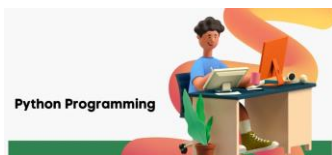


# LECTURE

# FOUR



## Python Programming Language

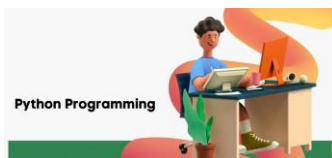
Art is an expression of human creative skill, hence programming is an art. The choice of programming language is important.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas the other languages use punctuations. Python is popular due to its simplicity and robustness.

- ❖ Python is **Interpreted**: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it.
- ❖ Python is **Interactive**: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs .
- ❖ Python is **Object-Oriented**: Python supports Object-Oriented style or technique of programming that encapsulates code within objects .

## History of Python

Python was developed by **Guido van Rossum** in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is written in C. Python is named after a TV Show called 'Monty Python's Flying Circus' (British Comedy Group) and not after Python-the snake.



## Python's versions

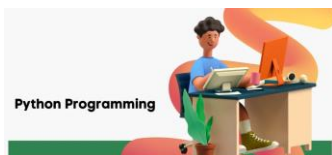
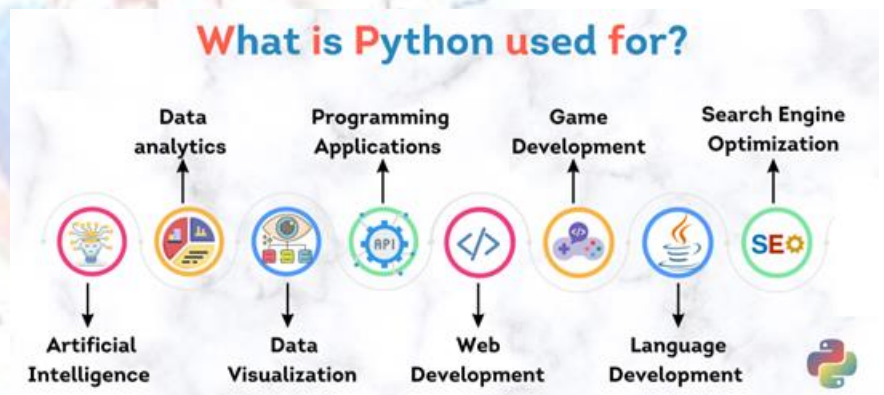
- Python 1.0 was released in November 1994.
- Python 2.0 was released In 2000.
- Python 3.0 was released in 2008.
- Python 3.12 will be released in 2023.

## Some of Python's features

- ❖ **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- ❖ **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
- ❖ **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.

## Some of Use Python

- ❖ Script
- ❖ Database
- ❖ Desktop Application
- ❖ Web Application
- ❖ Mobile Application
- ❖ Artificial intelligent
- ❖ Image processing
- ❖ Security
- ❖ Networks and communication
- ❖ Games
- ❖ Organizational
  - YouTube
  - Facebook
  - Yahoo
  - NASA



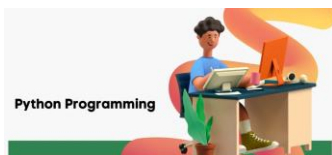
## Development Tools of Python

Programmers have a variety of tools available to enhance the software development process. Some **common tools** include:

- ❖ **Editors:** An editor allows the programmer to enter the program source code and save it to files. Most programming editors increase programmer productivity by using colors to highlight language features.
- ❖ **Interpreters:** An interpreter is like a compiler, in that it translates higher-level source code into target code (usually machine language). Interpreted languages are better suited for dynamic, explorative development, ideal for beginning programmers. Popular scripting languages include Python.
- ❖ **Debuggers:** A debugger allows a programmer to more easily trace a program's execution in order to locate and correct errors (also called bugs) and repairing programs that contain errors.

Many developers use **integrated development environments (IDEs)**. An IDE includes editors, debuggers, and other programming aids in one comprehensive program. Python IDEs include VSCode, pycharm ,IDLE and etc.

Fig. 1.1. While studying this text you will write many Python programs. Once your program is written and you are ready to try it you will tell the Python interpreter to execute your Python program so you can see what it does.





## Writing a Python Program

A program consists of one or more statements. A statement is an instruction that the interpreter executes.

The following statement invokes the **print function** to display a message:

### First\_program.py

```
print ("Hello world!")
```

Then, run the file. You will get the following output.

```
Hello world!
```

Congratulations! You just wrote your first program in Python.

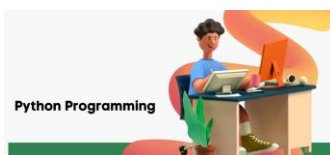
As we can see, it was pretty easy. This is the beauty of Python programming language.

## Python Keywords

- Keywords are the reserved words in Python.
- We cannot use a keyword as a variable name, function name or any other identifier. They are used to define the syntax and structure of the Python language.

### Python keywords

and	del	from	None	try
as	elif	global	nonlocal	True
assert	else	if	not	while
break	except	import	or	with
class	False	in	pass	yield
continue	finally	is	raise	
def	for	lambda	return	





## Python Identifiers

An identifier is a name given to entities like variables, functions, class, etc. It helps to differentiate one entity from another.

### Rules for writing identifiers:-

1. Keywords cannot be used as identifiers.
2. Identifiers can be a combination of letters in lowercase (a to z) or uppercase (A to Z) or digits (0 to 9) or an underscore
3. An identifier cannot start with a digit.
4. We cannot use special symbols like !, @, #, \$, % etc. in our identifier.

	
AbCd	123abc
aBcDa	123_abc
abc123	while
ABC123	1A2B
abc_123	abc@123
ABC_123	\$abc

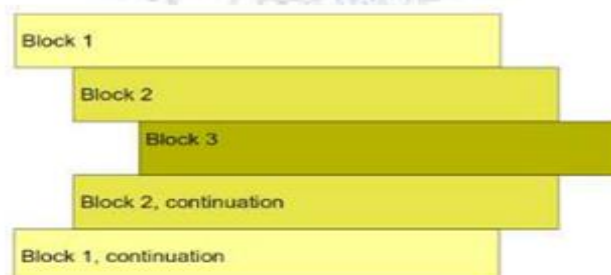
### Key Points to Remember:

- ❖ Python is a case-sensitive language. This means, Variable and variable are not the same.
- ❖ Always name identifiers that make sense.
- ❖ Multiple words can be separated using an underscore.

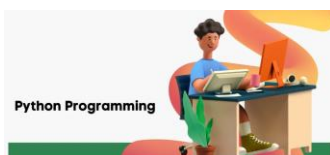
## Python Indentation

Most of the programming languages like C, C++, Java use braces { } to define a block of code. Python uses indentation. Generally four whitespaces are used for indentation and is preferred over tabs.

A block is a group of statements in a program. Usually it consists of at least one statement and of declarations for the block, Python programs get structured through indentation, i.e. code blocks are defined by their indentation.



The enforcement of indentation in Python makes the code look clean.



## Python Comments

Comments are very important while writing a program. It describes what's going on inside a program

In Python, we use the hash (#) symbol to start writing a comment.

Python Interpreter ignores comment.

```
#This is a long comment
```

**Multi-line comments:** If we have comments that extend multiple lines, by using triple quotes, either ''' or ''''

```
"""This is also a
perfect example of
multi-line comments"""
```

## Python Output

The actual syntax of the print() function is

```
print(value(s), sep=' ', end='\n', file=sys.stdout, flush=False)
```

- Here, the value(s) to be printed.
- The sep separator is used between the values. It defaults into a space character.
- After all values are printed, end is printed. It defaults into a new line.
- The file is the object where the values are printed and its default value is sys.stdout (screen).

### Example: print() function

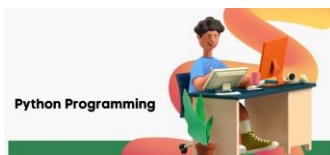
```
print(1,2,3,4)
print(1,2,3,4,sep='*')
print(1,2,3,4,sep='#',end='&')
```

Output:

1 2 3 4

1\*2\*3\*4

1#2#3#4&



**Python Input:** Up till now, our programs were static. To allow flexibility we might want to take the input from the user. In Python, we have the `input()` function to allow this.

The syntax for `input()` is

```
input([prompt])
```

where `prompt` is the string we wish to display on the screen. It is optional.

#### Example: `input()` function

```
x = input('Enter a number: ')\nprint(x)
```

#### Key Points to Remember:

Here, we can see that the entered value **x** is a **string**, not a number. So, we need Python Casting based on idea of program.

