

# Introduction to Python Programming

```
In [1]: # python has 33 keywords which has specific meaning and should not be used for anything
import keyword

# Get List of keywords
for i in keyword.kwlist:
    print (i)
```

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

## Data types

### Basic types

- int - Decimal, binary 0b, Octadecimal 0o, or Hexadecimal 0x
- float - Fractions
- complex - 3 + 2j
- bool - True or False (remember Keywords)
- string - use 'text' or "text"

- bytes - binary data ?

## Container types

- list
- tuple
- set
- dict

## User defined

- class (Refer last chapter)

## Vaiables and assignment

- variables should start with a letter (a - z, A - Z) or an underscore (\_)
- variable names cannot start with numbers (0 - 9)
- variables are case sensitive - Age and age are different variabes

```
In [7]: # All of the below are valid assignemnts
a = 25
b = 3.5
c = "cat"
d = e = f = 10
i, j = 56, 78
_user = 'ghost'
a9 = "Alphabet and Number combo"
camelCase = "Cool for long variable names"

print (a)
print (b)
print (c)
print (d)
print (e)
print (f)
print (i)
print (j)
print (_user)
print (a9)
print (camelCase)
print (CamelCase) # Note the error
```

```
25
3.5
cat
10
10
10
56
78
ghost
Alphabet and Number combo
Cool for long variable names
```

```
-----  
NameError                                Traceback (most recent call last)  
Cell In [7], line 22  
    20 print (a9)  
    21 print (camelCase)  
--> 22 print (CamelCase) # Note the error  
  
NameError: name 'CamelCase' is not defined
```

## Arthimatic operator

BEDMAS rule applies

```
In [17]: a = 5  
        b = 11  
        c = a + b  
        print (c)
```

16

```
In [18]: a / b
```

Out[18]: 0.45454545454545453

```
In [19]: b // a
```

Out[19]: 2

```
In [20]: b % a
```

Out[20]: 1

## Built-in functions

```
In [21]: print("NTPC")
```

NTPC

```
In [22]: max(1, 2)
```

Out[22]: 2

```
In [23]: min(12, 56)
```

Out[23]: 12

```
In [24]: abs(-9)
```

Out[24]: 9

```
In [27]: import math  
        math.pi
```

Out[27]: 3.141592653589793

```
In [28]: math.e
```

Out[28]: 2.718281828459045

```
In [29]: math.log(25)
```

Out[29]: 3.2188758248682006

```
In [30]: math.log10(25)
```

Out[30]: 1.3979400086720377

```
In [31]: math.factorial(16)
```

Out[31]: 20922789888000

```
In [33]: round(math.pi, 2)
```

Out[33]: 3.14

```
In [34]: import random
random.random()
```

Out[34]: 0.7512733205972268

```
In [36]: random.randint(10, 100)
```

Out[36]: 36

```
In [40]: # Generate levels between 20 and 30m elevation with
# 3 digit accuracy for a dummy site levelling package
random.randint(20, 29) + round(random.random(), 3)
```

Out[40]: 20.484

## Container types

- list
  - indexed collection
  - [1, 2, 3, 4, 5]
  - index starts with 0
- tuple
- set
- dict
  - Key: value pairs
  - { EDPM:"SNT", EDEN:"MS" }
  - { installedCapacity: 70, UnderConstruction: 20 }

```
In [49]: # Get values from a list using a loop or index
list = [1, 2, 3, 4, 5]

#Get a specific item
print("first item :", list[0])

# Get all items in a loop
for i in list:
```

```
    print (i)

# Get smallest and largest item
print("Smallest :", min(list))
print("Largest :", max(list))
print("Sum :", sum(list))
```

```
first item : 1
1
2
3
4
5
Smallest : 1
Largest : 5
Sum : 15
```

In [54]: *# Detail from a Dict can be accessed using the keys*  
**import** datetime

```
dict = {
    'project'      : 'Talcher',
    'zerodate'     : datetime.date(2022,9,27),
    'eic'          : "Deb"
}

print(dict['eic'])
```

Deb

In [ ]: