता है।
8. **Scientific and Numerical Application :** Python का प्रयोग Scientific एवं Numerical Computing के लिए किया जाता है। Scipy, Pondas, Ipython, Numeric Python प्रमुख library हैं, जो Engineering एवं Science के application development में प्रयोग किये जाते हैं।
9. **Business Applications :** Python के प्रयोग के माध्यम से हम ERP, e-commerce systems आदि को develop करते हैं।
10. **3D CAB Application :** 3D CAD Application को develop करने के लिए Fandango का प्रयोग किया जाता है।

■ Process of Programming

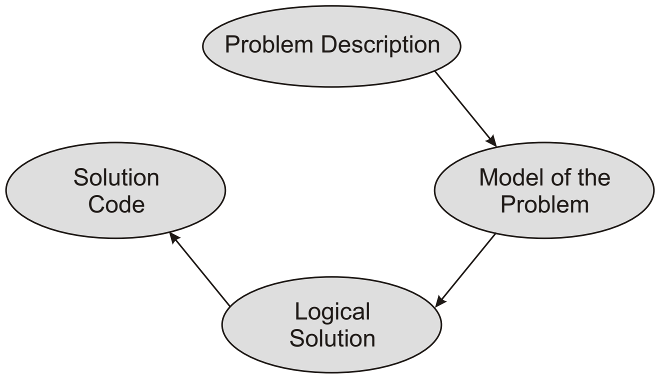
Step 1 : Problem को define करना तथा model करना : सर्वप्रथम किसी समस्या को define करना तथा उसकी Modeling करना है।

Step 2 : Problem का logical solution प्राप्त करना : Logical solution एक clear, finite तथा step-by-step procedure है किसी समस्या को हल करने का।

इसे ही हम Algorithm कहते हैं। इसे पिक्चर फार्म में प्रस्तुत करने के लिए ही flow chart का प्रयोग करते हैं।

उदाहरण के तौर पर, यदि रियल लाइफ में हमें चाप बनाना है और उसका प्रोसेस हम google पर सर्च करते हैं तो हमें उसका finite solution मिल जायेगा। दूसरे चरण में हमारे step छोटे तथा clear होने चाहिएँ।

Step 3 : प्रोग्राम लिखना : Programing : तीसरे step में हम algorithm को एक working code में परिवर्तित करते हैं, जिससे computer आसानी से समझ सके और उसके लिए हमें जरूरत होती है Programming language की, जिसमें हमें Program लिखना होता है।



Programing Cycle of Python

1. Write your program or edit your program
2. Run your program on an input
3. If output is not correct, return to coding step (step 1)
4. Repeat step 2, 3 for other inputs if any.

Classification of Programming Language

Programming language, किसी भी computer को निर्देश (instruction) देने का एक माध्यम है। जो कुछ syntactic तथा Semantic नियमों पर आधारित है। सूचना प्रौद्योगिकी की दुनिया में बहुत-सी Computer Programming Languages प्रयोग में लायी जाती हैं। उन्हें उनकी कार्य प्रणाली के आधार पर विभाजित किया जाता है। इसके मुख्य प्रकार निम्नवत् हैं—

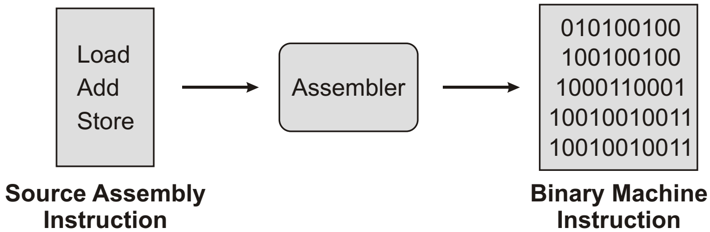
1. Machine language
2. Assembly language
3. High level language
4. System language
5. Scripting language
6. Domain Specific language
7. Glue language

Machine Language : इसे lowest level तथा First Generation Programming language के नाम से जाना जाता है। इस तरह की Programming language में केवल दो शर्तें होती हैं—True एवं False। इसलिए इसे Binary Programming language के नाम से भी जाना जाता है। Computer में सभी instructions को 0 एवं 1 रूप में ही फीड किया जाता है। यह Machine level language समझने में कठिनाई होती है।

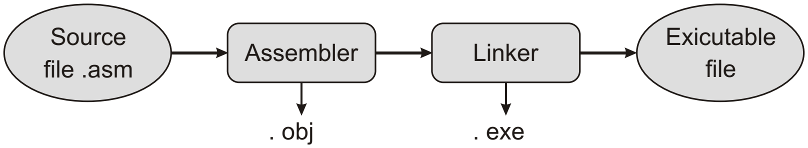
Assembly Language : इसे middle level language तथा Second Generation language के नाम से जाना जाता है। इसके निर्देश (instructions) भी Machine language की तरह ही होते हैं तथा instruction तथा variable के specific नाम होते हैं। जिसे हम command कहते हैं। इसमें निर्देशों की व्याख्या तथा समझने के लिए कुछ symbols का प्रयोग किया जाता है।

इसमें हर एक machine instruction के लिए एक statement लिखा जाता है। इसे मैनेज करने के लिए register allocation, call, stack, timer, jump, loop आदि का प्रयोग किया जाता है।

Computer की assembly language को समझने के लिए एक Assember को आवश्यकता होती है। जो assembly instruction को machine instruction में परिवर्तित करता है। Assembler एक सॉफ्टवेयर होता है।

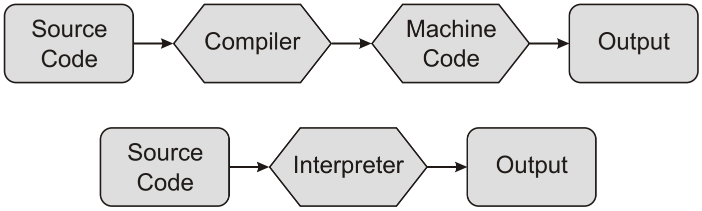


चित्र : Assembler



चित्र : Assembler and Linker

High Level Language : सभी प्रकार की High Level Language (HLL), procedure oriented language तथा machine independent होती है। यह upper language तथा third generation language के नाम से भी जानी जाती है। यह हमें program को साधारण English language लिखने, Mathematical एवं अन्य सिग्नल का प्रयोग करने की सुविधा प्रदान करती है। जिससे प्रोग्राम, आसान redable तथा समझने योग्य हो जाता है। ऐसे प्रोग्राम को Computer को समझने के लिए उसे Machine instruction में translate करना पड़ता है। जिसके लिए हमें एक सॉफ्टवेयर की आवश्यकता होती है। जिसे compiler या interpreter कहते हैं। इसे compiler एवं translator का मुख्य कार्य source कोड को machine understandable code में परिवर्तित करना होता है।



चित्र : Compiler and Interpreter

उदाहरण : FORTRAN, Pascal, C, C++, Java, ADA, COBOL, LISP, Prolog etc.

System Language : System Programming के लिए हम system programming language का प्रयोग करते हैं तथा उसकी मदद से System s/w बनाया जाता है।

उदाहरण : PL/I, D, Rust, Swift

Scripling Language : यह ऐसी progranming language होती है जो दूसरी programming language से communicate करने के लिए design की जाती है। इसे extremly high level तथा powerfull language कहा जाता है। जैसे Java Script, VB Script, PHP, Port Python, Rubly, ASP and TCC.

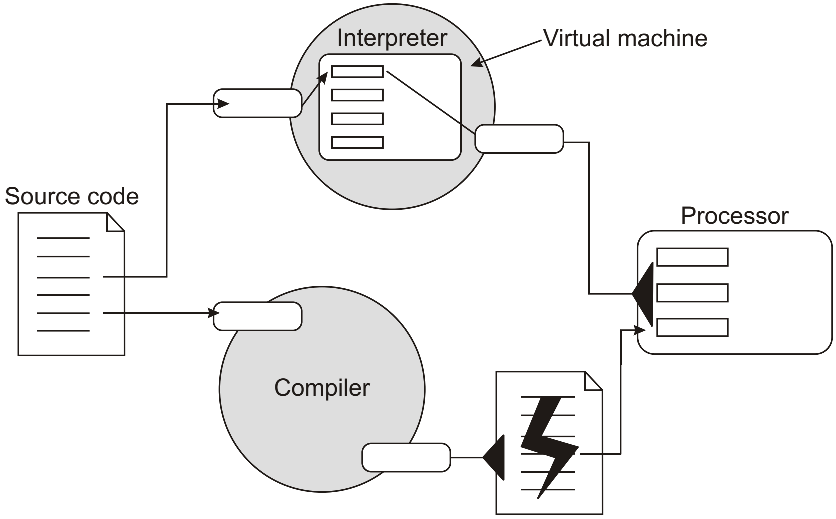
Domain Specific Language : यह programming languauge किसी विशेष कार्य को पूरा करने के काम में आती है। जैसे CSS, Ant, SQL आदि।

Glue Language : Glue language ऐसी programming language है जो दूसरे सॉफ्टवेयर component के प्रोग्राम को manage करने का कार्य करती है।

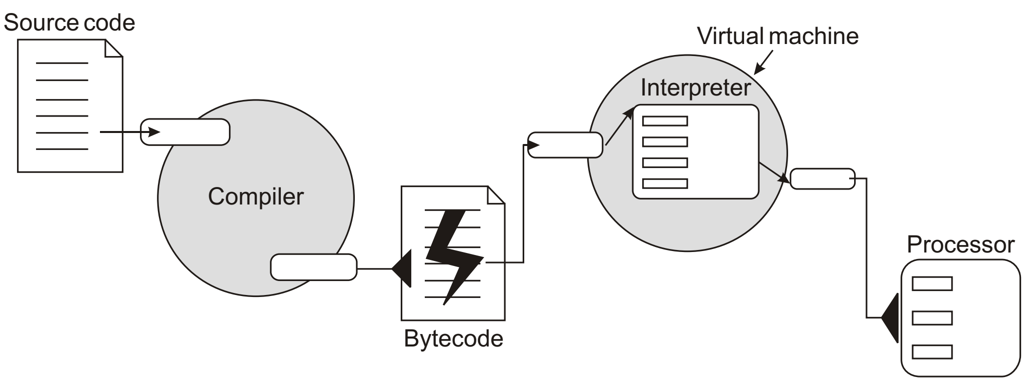
यह ऐसे s/w या program को integrate करने का कार्य करती है जो भिन्न-भिन्न प्रोग्रामिंग लैंग्वेज तथा प्लेटफार्म पर विकसित किये गये होते हैं। यह interconnecting को support करती है।

Glue programming language, प्रोटोटाइप environment में बहुत उपयोगी साबित होती है, जहाँ विभिन्न प्रकार के software से एक framework बनाया गया होता है।

उदाहरण : VBCript, Ruby, Python, Perl and PHP



चित्र : Compiler *Vs* Interpreter



चित्र : Bite Compile Code

1.1 History of Python

Python, सन् 1980 के लगभग Guido Van Rossum द्वारा develop की गयी थी जिसे Netherland की "National Research Institute for Mathematics and Computer Science" प्रयोगशाला में विकसित किया गया था।

Python को दूसरी प्रोग्रामिंग लैंग्वेज जैसे ABS Modula-3, C, C++, Afgol, Unix Shell आदि से derive किया गया है।

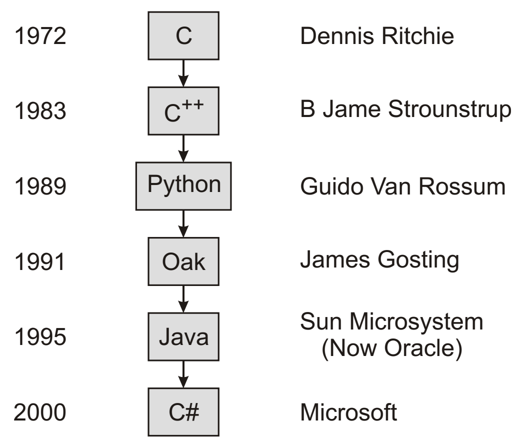
Python, copyrighted है। Python का source code GNU General Public Licence (GPL) के अन्तर्गत पब्लिक डोमेन में उपलब्ध है।

Python अभी भी एक कोर टीम द्वारा maintain की जाती है तथा Guido Van Rossum अभी भी उसकी progress को डायरेक्ट करते हैं तथा यह अधिकार उनके पास सुरक्षित है।

नवम्बर 1994 में Python 1.0 को जारी (release) किया गया था तथा सन् 2000 में Python 2.0 को जारी (release) किया गया था। Python 2.7.11 अभी भी Python 2.0 का सबसे नवीनतम version है।

सन् 2000 में Python 3.0 को जारी किया गया था। इसका नवीनतम version Python 3.5.1 अभी उपलब्ध है। Python 3, Python 2 दो अलग-अलग Python के version हैं।

Evaluation of Python



Why Python

Python एक General purpose programming language है। जो मुख्यत: scripting के लिए प्रयोग की जाती है। इसे object oriented scripting language भी कहा जाता है। यह oop के concent को support करती है। Python codefile को program के बजाय Script कहा जाता है।

Python के बहुत से लाभदायक features होने के कारण प्रोग्रामर इसे प्रयोग में लाना चाहते हैं। इसके कुछ महत्वपूर्ण गुण निम्नवत हैं, जो प्रोग्राम को आकर्षित करते हैं—

1. Simple and Easy to learn
2. Interpreted and Interactive
3. Object Oriented
4. Portable
5. Scalable
6. Extendable
7. Dynamic
8. GUI Programming and Data Bases
9. Board Standard Library

1.2 Python Versions

समय-समय पर Python software को update किया जाता है। तथा कुछ सुधारों तथा कुछ नये features के साथ उसका नया version जारी किया जाता है।

Python का पहला version Python 1.0 V, Jan 1994 में जारी किया गया था। इसके बाद Oct. 2000 में Python 2.0 V तथा Dec. 2008 में Python 3.0 V को जारी किया गया था। वर्तमान में उपलब्ध Python का Python 3.9.0 एवं Python 2.7.18 से नवीन version है।

नोट—Python 3 के वर्जन में (Python 2 से) Backward compatability नहीं होती है अर्थात् Python 2 का Program जरूरी नहीं है। Python 3 में हमेशा run करें।

	Python	Documentation Released
1.	Python 3.9.0	05 Oct 2020
2.	Python 3.8.6	23 Sept 2020
3.	Python 3.8.5	20 July 2020
4.	Python 3.8.4	13 July 2020
5.	Python 3.8.3	13 May 2020
6.	Python 3.8.2	24 Feb 2020
7.	Python 3.8.1	18 Dec 2019
8.	Python 3.8.0	14 Oct 2019
9.	Python 3.7.0	24 June 2018
10.	Python 3.6.0	23 Dec 2016
11.	Python 3.0	03 Dec 2008
1.	Python 2.7.18	20 April 2020
2.	Python 2.7.17	19 Oct 2019
3.	Python 2.7.16	02 March 2019
4.	Python 2.7.15	30 April 2018
5.	Python 2.7.14	17 Dec 2016
6.	Python 2.7	04 July 2010
7.	Python 2.6	01 Oct 2008
8.	Python 2.5	19 Sept 2006
9.	Python 2.4	30 Nov 2004
10.	Python 2.0	16 Oct 2000

Flavors of Python

1. C Python : यह Python का Standard Flavor है। इसका प्रयोग C language Applications के लिए होता है।

2. Jython or JPython : यह मुख्यत: Java Applications के लिए है। यह Jum पर run करता है।

3. Iron Python : यह C# तथा .Net Platform पर प्रयोग में लाया जाता है।

4. P_4P_4 : यह JIT Compiler पर कार्य करता है तथा PVM में उपलब्ध होता है।

5. Ruby Python : यह Ruby Platform पर कार्य के लिए होता है।

6. Anaconda Python : यह मुख्य रूप से large volume के data processing, machine learing तथा mathematical Calculation के लिए प्रयोग में लाया जाता है।

Where We Can Use Python

वर्तमान में Python को मुख्य तौर निम्न कार्य हेतु प्रयोग कर रहे हैं—

1. For Developing Desktop Applications
2. For Developing Web Application
3. For Developing Database Applications
4. For Network Programming
5. For Developing Games
6. For Data Analysis Application
7. For Machine Learning Application
8. For Developing Artificial Intelligence Applications
9. For Internal of Things (IOT) Application
10. For Image Processing and its Application

1.3 Installing Python

Python एक Cross platform programing language है। अर्थात् यह किसी भी Platform जैसे Windows, Macox, Linux आदि पर आसानी से run करती है। यह free तथा open source है।

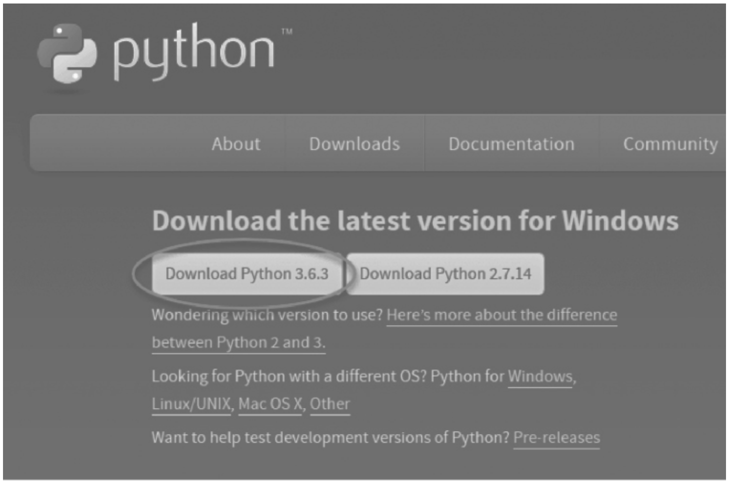
How to Install Python on Windows [Pycharm IDE]

PyCharm is a cross-platform editor developed by JetBrains. Pycharm provides all the tools you need for productive Python developmen. Below are the detailed steps for installing Python and PyCharm.

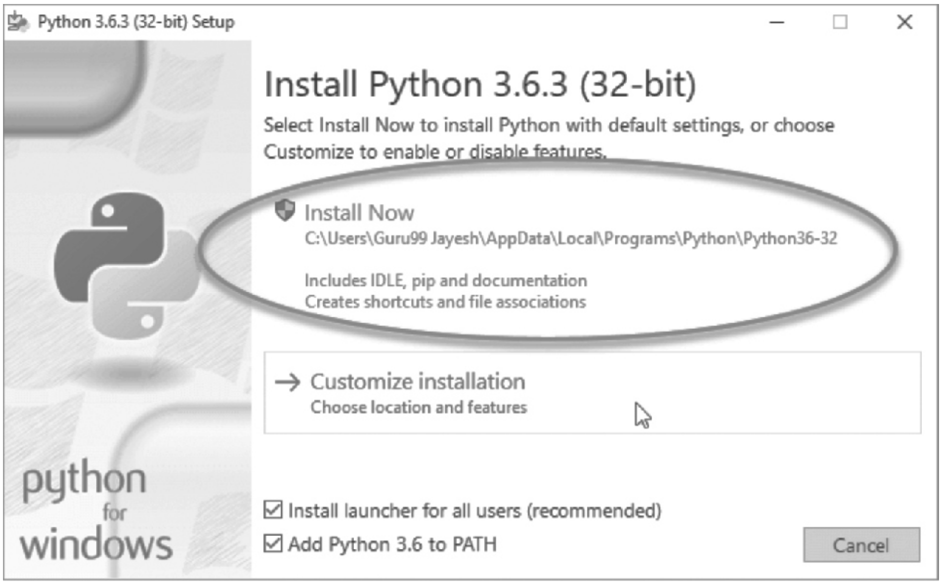
■ Installing Python

Step 1 : To download and install Python visit the official website of Python

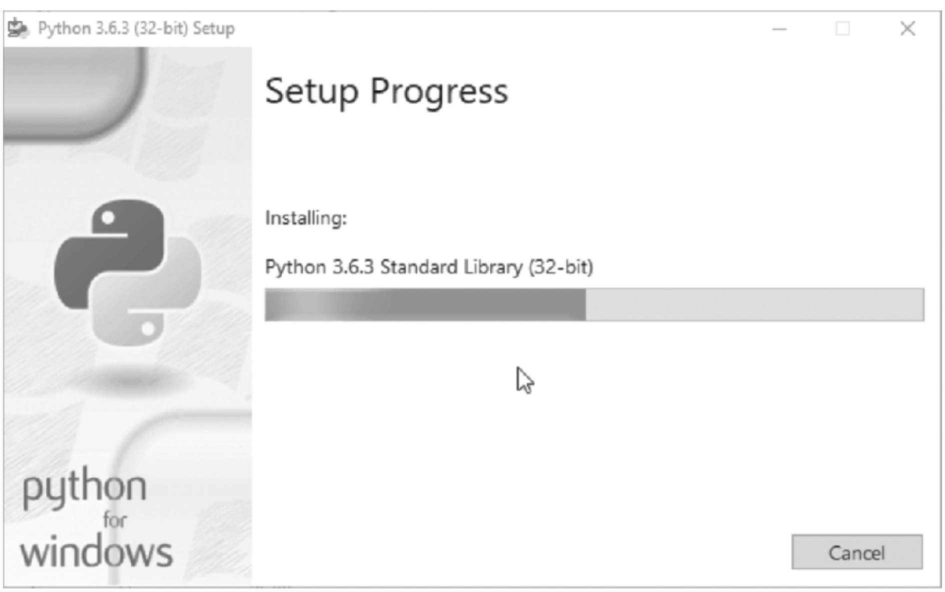
http://www.python.org/downloads/and choose your version. We have chosen Python version 3.6.3



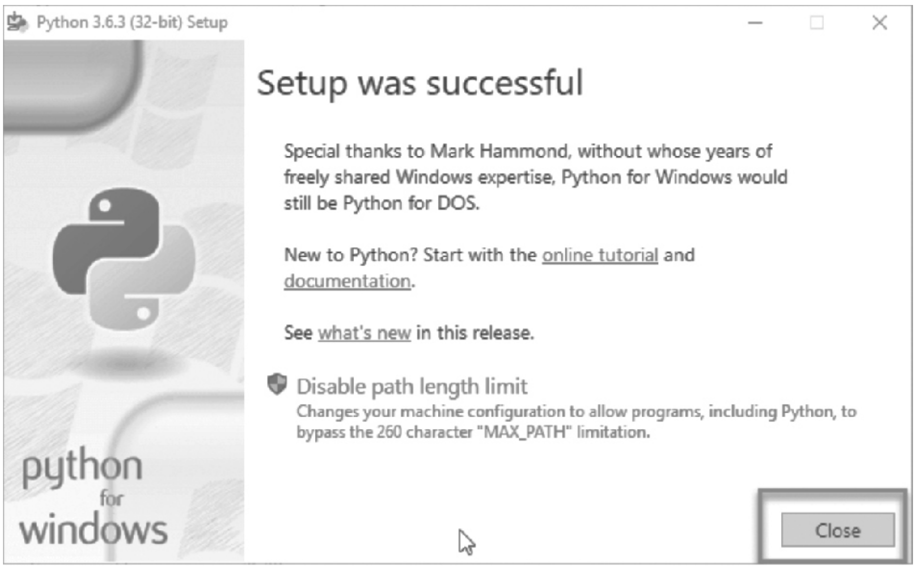
Step 2 : Once the download is complete, run the axe for install Python. Now click on Install Now.



Step 3 : You can see Python installing at this point.



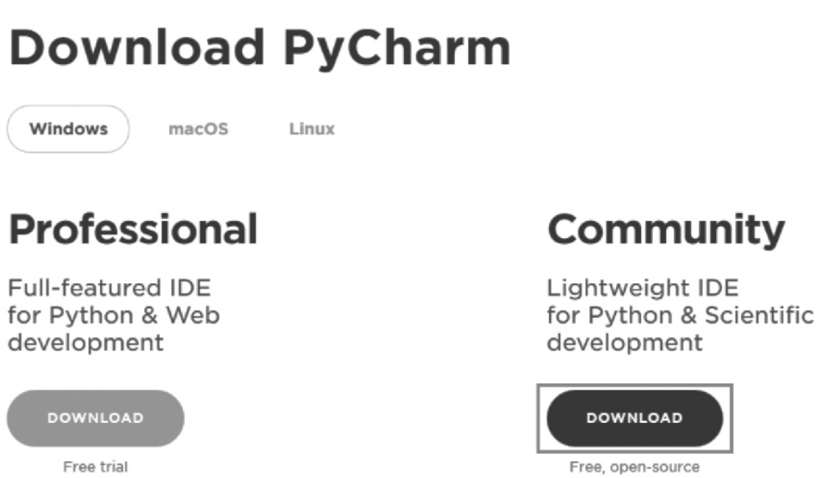
Step 4 : When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".



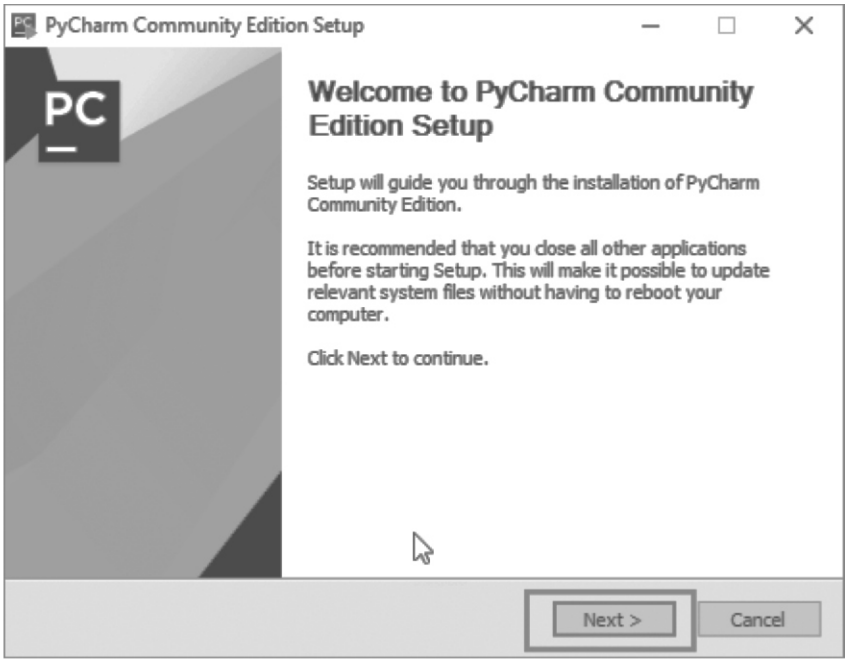
Installing Pycharm

Step 1 : To download PyCharm visit the website

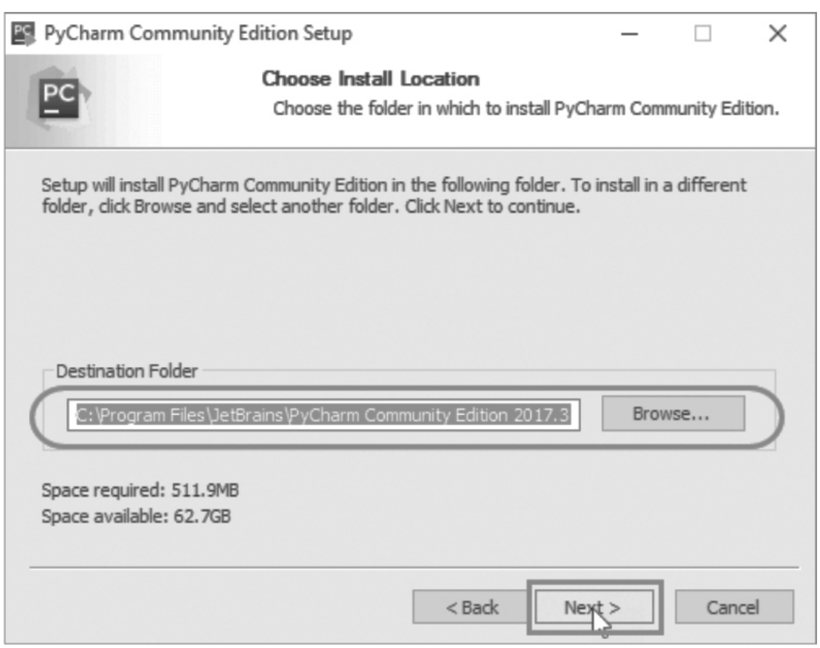
http://www.jetbrains.com/pycharm/download/and click the "DOWNLOAD" link under the Community Section.



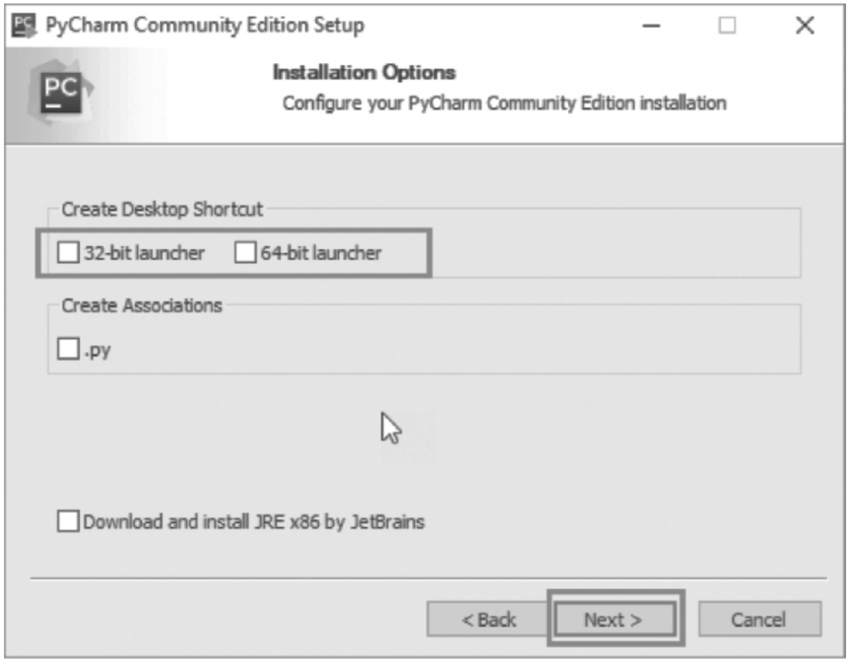
Step 2 : Once the download is complete, run the exe for install PyCharm. The stepup wizard should have started. Click "Next".



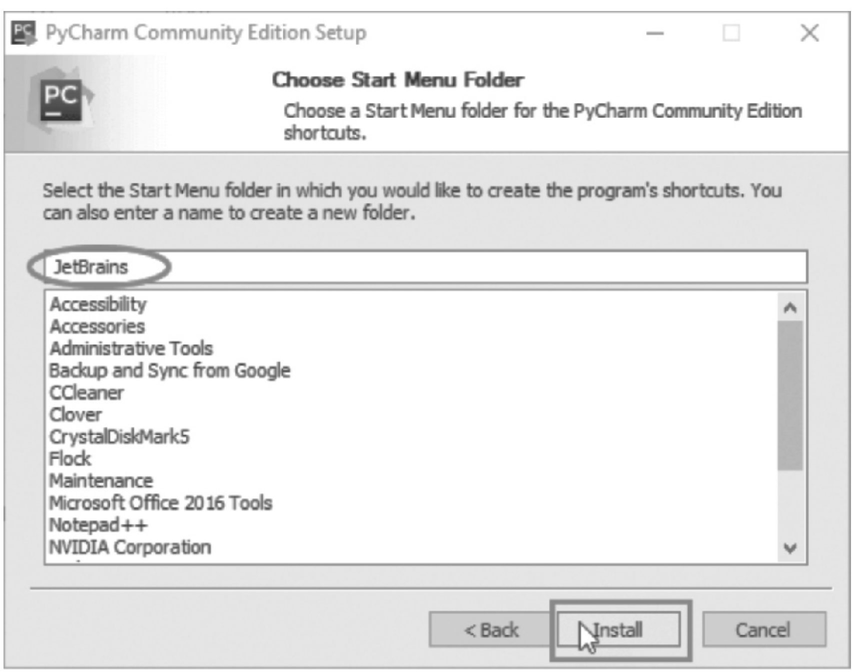
Step 3 : On the next screen, change the installation path if required.



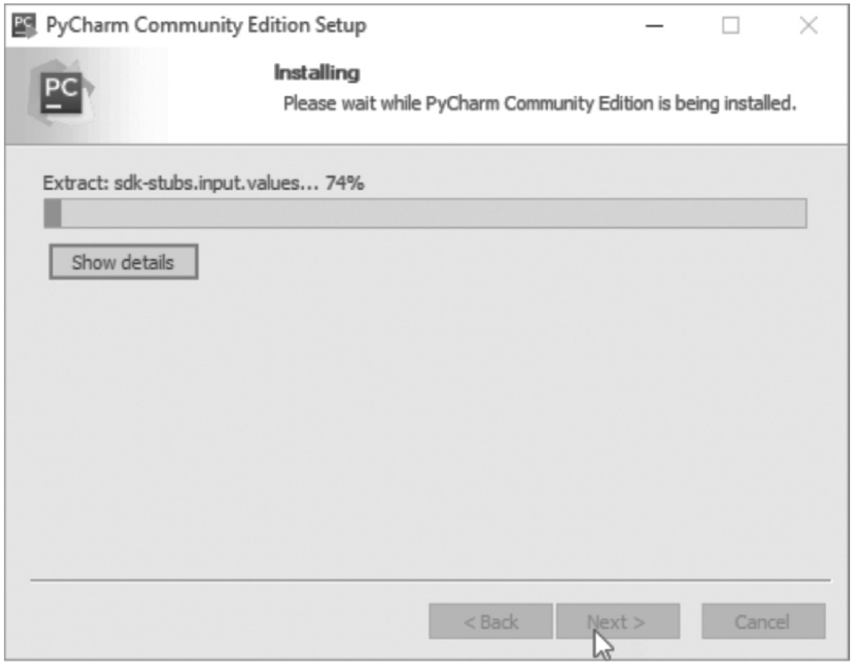
Step 4 : On the next screen, you can create a desktop shortcut if you want and click on "Next".



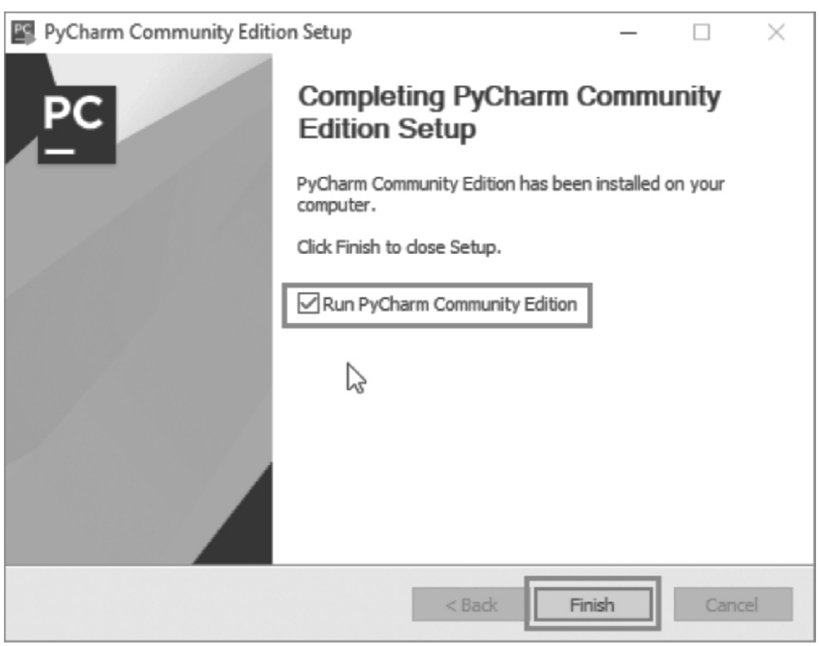
Step 5 : Choose the start menu folder. Keep selected Jet Brains and click on "Install".



Step 6 : Wait for the installation to finish.



Step 7 : Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the "Run PyCharm Community Edition" box first and click "Finish".



Step 8 : After you click on "Finish", the Following screen will appear.

1.4 Environment Variable

	Variable	Description
1.	PYTHONPATH	यह PATH के समान ही होता है। यह बताता है कि Interpereter को प्रोग्राम में import की गयी module file को कहाँ locate करना है। PYTHONPATH कभी-कभी Python installer द्वारा भी प्रस्तुत किया जाता है।
2.	PYTHONSTARTUR	यह उस file के path को रखता है जो Python Source कोड को initilize करती है। जब भी interpreter start होता है, तब यह execute होता है। यह PYTHONPATH को modify या utilities को लोड करने का command अपने पास रखता है।
3.	PYTHONCASEOK	यह imported statement में first case insensitive match के लिए Python को निर्देशित करता है। इसे activate करने के लिए इसे किसी भी value के साथ set किया जाता है।
4.	PYTHONHOME	यह Search Path का विकल्प है, जो PYTHONSTARTUR या PYTHONPATH में embedded रहता है।

File Extensions in Python

.py	Extension of normal python source file
.pyc	Compiled Bytecode file
.pyd	Window DLL file
.pyo	File created with optimization
.pyw	Python script for windows
.pyz	Python script archive

1.5 Executing Python from the Command Line

#One more thing to open python "command prompt".

Example in command prompt

```
>>>print ("hello"....)
    hello
>>>exit
use exit ( ) or ctrl-2 plus return to exit
>>>quit ( )
```

Example to open notepad :

```
C:\users/Bhanupratap>notepadIIT
C:\users\Bhanupratap>pythoniitk.py.STP_2020
C:\users\Bhanupratap>
```

Go to in your file and change name

```
C:\windows\system32\cmd.exe
```

Example of swapping program :

	a	b
a = 10		
a = 20	20	10

```
a = a + b
b = a – b
a = a – b
print
```

One more method to write swapping program

```
a = int (a*b)
b = int (a/b)
a = int (a/b)
print (" the new value of a is = "a")
print (" the new value of b is = "b")
```

1.6 IDLE

Python Program को run करने का एक माध्यम IDLE (Integrated Development Environment) है। Development Environment ऐसा tools का सेट है जो Program writing को आसान बनाता है। यह Program के लिए word processor की तरह कार्य करता है। यह आपके Program को Write करने, save करने, edit करने में मददगार होता है। IDLE में कार्य करने के दो प्रमुख मोड (Mode) दिये गये हैं।

1. Interactive Mode
2. Script Mode

1. Interactive Mode : किसी भी Python Program को लिखने का बहुत ही अच्छा तथा तेज (quick) तरीका होता है इसमें आप Python को बताते हैं कि आप क्या करना चाहते हैं और यह उसके अनुसार respond करता है।

Interactive Session में कोई प्रोग्राम लिखने के लिए सबसे पहले (IDEL) जो Python का GUI होता है, उसे open करना होगा।

Start Menu
All Programs
Python 3
IDLE

अब आपके Screen पर एक विन्डो खुल जायेगी। उस विन्डो (Window) को Python Shell कहा जाता है। अब यदि आप Command promot को open करके

```
print ("Hello Python")
```

टाइप करके Enter Key को Press करते ुएँ तो Interpreter respond करेगा और आपके विन्डो पर

```
Hello Python
```

दिखेगा (Hello Python)। यह एक प्रकार का Hello Word print करने का प्रोग्राम है।

Python एक Case sensitive language है। यदि आप print के स्थान पर Print या PRINT लिखेंगे तो यह प्रोग्राम कार्य नहीं करेगा।

2. Script Mode : जैसा कि हमने देखा Interactive mode आपको तत्काल (immediate) Feedback देता है। यह महत्वपूर्ण है क्योंकि आप तुरन्त प्रोग्राम का परिणाम (result/output) देख सकते हैं। लेकिन Interactive mode में यह सुविधा नहीं होती है कि आपने जो प्रोग्राम लिखा है, उसे सेव (Save) करके बाद में run कर सकें।

इसके लिए Python's IDLE का Script mode बनाया गया है, जिसमें आप कोई प्रोग्राम लिखकर उसे Save कर सकते हैं। बाद में edit, load या run कर सकते हैं।

यह आपके लिए Word processor की तरह है, जिसमें कोई प्रोग्राम लिखते समय उसमें find and replace, copy, cut, paste आदि functions का प्रयोग कर सकते हैं।

कोई प्रोग्राम लिखने के लिए आप एक Script Mode Window को open कर सकते हैं। File Menu से आप एक नई window open कर सकते हैं। यह window बिल्कुल note pad या word pad की तरह होती है। अब इसमें भी एक Program लिखते हैं तथा

```
Print ("Hello Python")
```

अब Enter Key press करते हैं। लेकिन अब कुछ नहीं होता है और न ही कोई आउटपुट हमें दिखाई देता है। अब इस प्रोग्राम को सबसे पहले Save करना होगा। फिर आप इसे run कर सकते हैं। इसके लिए :

goto file menu

Save As

Hello.py

अब आप का प्रोग्राम कम्प्यूटर में सेव (save) हो जायेगा।

Note : Python का कोई भी प्रोग्राम Save करते समय उसका file extension .py ही रखना होता है, जिससे IDLE Python की program की file को आसानी से recognize कर सकें।

अब प्रोग्राम को run करने के लिए run module पर click करना होगा, तब आपकी window पर प्रोग्राम का परिणाम (output) दिखाई देगा।

```
Hello Python
```

IDLE के प्रकार

Ipython तथा Jupyter notebooks दो बहुत ही महत्त्वपूर्ण Development environment हैं जो अपनी खूबियों की वजह से प्रचलन में हैं। कोई Software तैयार करते समय कोई भी यूजर ज्यादा ritchy features के IDE को प्रयोग करना चाहता है। IDE के कुछ प्रमुख उदाहरण निम्नवत् हैं—

1. PyDev (Free) IDE जो Eclipse platform पर Built है।
2. PyCharm from JetBrains (Free open source)
3. Python Tools for Visual Studio (For windows)
4. Spyder (Free, Anide Shiped with Anaconda)
5. Komodo IDE (Commercial)

1.7 Getting Help in Python

Python interpreter में एक built-in function help एक object के रूप में दिया गया होता है। यह interactive use के लिए उपलब्ध होता है, जो help system को invoke करता है। इसे invoke करने के लिए help या help ('object') type करना होता है।

तथा इससे exit करने के लिए 'q' को press करना होता है।

>>> help ('for') – This display the help for function.

1.8 Python Reserved Words

दूसरी Programming languages की भाँति Python में भी कुछ reserved या keyword होते हैं,जिन्हें हम सामान्य identifier के रूप में प्रयोग नहीं कर सकते हैं।

and	del	for	is	raise
assert	elit	from	lambda	return
break	else	global	not	try
class	except	it	or	while
continue	exec	import	pass	yield
det	finally	in	print	false
true	nonlocal	with		

Table : Python Reserved Words

Python में कुल 33 keyword हैं। लेकिन उनकी संख्या घट या बढ़ सकती है।

True, False, None को छोड़कर सभी keywords lowercase यानि छोटे अक्षरों में होते हैं।

Keywords in Python :

- ❖ Keywords are the reserved words.
- ❖ They are used to define the syntax and structure of Python language.
- ❖ We cannot use a keyword as a Variable name, Function name or any other identifier.
- ❖ In Python, keywords are case sensitive.
- ❖ All the keywords except True, False and None are in lowercase.
- ❖ There are 35 keywords in Python.

Keyword.is key word (X)

- ❖ Return true if X is a Python keyword.

What is identifier in Python :

- ❖ An idenfifier is a name used to identify a variable, function, class, module or other object.

Ex. – X = 10 X

10

Y = 20 Y

20

1.9 Naming Conventions

"Names" किसी भी प्रोग्रामिंग लैंग्वेज का बहुत ही महत्त्वपूर्ण हिस्सा होता है। हमें Module को एक नाम (उद्या० Covert) तथा Module के अन्दर Function देना होता है।

Variable : Variable का प्रयोग किसी Names को Value assign करने के लिए किया जाता है। जैसे C, F आदि।

Identifier : Python में किसी variable, function, class, module या अन्य object का जो नाम (Name) रखा जाता है या प्रयोग किया जाता है, उसे ही हम identifier कहते हैं।

लेकिन identifier बनाये जाने के कुछ नियम निर्धारित किये गये हैं जो अग्रवत् हैं—

1. कोई identifier, lowercase (a to z), uppercase (A to Z), digit (0 से 9), underscore (_) का एक मिश्रण (combination) होता है। या इनसे ही मिलकर बना होता है। उदाहरण—my_car, vor_1, print123, x_19 आदि।
2. कोई भी identifier कभी भी digit (0 से 9) से नहीं शुरू होगा। जैसे 1var एक invalid तथा var1 एक valid identifier है।
3. Keywords या Reserved word का प्रयोग हम identifier के रूप में नहीं कर सकते हैं।
4. Python में किसी भी identifier के अन्दर !, #, @, $, % आदि विशेष character को नहीं रखा जाना चाहिए।
5. किसी भी identifier के बीच में जगह (Space) नहीं होनी चाहिए।
6. Identifier की length की कोई सीमा निर्धारित नहीं की गई है।
7. Python एक case sesitive programming language होने के कारण Name एवं name दो अलग-अलग identifier माने जाते हैं।

Identifier	Validity	Reason
1. Roundoff	Valid	
2. 'Round off'	Invalid	–Use of ' '
3. Sum x	Invalid	–Space
4. 2020 worldcup	Invalid	–Digit first
5. Covid-19	Valid	
6. True	Invalid	–Keyword
7. My@name	Invalid	–Special char–@
8. Java2Shore	Valid	
9. Cash	Invalid	–Special char $
10. def	Invalid	–Keyword

z = x + y   z [30]

– It is a user defined name.

Rules for Writing Identifiers :

1. Identifiers can be combination of letters in lowercase (a to z) or uppercase (A to Z) or digit (0 to 9) or an underscore :

a = 10	✓
A = 10	✓
a1 = 20	✓
Aa = 20	✓
– = 20	✓
a_ = 20	✓
x–y = 100	✗

2. An identifier cannot start with a digit.

   Ex : 1x = 100 ✗

   x1 = 100 ✓

3. Keywords cannot be used as identifiers.

   Ex. : If = 10 ✗

   break = 20 ✗

Identifier naming convention for Python

- Class, name, start with an uppercase letter and all other identifiers with a lowercase letter.
- Starting an identifier with a single leading underscore indicates by convention that the identifier is meant to be 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.

Use of Python Language in Industry

- GUI based Desktop applications
- Image processing and graphic design applications
- Scientific and computational applications
- Games
- Web frameworks and web applications
- Enterprise and business applications
- Operating systems
- Language Development
- For Artificial Intelligence and Machine learning
- DIY projects such as Robots, Remote-controlled toys and video game consoles
- Prototyping

परिशिष्ट-I

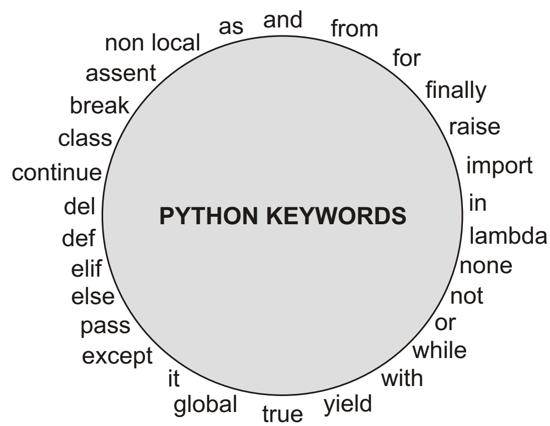
Keywords in Python

Keywords reserved words होते हैं।

Computer programming language में predefined words के set शामिल होते हैं जिन्हें keywords कहा जाता है। प्रत्येक keyword के use के लिए निर्धारित नियम को syntax कहा जाता है।

प्रत्येक keyword का एक special अर्थ और एक special operation होता है क्योंकि किसी भी keyword के नाम का प्रयोग python program में किसी identifier जैसे—variable, function, class name आदि के लिए नहीं किय जा सकता है।

Python में लगभग 33 keywords predefined हैं।



1. 'and' (Logical AND) Keyword

'and' keyword का use जब दो य दो से अधिक operands true होते हैं तब true return किया जाता है।

0 = True

1 = False

Truth Table of 'and'

x	y	x and y
0	0	0
0	1	1
1	0	1
1	1	1

Source Code :

```
Print(5! = 6 and 5 = = 6)          # False
Print(5! = 6 and 5 = = 5)          # True
Print(5! = 6 and 5 = = 6 and 4 < 6)     # False
```

Output : False
True
False

2. 'as' Keyword

'as' keyword का इस्तेमाल module को कोई नाम या उपनाम (alias) नाम देकर import किया जाता है।

Source Code :

```
import keyword as pykey
print(pykey.kwlist)
```

Output : ['False', 'None', 'True', 'and', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

3. 'assert' Keyword

Python program में debugging के लिए 'assert' keyword का use किया जाता है।

'assert' keyword के साथ जब condition true होतीं है तब कुछ नहीं होता और जब condition false होती है तब 'Assertion Error' occur होता है।

Source Code :

```
assert 4 = = 4   #No Exception
assert 4 = = 5   # Assertion Error Exception occured
```

Ouput : assert 4 = = 5 #Assertion Error Exception Occured Assertion Error

4. 'break' Keyword

'break' keyword, loop execution और transfer control को loop के अन्त तक समाप्त करने के लिए use किया जाता है।

Source Code :

```
for i in range (5) :
    if (i = = 3) :
    break
    print(i)
print("End of execution")
```

Output :
```
0
1
2
End of execution
```

5. 'class' Keyword

User के द्वारा classes को define करने के लिए 'class' keyword का use किया जाता है।

Classes में कुछ attributes और कुछ methods होते हैं। Classes 'OOP' में सबसे Important part है।

Source Code :

```
Class my Class :
    def func ( ) :
        print("Hello")
Myclass.func()
```

Output : Hello

6. 'Continue' Keyword

'Continue' keyword का प्रयोग iterate हो रहे कुछ statements को skip करने के लिए किया जाता है।

'Continue' का प्रयोग iterate न हो रहे loop को iterate करने के लिए किया जाता है।

Source Code :

```
a = 0
while (a < 10) :
if (a = = s) :
```

```
    print ("skipped value of a is" a)
    a = a + 1
    continue
    print ("value of a is", a)
    a = a + 1
```

Output : Value of a is 0
Value of a is 1
Value of a is 2
Value of a is 3
Value of a is 4
Value of a is 5
Value of a is 6
Value of a is 7
Value of a is 8
Value of a is 9

7. 'def' Keyword

'def' keyword का use user-defined function बनाने के लिए किया जाता है।

Source Code :

```
def func(a) :
    print("value of a is", a)
fun(5)
```

Output : Value of a is 5

8. 'del' Keyword

Python 'del' keyword का अर्थ delete होता है।

'del' keyword का प्रयोग list, tuple, dictionary या किसी और collection में element को delete करने के लिए किया जाता है।

Source Code :

```
list = [1, 4, 3, 5, 7]
del list [2]
print (list)
```

Ouput : [1, 4, 5, 7]

'del' keyword के अलावा remove () function का भी प्रयोग किया जाता है।

Source Code :

```
list = [1, 4, 3, 5, 7]
list.remove (3)
print(list)
```

Output : [1, 4, 5, 7]

9. 'elif' Keyword

'elif' keyword का प्रयोग if और else keyword के साथ किया जाता है। एक से अधिक conditions को check करने के लिए 'elif' का प्रयोग किया जाता है।

Source Code :

```
a = 11
if (a < 10)
    print ("a is less than 10")
elif (a > 10) :
    print("a is greater than 10")
else :
    print("a is greater than 10")
```

Output : a is greater than 10

10. 'else' Keyword

'else' keyword का प्रयोग if, elif या सिर्फ if के साथ किया जाता है। अगर if और elif की condition false होती है तो else का statement execute होता है।

Source Code :

```
a = 11
if (a%2 = = 0) :
    print ("Number of even.")
else :
    print ("Number of odd.")
```

Output : Number is odd.

11. 'except' Keyword

'except' Keyword का प्रयोग exception/error handling के लिए किया जाता है। try clause के साथ except का प्रयोग किया जाता है।

Example में try block में जो भी exception occur होगा, उसका तुरन्त block execute हो जायेगा।

Example 1 :

Source Code :

```
try :
a
except Zero Division Error :
    print ("Divided byZero Error")
except Name Error
    print ("as is not found")
```

Output : a is not found

Example 2 :

Source Code :

```
try :
```

```
    4/0
except ZeroDivisionError :
    print ("Divided by ZeroError")
except NameError :
    print("a is not found")
Output : Divided by Zero Error
```

12. 'nonlocal' Keyword

'nonlocal' keyword का प्रयोग nested function के लिए किया जाता है।

'nonlocal' variable global नहीं होते। अगर inner function में उनको लिया जाता है तो outer functions में उनकी value परिवर्तित नहीं होती है।

Without nonlocal

Source Code :

```
Var = 2
    def main func () :
        var = 5
    def subfunc ( ):
        var = 10
        print("SubFun C :", var)
SubFunc()
```

'False' keyword boolean false का वर्णन करता है। अगर दी गई condition या statement गलत हो जाता है तो False; return किया जाता है। False '0' के बराबर होता है।

Source Code :

```
print(False = = 0)
print(4 = = 5)
```

Output : True
True

14. 'Finally' Keyword

'Finally' keyword का प्रयोग 'try-except' या 'try' block के साथ किया जाता है।

exception/error हो या ना हो तो finally block execute होता है।

Source Code :

```
try :
    a
    except NameError :
    print("NameError Exception Occured")
Finally :
    print("Always executed")
```

Output : NameError Exception Occured
Always executed

```
    print("mainFunC :", var)
mainFunc()
print("global : ", var)
```

Output : SubFunC : 10
mainFunC : 5
global : 2

With nonlocal

Source Code :

```
Var = 2
def mainFunc() :
    Var = 5
    def SubFunc() :
    nonlocal var
    Var = 10
    print ("SubFunC :", var)
SubFunc()
print ("global", var)
```

Output :

SubFunc : 10

MainFunc : 10

Global : 2

15. 'for' Keyword

'for' Keyword का प्रयोग looping के लिए किया जाता है। Collection के elements को iterate करने के लिए 'for' loop का प्रयोग किया जाता है।

Source code :

```
list = ["Rakesh", "Ramesh", "Suresh"]
for n in list
        print(n)
```

Output : Rakesh
Ramesh
Suresh

16. 'from' Keyword

'from' Keyword का प्रयोग 'import' keyword के साथ किया जाता है। कुछ special function या attribute को import करने के लिए 'from' का प्रयोग किया जाता है।

अगर सिर्फ 'Module' को import किया जाता है और उसमें से fun1 () को लेना पड़ता है तो + 'Module.Func ()' लिखना पड़ता है।

Module.p4

(i) Source Code :

```
def funC1() :
    print("I am in funC1")
def funC2() :
    print("I am in funC2")
```

Sample.py

Source Code :

```
from Module import FunC1
funC(1)
```

Output : I am in fun 1

Example 2.

Source Code :

```
From math import sqrt
print (sqrt(25))
```

Output : 5

17. 'Global' Keyword

'Global' Keyword का प्रयोग function के variable में global बनाने के लिए किया जाता है।

Differnce between Local and Global variable in fun

test1.py

Source Code :

```
def funC( ) :
string = "Local variable"
print(string)
funC()
print(string)
```

Output : Local variable

print(string)

Name Error : name 'string' is not defined

18. 'if' Keyword

'if' keyword का प्रयोग if statement, if_else statement और if_elif_else statement में किया जाता है।

Source Code :

```
var = 0
if (var = = 0):
print ("var is equal to 0")
```

Output : var is equal to 0

19. 'Import' Keyword

'Import' keyword का प्रयोग Module को Current program पर import करने के लिए किया जाता है। Example पर math module को import किया जाता है।

Source Code :

```
Import math
print(Math.sqrt (16))
```

Output : 4.0

20. 'In' Keyword

'In' keyword का प्रयोग collections (list, tuple, string, dictionary) में element है या नहीं, यह check करके boolean value return करता है। For loop में भी 'in' keyword का होना अनिवार्य होता है।

Source Code :

```
list = [6, 8, 4, 7, 8]
print = (7 in list) # True
print = (10 in list) # False
```

Output :

```
True
False
```

21. 'is' Keyword

'is' keyword का प्रयोग object equality के लिए किया जाता है। ये keyword '= =' operator के थोड़ा–बहुत समान होता है। अगर दो objects equal होते हैं तो 'True' return होता है। यदि equal नहीं होते हैं तो 'false' return होता है।

Source Code :

```
print("one" is "are") # True
print ("one" is 1) # False
print ("" is "") # True
print ("" = = "") # True
print ([] is []) # False
print ([] = = []) # False
```

Output :

```
True
False
True
True
True
False
True
```

22. 'lambda' Keyword

'lambda' Keyword से anonymous (without name) function को create किया जाता है। name वाले function में 'def' keyword का प्रयोग किया जाता है और anonymous function में 'def' के बजाय 'lambda' keyword का प्रयोग किया जाता है। lambda function के लिए return statement नहीं होता है। Function के arguments function होने के कारण इनको किसी variable पर assign किया जाता है और उस variable के द्वारा call किया जाता है।

आगे दिए गए दोनों Examples का मतलब एक ही है।

Normal Function Example

Source Code :

```
def funC(a, b) :
    return a*b
print (funC (5, 5))
```

Output : 25

23. 'None' Keyword

Python में 'None' keyword का प्रयोग किया जाता है। लेकिन इसकी programming languages में 'null' keyword का प्रयोग किया जाता है। 'None' का मतलब कोई value नहीं होती। अगर function कोई value retrun नहीं करता है तो 'None' return करता है।

Source Code :

```
print ("None = =")
print (None = = 0)
print (None = = None)
print (type (None))
```

Output : False

False

False

<class 'none type'>

24. 'not' (Logical NOT) Keyword

'not' keyword का प्रयोग दिए गए operand को True से False या False से true में convert करता है।

0 = True

1 = False

True

Table of

x	not	x
0	1	
1	0	

Source Code :

```
print (not (5! = 6)      # False
print (not (3! = 5)      # True
```

Output : False

True

25. 'or' (Logical OR) Keyword

'or' keyword का प्रयोग जब दिए गए oprands में से एक भी oprand true होता है, तब true return किया जाता है।

0 = True

1 = False

True Table

of

'OR'

x	y	x or y
0	0	0
0	1	0
1	0	0
1	1	1

Source Code :

```
print(5! = 6 or 5 = = 6)      # True
print(5! = 6 or 5 = = 5)      # True
print(5! = 6 or 5 = = 6 or 4 < 6)      # True
print(5! = 6 or 5 > 6 or 6 < 6)      # False
```

Output : True
True
True
False

26. 'pass' Keyword

'pass' keyword का प्रयोग loops, functions और classes को रिक्त रखने के लिए होता है। ये एक null statement/block होता है। 'pass' का इस्तेमाल जब किया जाता है, तब ये कुछ नहीं करता है।

pass statement का इस्तेमाल एक ही line पर या seperate line पर भी किया जाता है।

Source Code :

```
def func() : pass
while 4 < 5 :
pass
```

Output :

27. 'Raise' Keyword

'raise' keyword का प्रयोग Exception handing के लिए किया जाता है। raise keyword से custom exception/error का निर्माण किया जाता है। ज्यादातर programming languages में 'throw' keyword का इस्तेमाल किया जाता है। लेकिन Python में 'raise' का इस्तेमाल किया जाता है।

Source Code :

```
1. a = 5
2. if (a > 2) :
3. raise Exception ('a should not exceed 2')
```

Output :

```
Exception : a should not exceed 2
```

28. return keyword

return keyword का प्रयोग function के आखिर में किया जाता है। Function को अगर कोई value return नहीं की जाती है तो null/none return किया जाता है।

Source Code :

```
def func(a) :
b = a*a
    return b
print (func(5))
```

Output : 25

29. 'True' Keyword

'True' keyword एक Boolen value है। logical या comparism operations में boobean value (True, False) return किया जाता है। True की value '1' और False की value '0' होती है।

Source Code :

```
print (True =  =1)
print (false = = 0)
print (1 = = 1)
print (True + True)
```

Output :

```
True
True
True
2
```

30. 'Try' Keyword

'Try' keyword का प्रयोग excepting handing में किया जाता है जो code exception check करता है। इसमें code को try block पर लिखा जाता है।

Source Code :

```
        Try :
        a
except Name Error :
print ("value of a is not defined")
```

Output : Value of a is not defined

31. 'While' Keyword

'While' keyword का प्रयोग looping के लिए किया जाता है। जब तक while condition true होती है, तब तक statement iterate होता रहता है और अगर condition false या loop break किया जाता है, तब loop का iteration बन्द हो जाता है।

Source Code :

```
a = 0
while a < 10 :
    print ("value of a is", a)
    a = a + 1
```

Output :

```
Value of a is 0
Value of a is 1
Value of a is 2
Value of a is 3
Value of a is 4
Value of a is 6
Value of a is 7
Value of a is 8
Value of a is 9
```

32. 'with' Keyword

'with' keyword का प्रयोग File handling के लिए किया जाता है। जब file के लिए 'with-as' statement का प्रयोग किया जाता है तब file को close करने की जरूरत नहीं पड़ती है।

नीचे दिए गए दोनों examples का मतलब एक ही है।

(1) without 'with'

Source code :

```
file = open ("text file.text", "w")
file.write ("Hellow world")
file.close ()
```

textfile.txt

```
Hellow world
```

(2) with 'with'

Source Code :

```
with open ("text file.text", "w") as file :
    file.write ("Hello world")
```

text file.txt

```
Hello world
```

33. 'yield' Keyword

'yield' keyword का प्रयोग function के लिए return statement जैसे किया जाता है। 'yield' keyword generator को return करता है। return statement; function के आखिर में function को suspend करके वहाँ से आगे वापस start होता है। Normal function जहाँ पर खत्म होता है, वहाँ पर वापस नहीं आता है।

Source Code :

```
def func-name ( ) :
```

```
        yield 5
        yield 10
        yield 15
    for y in func-Name ( ) :
        print (4)
```

Output : 5
10
15

next () function का प्रयोग 'yield' के लिए किया जाता है। next () function generate object को लेकर अगली value return करता है।

Source Code :

```
def func-name ( ) :
    yield 5
    yield 10
    yield 15
y = func-name ( )
    print (next (4))
    print (next (4))
    print (next (4))
```

Output : 5
10
15

महत्त्वपूर्ण प्रोग्राम

Program 1

#Write a Program to print Hello World and print your name

```
print("Hello World")
print("Rajshree")
#output:
```

अभ्यास प्रश्नावली

✍ A. खाली स्थान भरो (Fill in the blanks)

1. Python was created by ……… .
2. To start python from the command prompt, use the command ……… .
3. To run python script file named t.py, use the command ……… .
4. A python paragraph comment uses the style ……… .
5. In python, a syntax error is detected by the ……… at ……… .

✍ B. लघु उत्तरीय प्रश्न (Short Answer Questions)

1. Name some features of Python.
2. What is the purpose of PYTHONPATH variable ?
3. What is the difference between tuples and lists ?
4. What are the applications of Python?
5. Differentiate between Python 2 and Python 3.
6. Why it is necessary to define environment variables in python ?
7. Explain Python reserved words with example.
8. What are the naming conventions used in Python ?
9. What is IDLE ?
10. How is Python an interpreted language ?
11. Give the brief history of python.
12. Write short note on environment variables.
13. Explain IDLE.
14. Explain help function in python.
15. Why python is dynamically typed ?
16. Name any 5 reserved keywords in python.
17. Explain all the general naming conventions in python.
18. How can we install python in windows ?
19. Write a short note on IDLE editor window?
20. How can you documentation ?

✍ C. महत्वपूर्ण लघु उत्तरीय प्रश्न (Important Short Answer Questions)

1. पाइथन का परिचय देते हुए इतिहास के बारे में बताइए।
2. पाइथन प्रोग्रामिंग की विशेषताओं का उल्लेख कीजिए।
3. लोकल कम्प्यूटर पर पाइथन इंस्टॉल करने की विधि बताइए।
4. पाइथन के विभिन्न अनुप्रयोगों का उल्लेख कीजिए।
5. पाइथन keyword से आप क्या समझते हैं? विस्तारपूर्वक वर्णन कीजिए।

6. पाइथन को dynamically typed क्यों कहा जाता है?
7. IDLE क्या है? इसके मुख्य उद्देश्य बताइए।
8. Keyword से आप क्या समझते हैं? Python में कुल कितने Keyword होते हैं?
9. Python के मुख्य features क्या हैं?
10. Python के Applications को लिखिए।
11. Python में variable को कैसे declare किया जाता है? उदाहरण सहित लिखिए।
12. IDLE से आप क्या समझते हैं? इसकी आवश्यकता एवं औचित्य को लिखिए।
13. Programing एवं scripting से आप क्या समझते हैं? दो अन्तर स्पष्ट कीजिए।
14. Python में Environment variable क्या होते हैं? उदाहरण सहित लिखिए।
15. Python documentation से आप क्या समझते हैं?
16. Dynamically typed language को कब चयन किया जाता है? Python के बारे में आप की क्या राय है?
17. Python में Identifier से आप क्या समझते हैं? इसको बनाने का नियम क्या है?

✍ D. बहुविकल्पीय प्रश्न (Objective Questions)

1. **What is the maximum possible length of an indentifer ?**
   (a) 31 characters (b) 63 characters
   (c) 79 characters (d) None of the mentioned
2. **Which of the following is invalid ?**
   (a) _a = 1 (b) __a = 1
   (c) __str__ = 1 (d) None of the mentioned
3. **What is output for :**
   a = ['he', 'she', 'we']
   ' ' . join (a)
   (a) ['heshewe'] (b) 'heshewe'
   (c) ['he she we'] (d) 'he she we'
4. **Pylab is a package that combine ………, ……… & ……… into a single namespace.**
   (a) Numpy, scipy & matplotlib (b) Numpy, metplotlib & pandas
   (c) Numpy, pandas & matplotlib (d) Numpy, scipy & pandas
5. **Which of the following is an invalid statement ?**
   (a) abc = 10,00,000 (b) a b c = 1000 2000 3000
   (c) a, b, c = 1000, 2000, 3000 (d) a_b_c = 10,00, 000
6. **Which statement is correct ……… ?**
   (a) List is mutable && Tuple is immutable (b) List is immutable && Tuple is mutable
   (c) Both are Immutable (d) Both are Mutable
7. **The format function, when applied on a string returns :**
   (a) list (b) bool
   (c) int (d) str

8. Output of the following code would be ?

```
a = True
b = False
c = True
if not a of b :
    print "a"
elif not a or not b and c :
    print "b"
elif not a or b or not be and a :
    print "c"
else :
    print "d"
```

(a) a (b) b
(c) c (d) d

9. If a = 'cpp', b = 'buzz' then which of the following operation would show 'cppbuzz' as output?

(a) a + b (b) a + " +b
(c) a + " " + b (d) All of the above

10. How to find the last element of list in Python? Assume 'bikes' is the name of list.

(a) bikes [0] (b) bikes [– 1]
(c) bikes [1 pos] (d) bikes [: – 1]

11. Which of the following is correct way to declare string variable in Python?

(a) fruit = 'banana' (b) fruit = "banana"
(c) fruit = banana (d) fruit = (banana)

12. Select the reserved keyword in python :

(a) else (b) import
(c) raise (d) all of these

13. Which operator has higher precedence in following :

(a) % vModulus (b) & BITWISE AND
(c) ** exponent (d) > comparison

14. Which of the following is not a keyword ?

(a) eval (b) assert
(c) nonlocal (d) pass

15. What is the order of precedence in python ?

(i) Parentheses (ii) Exponential
(iii) Multiplication (iv) Division
(v) Addition (vi) Subtraction
(a) i, ii, iii, iv, v, vi (b) ii, i, iii, iv, v, vi
(c) ii, i, iv, iii, v, vi (d) i, ii, iii, iv, vi, v

16. Jython जो कि Python का ही implementation है। किस लॉवेज में लिखा गया है–

(a) C (b) C++

(c) Java (d) C#

17. Python केके सभी Object होते हैं।

(a) either Muable or immutable (b) mutable

(c) immutable (d) none

18. Python support करता है।

(a) Mixed Programming Language (b) GUI programming

(c) Interactive computing (d) All of above

19. निम्न से कौन Scripting language नहीं है–

(a) Java Scrpt (b) C

(c) VBScript (d) Ruby

उत्तरमाला

1.	()	2.	()	3.	()	4.	()	5.	()	6.	()	7.	()	8.	()	9.	()	10.	()
11.	()	12.	()	13.	()	14.	()	15.	()	16.	()	17.	()	18.	()	19.	()		

❑