Python Features

- Highly Readable Language
- Clean Visual Layout
- Atomatic Memory Management
- Free & Open Source
- Support OOPs

Python Versions

- Python 2.
- Python 3.

In [ ]:

```
# python-2 -> print 'hello world'
# python-3 -> print("hello world")
```

Python Keywords

In [2]:

```
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 [5]:

```
import keyword
print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'fo
r', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'no
t', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

Python Operators

- Arithmatic Operators

- Relational Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

In [6]:

```
# Arithmatic Operators (+,-,*,/,%,//,**)
2+2
```

Out[6]:

```
4
```

In [7]:

```
8-2
```

Out[7]:

```
6
```

In [8]:

```
4/2
```

Out[8]:

```
2.0
```

In [9]:

```
4//2
```

Out[9]:

```
2
```

In [10]:

```
7//2
```

Out[10]:

```
3
```

In [11]:

```
7/2
```

Out[11]:

```
3.5
```

In [12]:

```
2**2
```

Out[12]:

```
4
```

In [13]:

```
2**8
```

Out[13]:

```
256
```

In [14]:

```
2.5+6.3
```

Out[14]:

```
8.8
```

In [15]:

```
# Relational Operators(==, !=,>,<,>=,<=)
2==5
```

Out[15]:

```
False
```

In [16]:

```
2>8
```

Out[16]:

```
False
```

In [17]:

```
8>2
```

Out[17]:

```
True
```

In [18]:

```
9<9
```

Out[18]:

```
False
```

In [19]:

```
5>=9
```

Out[19]:

```
False
```

In [20]:

```
9>=5
```

Out[20]:

```
True
```

In [21]:

```
3<=9
```

Out[21]:

```
True
```

In [22]:

```
8!=7
```

Out[22]:

```
True
```

In [23]:

```
#  Assignmant Operators(=,+=,-=,/=,*=,%=,//=,**=)
```

User Defind Varibles

- varibles don't start with Integers
- spaces are not allowed
- only accepted '_' reming special charcters not accepted
- can't match with Keywords

Python is a Casesensitive

- if upper & lower case consided as same that language is called as NON Case Sensitive
- if upper & lower case consided as same that language is called as Case Sensitive

In [24]:

```
A= 4
```

In [25]:

```
a=8
```

In [26]:

```
A
```

Out[26]:

```
4
```

In [27]:

```
a
```

Out[27]:

```
8
```

In [30]:

```
t1hing = 45
```

In [31]:

```
t1hing
```

Out[31]:

```
45
```

In [32]:

```
temp1=6
```

In [34]:

```
one_two=12
```

In [35]:

```
one_two
```

Out[35]:

```
12
```

In [37]:

```
f_w=3
```

In [38]:

```
class = 6
```

```
  File "<ipython-input-38-9d07ef8f6a6c>", line 1
    class = 6
          ^
SyntaxError: invalid syntax
```

In [39]:

```
True= 7
```

```
  File "<ipython-input-39-96b8f10cc94a>", line 1
    True= 7
    ^
SyntaxError: cannot assign to True
```

In [40]:

```
temp = 1
```

In [41]:

```
one = 5
```

In [42]:

```
one+10
```

Out[42]:

```
15
```

In [43]:

```
one
```

Out[43]:

```
5
```

In [44]:

```
one+=10
```

In [45]:

```
one
```

Out[45]:

```
15
```

In [46]:

```
one-=7
```

In [47]:

```
one
```

Out[47]:

```
8
```

In [48]:

```
t=9
```

In [49]:

```
t*=4
```

In [50]:

```
t
```

Out[50]:

```
36
```

In [51]:

```
q= 2
```

In [52]:

```
q
```

Out[52]:

```
2
```

In [53]:

```
q%=4
```

In [54]:

```
q
```

Out[54]:

```
2
```

In [55]:

```
q/=2
```

In [56]:

```
q
```

Out[56]:

```
1.0
```

In [57]:

```
q=2
```

In [58]:

```
q//=2
```

In [59]:

```
q
```

Out[59]:

```
1
```

In [60]:

```
# Bitwise Operators
5 & 4
```

Out[60]:

```
4
```

In [61]:

```
# Binary 5 ,4
  101
  100
#-----
  100
```

Out[61]:

```
100
```

In [62]:

```
4 | 5
```

Out[62]:

```
5
```

In [63]:

```
4 << 5
```

Out[63]:

```
128
```

In [ ]:

```
100
1000
10000
100000
1000000
10000000
```

In [68]:

```
int(0b10000000)
```

Out[68]:

```
128
```

In [69]:

```
4>>2
```

Out[69]:

```
1
```

In [70]:

```
100
010
001
```

. . .

In [71]:

```
4 ^ 2
```

Out[71]:

```
6
```

In [72]:

```
# Logical Operators (and,or, not)
4>5 and 6>9
```

Out[72]:

False

In [73]:

```
5>4 and 1>0
```

Out[73]:

True

In [74]:

```
5>4 and 8>9
```

Out[74]:

False

In [75]:

```
8>9 and 5>4
```

Out[75]:

False

In [76]:

```
False and True
```

Out[76]:

False

In [77]:

```
True or False
```

Out[77]:

True

In [78]:

```
False or False
```

Out[78]:

False

In [79]:

```
True or True
```

Out[79]:

```
True
```

In [82]:

```
4 and False
```

Out[82]:

```
False
```

In [81]:

```
5 and 6
```

Out[81]:

```
6
```

In [83]:

```
0 and 5
```

Out[83]:

```
0
```

In [84]:

```
not 4==8
```

Out[84]:

```
True
```

In [85]:

```
4==8
```

Out[85]:

```
False
```

In [86]:

```
not 4==8
```

Out[86]:

```
True
```

In [87]:

```
# Membership Operators (in, not in )
'g' in "programming"
```

Out[87]:

```
True
```

In [88]:

```
'z' in "school"
```

Out[88]:

```
False
```

In [89]:

```
'car' not in "Ramu have Pen"
```

Out[89]:

```
True
```

In [90]:

```
'Pen' not in "Ramu have Pen"
```

Out[90]:

```
False
```

In [91]:

```
# Identitiy Operators (is, is not) # checking Id's
x= ['apple',"banana"]
y =['apple','banana']
z=y
```

In [92]:

```
print(x is z)
```

```
False
```

In [93]:

```
print(x is y)
```

```
False
```

In [95]:

```
print(z is y)
```

```
True
```

In [96]:

```
id(x)
```

Out[96]:

```
2420108948800
```

In [97]:

```
id(y)
```

Out[97]:

```
2420108965696
```

In [98]:

```
id(z)
```

Out[98]:

```
2420108965696
```

In [99]:

```
x is not y
```

Out[99]:

```
True
```

In [100]:

```
z is not y
```

Out[100]:

```
False
```