What is an Operating System ?

- It is a medium or an interface between the user and the hardware of the system
- It is a resource manager that provides resources to different applications to run on a machine
- Resources like input/outpur devices, memory unit, network, RAM , disk space

Example:- Windows, MacOS, linux, unix, ubuntu etc

What is a Programming language ?

- A programming langauge is a computer language which is used to communicate with the computer system
- A programming langauge is a set of some pre-defined words and characters that are used accordingly to some rules and these rules are called the syntax.
- Alphabets and words are considered as the character whereas the syntax can be called as the grammer of the progarmming langauge

Types of Programming

- High level language
- Low level language

High level language

```
- It is a human friendly computer language
- The code is written in simple language such as english
- It is easier to read and understand by a human being
- Example:- C, C++, Java, Python, Ruby, Julia, R etc
```

Low level langauge

```
- It is a machine friendly computer language
- The code written in this format is in binary digits also known as b
its (0,1)
- It is almost impossible for us human to understand it
- Examples - assembly langauge , Pascal , fortrain etc
```

Translators

- It is a software that converts the high level language into a machine level language and vice- versa

Types of Translators

```
- Compiler
- Interpreter
```

Compiler

```
- A compiler is a software that reads the whole high level code at on
ce and it converts the whole code in machine langauge at once
```

Interpreter

```
- A interpreter is a software that reads the high level code line by
line and it will convert the code into machine language line by line
```

Compiled Language and Interpreted Language

Compiled Languages

```
- Those high level languages that uses a compiler as their translator
are knonw as compiled languages
- Example- C, C++, Java
```

Interpreted Languages

```
- Those high level languages that uses an inerprter as their translat
or are known as interpreted languages
- Example- Python , R, ruby, PHP etc
```

Python --- Both a compiled language and Interpreted Language

- Python is both compiled and interpreted language, it has two main steps that are involved:-
  - Compilation --> when we write a code it will first compiled into a lower level language called as bytecode, this compilation is being done with the help of the compiler
  - Execution --> The bytecode is then converted into a machine language by the interpretr line by line

Source Code

```
- The code that we write as a developer is known as a source code
- The extension of the python source code us .py
```

Byte code

```
 - As soon as the code is executed (run), compilation of the source c
ode starts
 - This whole step is known as the compilation because at the end we
are going to have one code written in a language which still needs a
translator to convert it into a machine langauge
 - Here the code will be converted in one go
 - The extension for the byte code is .pyc
 - If during compiling, the compiler finds an error, bytecode generat
ion will not going to happen
```

Machine code

```
    - As soon as the we get the bytecode, the interpreter comes into
play
    - The interpreter converts the whole bytecode into a machine code
line by linbe
    - As soon as an error is detected, the execution will stop
    - In this case, the code is partially executed
```

In [2]:

```python
def add(a,b):
    return a+b

add(7,6)
```

Out[2]:

```
13
```

In [3]:

```python
import dis

dis.dis(add)
```

```
  1           0 RESUME                   0

  2           2 LOAD_FAST                0 (a)
              4 LOAD_FAST                1 (b)
              6 BINARY_OP                0 (+)
             10 RETURN_VALUE
```

In [ ]:

```
How machine code are interpreted
```

In [4]:

```python
a = 10 # 1010
b = 20 # 10100
c = a+b #operation will take place (the result will be in machine code on
print(c)

```

```
30
```

Type of Error

```
- Compiled time error
- Run time error
- Logical error (print the result but it will be wrong or undesirable
one)
```

Both the compile time error and the run time error will stop the execution of the code whereas the logical error will give you the undesirable result)

Compile Time Errors

- The error we get at the time of compilation (when the conversion of source code to byte code happens)
  - Syntax error
  - Indentation error

In [5]:
```python
for i in range(0,5)
    print(i)
```

```
  Cell In[5], line 1
    for i in range(0,5)
                       ^
SyntaxError: expected ':'
```

In [10]:
```python
def add(a,b):
return a+b
```

```
  Cell In[10], line 2
    return a+b
    ^
IndentationError: expected an indented block after function definition on li
ne 1
```

Run time error or Exceptions

- The run time error are those error that are syntactically correct but they create some issue while conversion of bytecode to machine code happens
  - Name error
  - Type error
  - Value error
  - index error
  - file not found error
  - stop iteration

- key error etc

In [11]:
```python
a = 10
b = 20
c = a+b
print(d)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[11], line 4
      2 b = 20
      3 c = a+b
----> 4 print(d)

NameError: name 'd' is not defined
```

In [12]:
```python
1+'mayank'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[12], line 1
----> 1 1+'mayank'

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

In [14]:
```python
while True:
    print('Mayank')
```

```
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
Mayank
```

In [15]:

```
import pandas as pd
a = pd.read_csv('titanic.csv')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[15], line 2
      1 import pandas as pd
----> 2 a = pd.read_csv('titanic.csv')

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:912, in read
_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols,
dtype, engine, converters, true_values, false_values, skipinitialspace, skip
rows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, ski
p_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parse
r, date_format, dayfirst, cache_dates, iterator, chunksize, compression, tho
usands, decimal, lineterminator, quotechar, quoting, doublequote, escapecha
r, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whitespa
ce, low_memory, memory_map, float_precision, storage_options, dtype_backend)
    899 kwds_defaults = _refine_defaults_read(
    900     dialect,
    901     delimiter,
   (...)
    908     dtype_backend=dtype_backend,
    909 )
    910 kwds.update(kwds_defaults)
--> 912 return _read(filepath_or_buffer, kwds)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:577, in _rea
d(filepath_or_buffer, kwds)
    574 _validate_names(kwds.get("names", None))
    576 # Create the parser.
--> 577 parser = TextFileReader(filepath_or_buffer, **kwds)
    579 if chunksize or iterator:
    580     return parser

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1407, in Tex
tFileReader.__init__(self, f, engine, **kwds)
   1404     self.options["has_index_names"] = kwds["has_index_names"]
   1406 self.handles: IOHandles | None = None
-> 1407 self._engine = self._make_engine(f, self.engine)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1661, in Tex
tFileReader._make_engine(self, f, engine)
   1659     if "b" not in mode:
   1660         mode += "b"
-> 1661 self.handles = get_handle(
   1662     f,
   1663     mode,
   1664     encoding=self.options.get("encoding", None),
   1665     compression=self.options.get("compression", None),
   1666     memory_map=self.options.get("memory_map", False),
   1667     is_text=is_text,
   1668     errors=self.options.get("encoding_errors", "strict"),
   1669     storage_options=self.options.get("storage_options", None),
   1670 )
   1671 assert self.handles is not None
   1672 f = self.handles.handle

File ~\anaconda3\Lib\site-packages\pandas\io\common.py:859, in get_handle(pa
th_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage
_options)
    854 elif isinstance(handle, str):
    855     # Check whether the filename is to be opened in binary mode.
    856     # Binary mode does not support 'encoding' and 'newline'.
    857     if ioargs.encoding and "b" not in ioargs.mode:
    858         # Encoding
--> 859         handle = open(
    860             handle,
    861             ioargs.mode,
    862             encoding=ioargs.encoding,
    863             errors=errors,
    864             newline="",
    865         )
    866     else:
    867         # Binary mode
    868         handle = open(handle, ioargs.mode)

FileNotFoundError: [Errno 2] No such file or directory: 'titanic.csv'
```

In [20]:

```python
a = 10
b = 0
c = a/b
print(c)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[20], line 3
      1 a = 10
      2 b = 0
----> 3 c = a/b
      4 print(c)

ZeroDivisionError: division by zero
```

In [21]:

```python
a = 'Mayank'
a[7]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[21], line 2
      1 a = 'Mayank'
----> 2 a[7]

IndexError: string index out of range
```

In [ ]:

```python

```