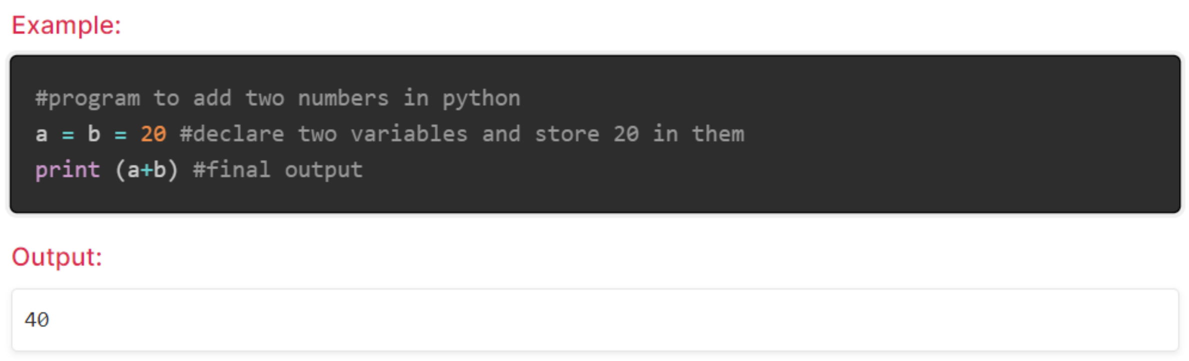
What is Python?

Python is an advanced, interpreted programming language known for its readability and simplicity. It supports various programming paradigms, such as procedural, object-oriented, and functional. Python's wide range of applications, from web development and app creation to data analytics and artificial intelligence, underscore its usefulness. Its robust assortment of libraries and frameworks contributes to its popularity among novices and professionals.

Python is a programming language that includes features of C and Java. It provides the style of writing elegant code like C, and for object-oriented programming, it offers classes and objects like Java.

In Python, the program to add two numbers will be as follows:



Example:

```
#program to add two numbers in python
a = b = 20 #declare two variables and store 20 in them
print (a+b) #final output
```

Output:

```
40
```

Facts about python:

- At the National Research Institute for Mathematics and Computer Science in the Netherlands, Guido van Rossum created Python in the late 1980s as an alternative to the ABC language that could handle exceptions and interfaces.
- Python is derived from programming languages such as ABC, Modula 3, small talk, and Algol-68.
- Van Rossum picked the name Python for the new language from a TV show, Monty Python's Flying Circus.
- A Python page is a file with a .py extension that can contain a combination of HTML tags and Python scripts.
- Python is an open-source scripting language.
- Python is open-source, meaning anyone can download it freely from www.Python.org and use it to develop programs. Its source code can be accessed and modified as required in the project.
- Python is one of the official languages used by the tech giant Google.

Features of Python

Python is gaining popularity in the programming community; there are many reasons behind this.

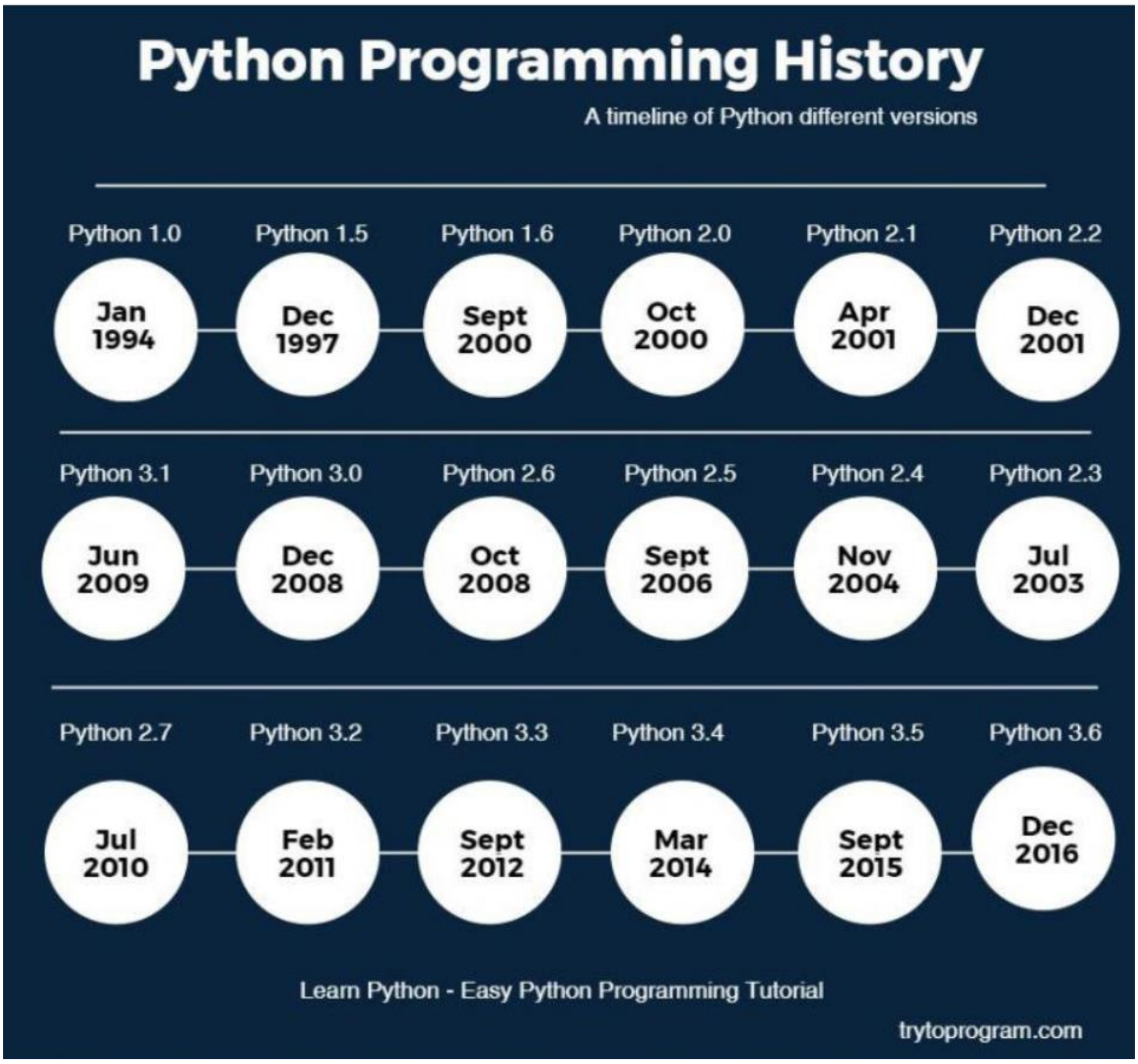
- **Interpreted Language**: Python is processed at runtime by Python Interpreter.
- **Object-Oriented Language**: It supports object-oriented features and techniques of programming.
- **Interactive Programming Language**: Users can directly interact with the Python interpreter to write programs.
- **Easy language**: Python is simple to learn, particularly for newcomers.
- **Straightforward Syntax**: The formation of Python syntax is simple, making it popular.
- **Easy to read**: Python source code is clearly defined and visible.
- **Portable:** Python codes can be run on various hardware platforms with the same interface.
- **Extendable**: Users can add low level-modules to the Python interpreter.
- **Scalable**: Python provides an improved structure for supporting large programs than shell scripts.

Evolution of Python

The language was finally released in 1991. When it was released, it used a lot fewer codes to express the concepts, when we compare it with Java, C++ & C. Its design philosophy was quite good too. Its main objective is to provide code readability and advanced developer productivity. When it was released, it had more than enough capability to provide classes with inheritance, several core data types of exception handling and functions.

The first ever version of Python(i.e. Python 1.0) was introduced in 1991. Since its inception and introduction of Version 1, the evolution of Python has reached up to Version 3.x (till 2017).

Here is the brief chart depicting the timeline of the release of different versions of Python programming language.



Execution Modes in python

As we know that python uses an interpreter for the execution of source code. now, there are two ways in which we can use the interpreter. They are as follows:

- Interactive Mode
- Script Mode

Interactive Mode

In this mode, we can execute a single statement at a time. Moreover, to use the interactive mode, we have to write the statement in front of '>>>' and press enter

This results in the output of that particular statement immediately. This mode is easy and convenient to use to see the instant output. But, at the same time, we cannot save the whole code and have to write it again and again to execute it.

Script Mode

In this mode we have to write the whole source code and save it as a Python source code file. Furthermore, we can execute this file using the interpreter. Moreover, we save the python source code file with the extension ‘.py’.

Python Identifiers:

All the variables, class, object, functions, lists, dictionaries etc. in Python are together termed as Python Identifiers. Identifiers are the basis of any Python program. Almost every Python Code uses some or other identifiers.

Rules for using Python Identifiers:

- An identifier name should not be a keyword.
- An identifier name can begin with a letter or an underscore only.
- An identifier name can contain both numbers and letters along with underscores (A-z, 0-9, and _).
- An identifier name in Python is case-sensitive i.e, sum and Sum are two different identifier.

Reserved keywords:

Python has a set of keywords that are reserved words that cannot be used as variable names, function names, or any other identifiers:

Keyword	**Description**
and	A logical operator
as	To create an alias
assert	For debugging
break	To break out of a loop
class	To define a class
continue	To continue to the next iteration of a loop
def	To define a function
del	To delete an object
elif	Used in conditional statements, same as else if
else	Used in conditional statements
except	Used with exceptions, what to do when an exception occurs

False	Boolean value, result of comparison operations
finally	Used with exceptions, a block of code that will be executed no matter if there is an exc
for	To create a for loop
from	To import specific parts of a module
global	To declare a global variable
if	To make a conditional statement
import	To import a module
in	To check if a value is present in a list, tuple, etc.
is	To test if two variables are equal
lambda	To create an anonymous function
None	Represents a null value
nonlocal	To declare a non-local variable

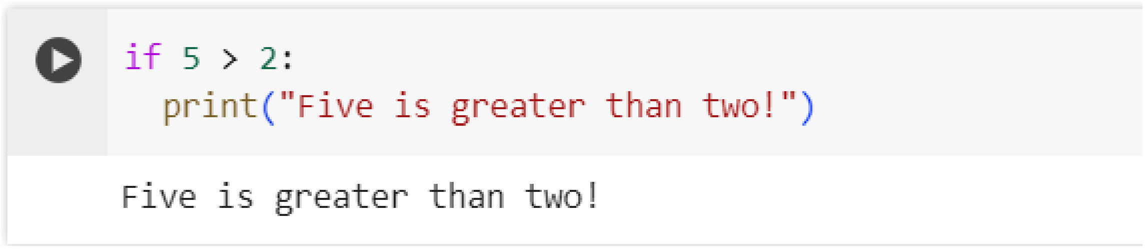
not	A logical operator
or	A logical operator
pass	A null statement, a statement that will do nothing
raise	To raise an exception
return	To exit a function and return a value
True	Boolean value, result of comparison operations
try	To make a try...except statement
while	To create a while loop
With	Used to simplify exception handling
Yield	To end a function, returns a generator

Python Indentation

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

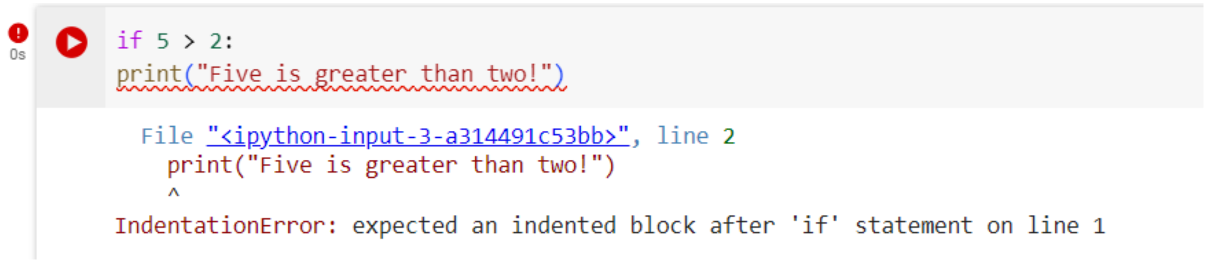
Python uses indentation to indicate a block of code.



```python
if 5 > 2:
  print("Five is greater than two!")
```

```
Five is greater than two!
```

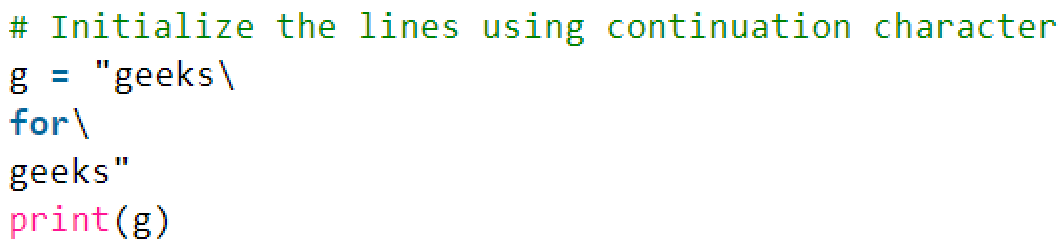
Python will give you an error if you skip the indentation:



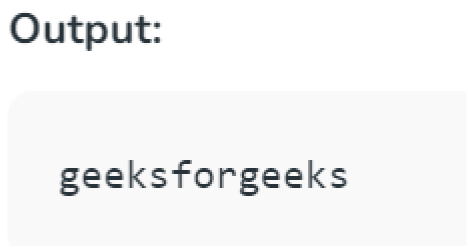
```python
if 5 > 2:
print("Five is greater than two!")
```

```
  File "<ipython-input-3-a314491c53bb>", line 2
    print("Five is greater than two!")
    ^
IndentationError: expected an indented block after 'if' statement on line 1
```

Multi-line Statement in Python:
In Python, the statements are usually written in a single line and the last character of these lines is newline. To extend the statement to one or more lines we can use braces {}, parentheses (), square [], semi-colon ";", and continuation character slash "\". we can use any of these according to our requirement in the code. With the line continuation character, we can explicitly divide a long statement into numerous lines (\).



```python
# Initialize the lines using continuation character
g = "geeks\
for\
geeks"
print(g)
```



Output:

```
geeksforgeeks
```

Quotation:
Single and Double Quotes | Python
Python string functions are very popular. There are two ways to represent strings in python. String is enclosed either with single quotes or double quotes. Both the ways (single or double quotes) are correct depending upon the requirement. Sometimes we have to use quotes (single or double quotes) together in the same string, in such cases, we use single and double quotes alternatively so that they can be distinguished.

```
#Gives Error
print('It's python')
```

Explanation **–** It gives an invalid syntax error. Because a single quote after “it” is considered the end of the string and rest part is not part of a string. It can be corrected as:

```
print("It's Python !")
```

Output:

```
It's Python!
```

Python Comments

Comments can be used to explain Python code.

Comments can be used to make the code more readable.

Comments can be used to prevent execution when testing code.

Creating a Comment

Comments starts with a #, and Python will ignore them:

```
#This is a comment
print("Hello, World!")
```

Output

```
Hello, World!
```

A comment does not have to be text that explains the code, it can also be used to prevent Python from executing code:

```
#print("Hello, World!")
print("Cheers, Mate!")
```

```
Cheers, Mate!
```

Multiline comments:

```
"""
This is a comment
written in
more than just one line
"""
print("Hello, World!")
```

```
Hello, World!
```

Using Blank Lines:

The print() function in Python is often used to display output on the screen. It can also be used to print an empty line.

The simplest way to do this is to call the print() function without arguments.

Code	Output
print()	

The code above calls the print() function with no arguments, resulting in an empty line being printed to the console.

This method is particularly useful if you need to include an empty line between two lines of text:

Code	Output
print(“Hello, World!”)	Hello, World!
print()	
print(“Goodbye, World!”)	Goodbye, World!

The code above will print “Hello, World!” on one line, an empty line on the next, and then “Goodbye, World!” on the following line.

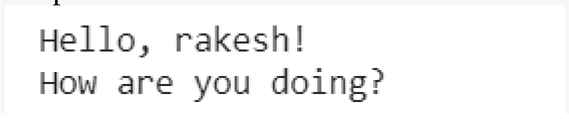
Multiple Statement Groups as Suites

In Python, a suite is a block of code that consists of one or more statements. It is used to group related statements together, and is typically used when defining functions, classes, and control structures.
A suite is typically indented, with each line of the suite being indented the same amount. The indentation is used to indicate that the statements in the suite belong together and should be treated as a single block.
Here is an example of a suite in a function definition:

```
def greet(name):
  print("Hello, " + name + "!")
  print("How are you doing?")

greet("rakesh")
```

Output:



```
Hello, rakesh!
How are you doing?
```

Running Python:

1st way:

Google colab:

1. Search for google colab
2. Register and sign in
3. Open new notebook
4. Type the code and click on debug button

2nd way:

Using Text Based:

If you find that you do not have Python installed on your computer, then you can download it for free from the following website: https://www.python.org/

To Run the Python program in a command prompt

```
C:\Users\Your Name>python helloworld.py
Hello, World!
```

Using IDE:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> print("Hello, World!")
Hello, World!
```