# JUST FOR CODE

## LEARN SOFTWRE ENGINEERING

**Programming**

The method of creating software is called programming. These methods are written in a particular language and that language is called programming language.

**Name of some programming languages**

1. Python
2. C
3. Java
4. Php

**Coding or code**

The words we use while programming are called coding.

**Note :-**

Low-Level language is the only language which can be understood by the computer. Low-level language is also known as Machine Language. The machine language contains only two symbols 1 & 0. All the instructions of machine language are written in the form of binary numbers 1's & 0's.

| High-Level Language<br><br>Write in english | Low-Level Language<br><br>Write in binary |
|---|---|
| These are programmer-friendly languages that are manageable, easy to understand, debug, and widely used in today's times. | These are machine-friendly languages that are very difficult to understand by human beings but easy to interpret by machines. |

| | |
|---|---|
| High-level languages require the use of a compiler or an interpreter for their translation into the machine code. | Low-level language requires an assembler for directly translating the instructions of the machine language. |

| | |
|---|---|
| These are portable from any one device to another. | A user cannot port these from one device to another. |
| High-level languages do not depend on machines. | Low-level languages are machine-dependent and thus very difficult to understand by a normal user. |
| High-level languages take more time for execution as compared to low-level languages because these require a translation program. | The translation speed of low-level languages is very high. |
| One does not require a knowledge of hardware for writing programs. | Having knowledge of hardware is a prerequisite to writing programs. |
| High-level languages do not provide various facilities at the hardware level. | Low-level languages are very close to the hardware. They help in writing various programs at the hardware level. |
| Some examples of high-level languages include python, php, java | Some examples of low-level languages include the Machine language and Assembly language. |

**Compiler**

Compiler is a translater which translates english word into binary (machine code)

**Interpreter**

Interpreter is a translater which translates english word into binary (machine code)

**Differences between compiler and interpreter**

A compiler translates the entire source code in a single run. An interpreter translates the entire source code line by line

**Python definition**

Python is a high-level programming language.

**Python Inventions**

Invented by Guido van rossum on 20 february 1991

## Data
Number character files commonly these are known as data

### Characters
a, b, c, d

## String
Just, for, code,

## Files
test.mp4, test.jpg, test.pdf test.mp3, test.audio

## Statements or instructions
A statement is a single line of code that performs a specific task

## Variables
Variables are the way to store data in programming.

### Lowercase

**All letters in small** **Example** : myvariable

### Uppercase

**All letters in capital** **Example** : MYVARIABLE

### Camelcase

**First letter is small but next word first letter is capital** **Example** : my**V**ariable

### Snakecase

**First letter is capital and also next word first letter is capital** **Example** : **M**y**V**ariable

### Variable declaration

**x = 18**
**Y = 32**

**Variable declaration in shorthand**

x , y = 14 , 16

### Same data in multiple variables

x = y = z = 22

## Keyword

Keywords are predefined, reserved words used in programming that have special meanings to the compiler

## 35 Keywords are there in pyhton

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

## Keyword`s Uses

**def** : def keyword is used to create a function

**return** : return keyword is used return something from a function

**Functions**

A function splits the code into several parts to keep all the code organized

## There are two types of function

1. **User-defined function** : A function that is created by user (**we define and we call**)

2. **Built-in Or Predefined function** : A function that is created by programming language itself (**we only call but we  can not define**)

## Some Built-in functions

1. print()
2. int()
3. float()
4. str()
5. input()

## Data Types

1. String type
2. Numeric type
3. Boolean type
4. Set type
5. Sequence type
6. Mapping type
7. Binary type
8. None type

String Type

| **Type** | **Example** |
|---|---|
| str | “alok kumar”, ‘a’ |

Numeric Type

| **Type** | **Example** |
|---|---|
| int | 53151, -211 |
| float | 120.22, -52.36 |

Boolean Type

| **Type** | **Example** |
|---|---|
| bool | True , False |

**Example to checking data type**

```
type = type(123) print(type)
```

```
type = type(120.36) print(type)
```

```
type = type(“alok kumar”) print(type)
```

```
type = type(True) print(type)
```