In [1]:

```
# System Version
import sys
sys.version  # science.physics. #commerce.economics
```

Out[1]:

```
'3.7.6 (default, Jan  8 2020, 13:42:34) \n[Clang 4.0.1 (tags/RELEASE_4
01/final)]'
```

In [2]:

```
print(sys.version)
```

```
3.7.6 (default, Jan  8 2020, 13:42:34)
[Clang 4.0.1 (tags/RELEASE_401/final)]
```

In [3]:

```
# modules of python
help('modules')
```

```
_threading_local    email               patsy               timeit
_tkinter            encodings           pdb                 tkinter
_tracemalloc        ensurepip           pep8                tlz
_uuid               entrypoints         pexpect             token
_warnings           enum                pickle              tokenize
_watchdog_fsevents  errno               pickleshare         toml
_weakref            et_xmlfile          pickletools         toolz
_weakrefset         fastcache           pip                 tornado
_xxtestfuzz         faulthandler        pipes               tqdm
_yaml               fcntl               pkg_resources       trace
abc                 filecmp             pkginfo             traceback
aem                 fileinput           pkgutil             tracemall
oc
aifc                filelock            platform            traitlets
alabaster           flake8              playsound           tty
anaconda_navigator  flask               plistlib            turtle
anaconda_project    fnmatch             pluggy              turtledem
o
antigravity         formatter           ply                 typed_ast
```

Tokens in Python

The smallest individual unit in a program is termed as **Token** or **lexical unit**

1. keywords
2. Identifier(Names)
3. Literals
4. Operators
5. Punctuators

In [4]:

```
# A sample Python Program
for a in range(1, 10):
    if a % 2 == 0:
        print(a)
```

```
2
4
6
8
```

Keywords

Keywords are nothing but some special names or reserved names that are already present in python.
`Predifined` words with special meaninigs.

In [5]:

```
import keyword
print(keyword.kwlist) #deep green  = keywords
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'br
eak', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'fi
nally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda',
'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'w
ith', 'yield']
```

In [6]:

```
help('keywords')
```

```
Here is a list of the Python keywords.  Enter any keyword to get more
help.

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

In [7]:

```
print(len(keyword.kwlist))
```

```
35
```

Identifier

Identifier are names given to diff part of your program.

In [9]:

```
# for a in range(45):
'a is the identifer here'
```

Out[9]:

```
'a is the identifer here'
```

Literals

Literals are fixed values / constants.

1. String ('anything under ("") or (''))
2. Numeric (12, 12.0, 12 + 1j)
3. Boolean (True, False, 1, 0)
4. Special Literals (None, Tuple)

Operators

Operators are tokens that triger some computation when applied to a variable

1. Arithmetic
2. Assignment
3. Bitwise
4. Shift
5. Identity
6. Relational
7. Logical
8. Membership

In [10]:

```
ext = ['pdf','docx']
```

In [11]:

```
'excel' in ext
```

Out[11]:

```
False
```

In [12]:

```
'excel' not in ext
```

Out[12]:

```
True
```

Punctuators

Punctuators are the symbols used in teh programming language.

"'#/\()[];:

Comments

Types = Single Line / MultiLine

In [13]:

```
# This is a Single Line Comment
```

In [14]:

```
# This is your
# Multi Line
# Comments

# I hope its clear
```

Variables = represent storage locations and store values

In [15]:

```
A =  20
B = 34.2
C = 'Machine.'
```

Checking Type

In [16]:

```
type(A), type(B), type(C)
```

Out[16]:

```
(int, float, str)
```

In [18]:

```
Integers = 'Whole Numbers (-∞ to ∞)'
```

In [19]:

```
Float = 'Real Numbers ...Decimal'
```

In [20]:

```
Complex = 'Real + Imag  a+bj'
```

In [21]:

```
D = 34 + 5j
E = 34j
```

In [22]:

```
1 type(D), type(E)
```

Out[22]:

```
(complex, complex)
```

In [23]:

```
1 D.real, D.imag
```

Out[23]:

```
(34.0, 5.0)
```

In [25]:

```
1 E.real, E.imag
```

Out[25]:

```
(0.0, 34.0)
```

In [ ]:

```
1
```