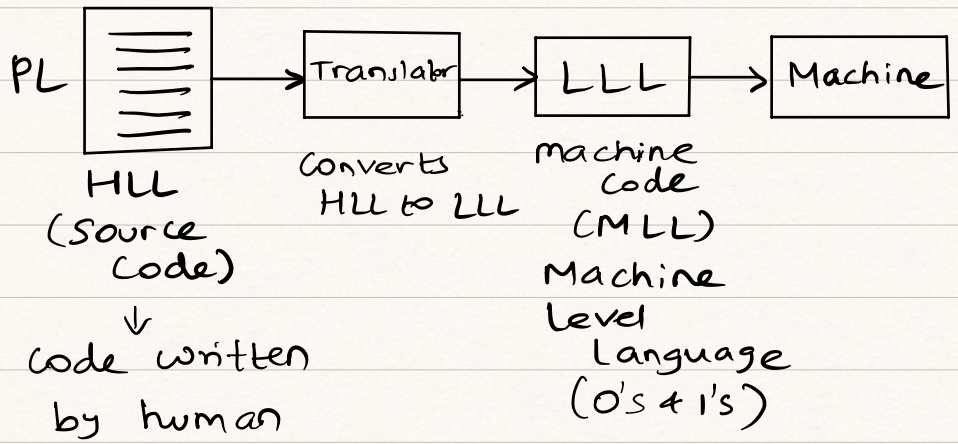
Python

Python Programming Language: It is a Programming Language by using which we write step by step instructions to machine to perform the tasks.



Translator

Compiler	Interpreter
1) will take the entire code and convert to LLL at a time.	will take one line at a time and convert the code to LLL.
2) Hard to debug	Easy to debug

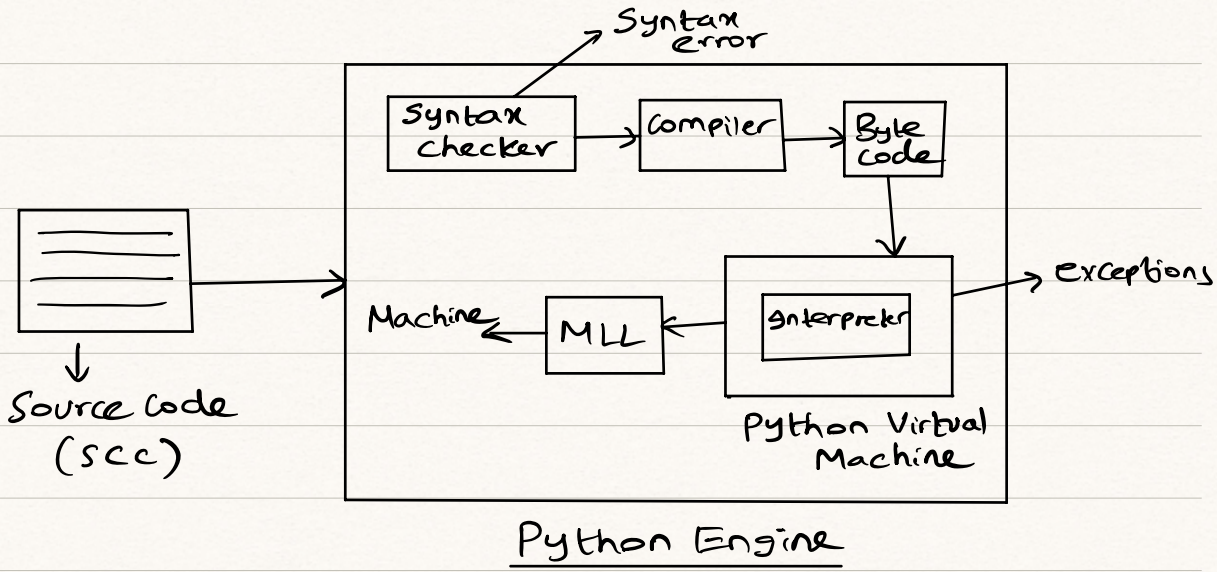
For us to use the python programming language we need to use it in the environment.
So first this environment needs to be installed.

Things to learn:

1) **Syntax (Rules)** : we have to follow these rules to write the Python Programming language.

2) **Code.**

Workflow of Python



1) Source code is written.
2) The code will be passed to the Python Engine.
3) In Python engine the code first goes through the Syntax checker to check syntax.
   - At this stage if there is an error we will get a syntax error.
4) After the syntax check the code will be given to Compiler if no syntax error exists.
5) Compiler converts the source code to the byte

code called intermediate code.

6) since byte code cannot be understood by machines. we have a Python virtual machine.

7) Python virtual machine has a interpreter in it which converts from byte code to machine code and is passed to a machine.

   └ Errors created when converting byte code to machine code are called Exceptions (run-time errors) in Python. Here error is logic error.

* we use both compiler and interpreter here to make it little faster. Also it would give out logical errors.

* Debugging is the process of finding out the error and rectifying it.

Python Environment

two Versions
- Python 2 version
- Python 3 version

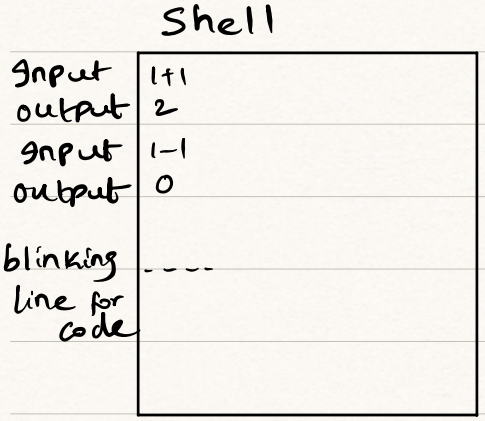
when we install PE (Python Environment) we get a tool called IDLE (Integrated Development and Learning Environment). i.e Learn and apply.

two tools
- Shell
- Document Editor

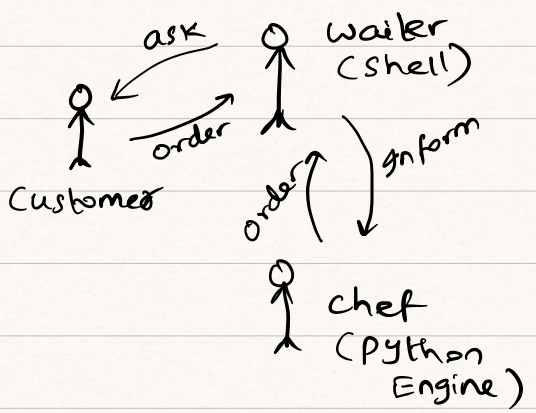
we can use Python Programming language in these tools.

Shell: Tool where we can use Python programming Language.

- Shell can only take single line code at a time.
- For multiple code lines at a time we cannot use shell
- Code once executed cannot be saved
- Single line input and output query.



ex:- Pista house



Document Editor;

1) similar to shell but here we can execute multiple lines of code at a time.
2) Document editor code can be saved for future execution.
3) Python code when it is saved it will be saved with an extention of .py i.e file.py
4) Document editor has use interface problem because input is in one window and output is in another window.

* So instead of shell or Document Editor we use Jupiter Notebook.

Other tools that we can use are

- Visual studio
- Spyder
- Pycharm
- Atom

To install Jupyter Notebook we install an application called Anaconda. With this we get

- Jupyter Notebook
- Visual studio
- Spyder

* Jupyter Notebook will save the file as .ipynb
  .ipynb (i python notebook)
* while using Jupyter Notebook the command prompt should not be closed.
* R Language is best for statistics purpose but cannot be used for deep learning.

Different cases in Naming convention:

In programming spaces are reserved characters. so if the name of the variable is more than just one word we cannot use spaces to seperate them.

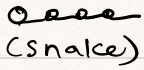
number of digits = 34 ✗

here each word is treated as seperate enlity as space is in between them.

For our program to work correctly we need to remove all spaces and combine the words in to a single string in a specific way. And there are a few ways available to combine them.

- **Snake case**: Snake case seperates each word with an underscore character (_)

ex:-

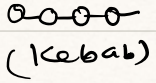

(snake)

number_of_digits = 34

Snake case is used for variables and identifiers.

- **Kebab case**: Simillar to snake case but uses dash (-) character instead

ex:-


(kebab)

number-of-digits = 34

mostly used in URL's

- **Camel case**: when using camel case we start by making the first word lower case. Then we capitalize the first letter of each word that follows.

  ex:- numberOfDigits = 34 (camel)

  used for function names in python.

- **Pascal case**: pascal case is similar to camel case. The only difference is in pascal case even the first letter of the first word is also capital case.

  ex:-

  NumberOfDigits = 34

  used for class names in python.

** we use different cases for different objects to easily diffrentiate them.

Python Syntax

Statement: It is a line (or piece of code) that python can execute

Rule 1 : Single line statement

Single statement should be written in a single line.

ex:- 1+1 ✓

2+2 ✓

1+1 2+2 ✗

Rule 2 : Multi-line statement (Single statement in multiple lines)

whenever we want to write single statement in multiple lines we have to use backward slash [\].

ex:- 1+\ 1+ 1+2\

1+\ 1+ +3\

2 ✓ 2 ✗ 4 ✓

* python statement ends with a token (\n) Newline character. But we can extend the statement over multiple lines using line continuation character (\). This is known as explicit continuation.

* At the end we cannot use back slash [\].

* we can write multi-line statements inside brackets ( ), [ ], { }. whatever we add inside the brackets will be treated as a single statement even if it is placed on multiple lines. This is called implicit continuation.

Rule 3: Multiple statements in a single line

whenever we want to write multiple statements in a single line we have to use semi-colon (;) between the statements.

ex:- 1+1 ; 2+2 ✓

1+1 2+2 ✗

* output is only shown for the last statement.

Rule 4: Comments

These are the statements that the python Engine ignores.

whenever we want to convert a statement in to comment we use hash(#) before the statement.

1+1 ✓

In python we will have only Single line Comments.

1+1 ✓

1+2+3+4 ✓

In python multiline comments are not supported

(""""" """"" triple quote concept does not work)

Rule 5: Indentation

Indentation is used to specify a block of code.
Default indentation is four white spaces.

ex:- if True:
    Print("hi")

four white spaces

we can also use one, two, or three white spaces.
If no white space is used we will get Syntax error.

Rule 6: Keywords

Keywords are the reserved words which will have a special meaning in Python Engine(PE).
In Python version 3 we have around 36 keywords.

ex:- if, for, and, else, ...

out of 36 keywords there will be 3 keywords with first word capital.

ex:- True, False, None.

All other keywords i.e around 33 will have all lower case.

Rule 7: Identifiers

Identifier is a name given to an entity (i.e a class,

a variable, function, module etc)

Entity is a single occurence of something (objects).

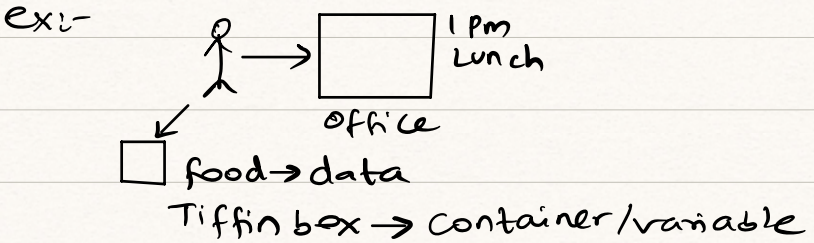
ex:- Object1 (O,) is given a name and this name is an identifier.

Rule 8: Variable

Simplified definition: It is a container where we will store data.

Variable is a subset of identifier.

ex:-



To identify variable / identifier we have to name them.

Naming Convention for variable / identifier:

These are set rules by which we will try to name the variable (or identifier)

1) The variable name should start with an alphabet (upper or lower case) or underscore (_).

2) Name should not start with a number (0-9) or special character (@, $, ₹, *, ␣ (space), ...)

3) Variable names can be a mixture of alphabets or numbers. (a-z, A-Z or 0-9)

4) Variable names should not use any special characters (@, $, ₹, #, *, ␣ ....)

5) It is a case-sensitive language.

ex:- X and x are different variables.

ex:- a2bc ✓
a@bc ✗
a_23 ✓
_a23 ✓
a␣23 ✗

If naming convention is not used we will get a syntax error.

* In python data is assigned to a variable using an Assignment operator.

variable = Data

* Python programming language is case-sensitive

* we can Assign a single value to multiple variables at the same time

ex:- a = b = c = 10

* we can Assign multiple values to multiple variables at the same time.

ex:- a, b, c = 10, 20, 30

Variable scope: The scope of the variable refers to the places access a variable

two types
- Local variable
- Global variable

Local variable: A local variable is a variable that is accessible inside a block of code only where it is declared. That means, if we declare a variable inside a function / method, the scope of the local variable is limited to the method only.

Global variable: A global variable is a variable that is defined outside of function / method (block of code). That is accessible anywhere in the code file.

Data: It is a piece of information.

There will be different Data types in Python.

Python Data types

four different types
- Numeric Data type
- String Data type
- Boolean Data type
- Collection Data type

Numeric Data types:
- Integer
- float
- Complex

Integer: Real numbers (range is $(-\infty, +\infty)$)

ex:- -1, -2, 1, 2, 0, ...

float: We have a decimal value $(-\infty, +\infty)$

ex:- -1.1, -2, 1.5, 10.5, ...

floating-point values can be represented using the exponential form, also called Scientific notation. The benefit is we can represent large values using less memory.

ex:- 1.22e4 (i.e 12200.0)

Complex: we have 2parts, real and imaginary.

$(a \pm bj)$

ex:- 2+5j

real part ↙ ↘ imaginary part

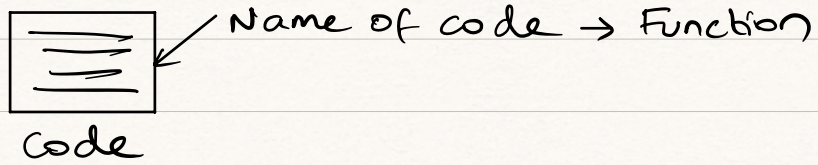
* imaginary part can only be 'J' or 'j' in python.
* 'j' should always be accompanied by real number.

ex:- 2+j ✗

2+1j ✓

* The real integer value can be in the form of either decimal, float, binary or hexadecimal. But the imaginary part should be represented using the decimal form only. we will get error if we by binary, hex or octal.

Function : It is a name given to a piece of code.



When we want to execute this piece of pre-existing code we call this function.

Functions

two types
- Inbuilt function
- user-defined function

Inbuilt functions :

Functions that are pre-existing in Python programming language.

1) **type( ) function**

here ( ) represents calling that function ex:- x( ).

type() function is an inbuilt function mainly used to know the data type of the data.

ex:- x=100

type(x)

↳ int (integer)

y=1+2j

type(y)

↳ complex

Type Conversion : It is a process by using which we can convert from one data type to another data type.

Type Conversion
- two types
  - Implicit Conversion
  - explicit Conversion

Implicit type Conversion :

This is the conversion of one data type in to another data type by python itself during the execution of the program.

ex:- 2.0 + 1 + 2j

= 3+2j (everything converted to Complex)

During implicit type conversion

first preference → Complex

second preference → float

third preference → Integer

* In implicit conversion we will not loose any data.

ex:- 1 + 2.2 = 3.2

↓ ↓ ↓

Int float float

Explicit type conversion :

When the user explicitly does the type conversion. For using explicit type conversion we take the help of constructors.

Constructors :-

Simple definition is it will construct something.

There are four types of constructors.

- integer constructor
- float constructor
- complex constructor
- string constructor (will be taught later)

integer constructor: This will construct integer from integer and float.

int( )

ex:- x=10

int(x)

↳ 10

y=10.2

int(y)

↳ 10 (we can loose data)

float constructor: This will construct float from float and integer.

float( )

ex:- x=100

float(x)

↳ 100.0

y= 20.53

float(y)

↳ 20.53

Complex constructor: This will construct Complex type from integer, float and complex.

Complex( )

ex:- x=100

y=100.53

complex(x)
↳ 100+0j

complex(y)
↳ 100+0j
↓
(we can loose data)

* In explicit conversion we can loose data.

Type error: This is an error when we are using a type conversion.

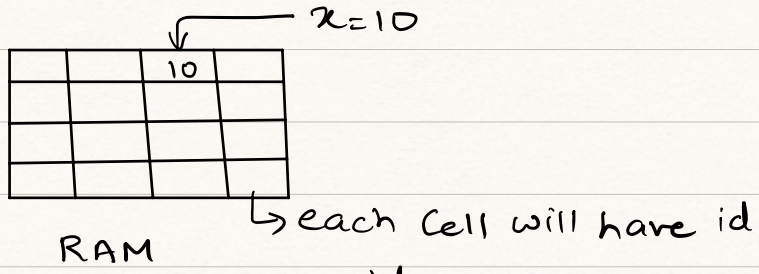
ex:- x = 1+2j

int(x)

↳ Type error as we cannot convert complex to integer.

Variable:

Actual definition: It is a name given to a memory location.



So here xxa10234 will be referred as x here.
So variable is a temporary name given to the memory location.

Inbuilt function

id( ): this function gives the id of the memory location.

ex:- x=10

id(x)

↳ xxa10234

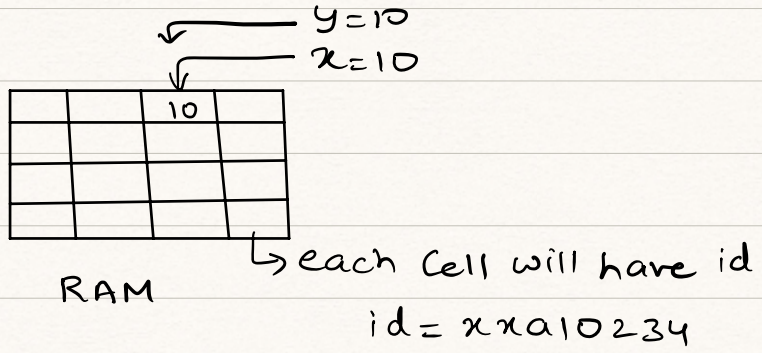
* Python will not waste memory

for ex:- x=10

y=10

So here y is not stored in seperate space. Instead it will refer to x.

So different variables having same data will have only one storage location.



so id(x) is same as id(y).