1.Base Conversion

1. bin()-convert any base to Binary.
2. oct()-convert any base to Octal.
3. Hex()-convert any base to Hexadecimal.

In [3]:
```
print(bin(15))
print(bin(0o11))
print(bin(0x10))
print(hex(0b110011))
print(oct(0xABCD))
```
```
0b1111
0b1001
0b10000
0x33
0o125715
```

2.float()

In [4]:
```
x=1.325
y=20.125
a=52e2
print(type(x))
print(type(y))
print(type(a))
```
```
<class 'float'>
<class 'float'>
<class 'float'>
```

3.Sequence Type

1. List()
2. Tuple()

Overread

allow duplicate values

Indexed

list()---changeable tuple()--unchangeable

In [8]:
```
l=[1,2,3,4,3,5]
print(l)
l[0]=34
print(l)
```
```
[1, 2, 3, 4, 3, 5]
[34, 2, 3, 4, 3, 5]
```

In [7]:
```
t=(12,3,38,5,25,6,7)
print(t)
print(t[0])
t[0]=6
print(t)
```
```
(12, 3, 38, 5, 25, 6, 7)
12
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-7-8321640dfbf1> in <module>
      2 print(t)
      3 print(t[0])
----> 4 t[0]=6
      5 print(t)

TypeError: 'tuple' object does not support item assignment
```

4.Range

--In Range the number will be excluded.

In [3]:
```
for i in range(10):
    print(i,end=" ")
print("----------")
for i in range(1,10):
    print(i,end=" ")
print("----------")
for i in range(1,10,3):
    print(i)
```
```
0 1 2 3 4 5 6 7 8 9 ----------
1 2 3 4 5 6 7 8 9 ----------
1
4
7
```

5.Mapping Type

1. dict --Key-Value pairs. --overread. --Changeable. --Does not allow duplicates.

In [12]:
```
d={10:"Aman",20:"Aryan",30:"Dhairya"}
print(d)
print(d[10])
print(d[20])
```
```
{10: 'Aman', 20: 'Aryan', 30: 'Dhairya'}
Aman
Aryan
```

In [15]:
```
d={10:"Aman",20:"Aryan",30:"Dhairya",10:"Jenil"}
print(d)
print(d[10])
print(d[20])
```
```
{10: 'Jenil', 20: 'Aryan', 30: 'Dhairya'}
Jenil
Aryan
```

6.Set Type

--Unoderd,Unindexed

In [1]:
```
s={"Apple","Banana","Orange","Cherry"}
print(s)
```
```
{'Apple', 'Cherry', 'Banana', 'Orange'}
```

In [2]:
```
s={"Apple","Apple","Orange","Cherry"}
print(s)
```
```
{'Apple', 'Cherry', 'Orange'}
```

7.Boolean Type

In [4]:
```
print(20>8)
print(20==9)
print(45<2)
print(bool("abd"))
print(bool(" "))
print(bool(123))
print(bool(0))
```
```
True
False
False
True
True
True
False
```

8.Reserved keywords

True, False, None, and, break, assert, or, not, is, if, elif, else, while, for, continue, return, in, yield, try, expect, finaly, raise, import, from, as, class, def, pass, global, nonlocal, del, with

9.Variables

In [5]:
```
a,b,c="Apple","banana","cherry"
print(type(a))
print(type(b))
print(type(c))
```
```
<class 'str'>
<class 'str'>
<class 'str'>
```

In [8]:
```
a="Apple","banana","cherry"
print(a)
print(type(a))
```
```
('Apple', 'banana', 'cherry')
<class 'tuple'>
```

10.Global variable vs Local variable

1. A variable which is defined inside the function considered as local variable.
2. A variable which is defined outside the function considered as Global variable.

In [9]:
```
a="python"
def test():
    global a
    a="java"
    print(a)
test()
print(a)
```
```
java
java
```

In [10]:
```
a="python"
def test():
    #global a
    a="java"
    print(a)
test()
print(a)
```
```
java
python
```

11.Comments

1. one line comment="# ..."
2. multiple line comment=" '''.....''' "

In [11]:
```
# this is program
s=10
'''java
python
css
js'''
print(s)
```
```
10
```

12.Reading input from user.

1. input()

In [12]:
```
# to add string
a=input("Enter Data for a:")
print(a)
print(type(a))
```
```
Enter Data for a:12
12
<class 'str'>
```

In [13]:
```
a=int(input("Enter Data for a:"))
b=int(input("Enter Data for b:"))
print(a+b)
print(type(a+b))
```
```
Enter Data for a:23
Enter Data for b:25
48
<class 'int'>
```

13.Type Casting

1. int()
2. float()
3. bool()
4. str()

In [26]:
```
print(int(123.654789))
print(int(True))
print(int(False))
print(int("10"))
```
```
123
1
0
10
```

In [16]:
```
print(int("52.102"))
print(int("gdeje"))
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-16-6ccc2247fb43> in <module>
----> 1 print(int("52.102"))
      2 print(int("gdeje"))

ValueError: invalid literal for int() with base 10: '52.102'
```

In [20]:
```
print(int("0B1111"))
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-20-d69b2a855e10> in <module>
----> 1 print(int("0B1111"))

ValueError: invalid literal for int() with base 10: '0B1111'
```

In [21]:
```
print(int(0B110101))
```
```
53
```

In [24]:
```
print(float(10))
print(float(True))
print(float(False))
print(float("10"))
print(float("100.20"))
```
```
10.0
1.0
0.0
10.0
100.2
```

In [25]:
```
print(float("ten"))
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-25-6628e595e84a> in <module>
----> 1 print(float("ten"))

ValueError: could not convert string to float: 'ten'
```

In [27]:
```
print(float("0B1010"))
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-27-9a98d2baa401> in <module>
----> 1 print(float("0B1010"))

ValueError: could not convert string to float: '0B1010'
```

In [28]:
```
print(float(0B1010))
```
```
10.0
```

In [29]:
```
print(bool(1))
print(bool(0))
print(bool(1.0))
print(bool(0.0))
print(bool("True"))
print(bool("False"))
```
```
True
False
True
False
True
True
```

In [30]:
```
print(str(10))
print(str(10.5))
print(str(True))
```
```
10
10.5
True
```

In [31]:
```
a=[1,2,3,5,6,9]
print(type(a))
print(a)
b=tuple(a)
print(type(b))
print(b)
```
```
<class 'list'>
[1, 2, 3, 5, 6, 9]
<class 'tuple'>
(1, 2, 3, 5, 6, 9)
```

14.Python Operators

1. Arithmetic Operators
   A. "+" --> Addition(str--> Concatation)
   B. "-" --> Subtraction
   C. "*" --> Multiplication(str--> String Multiplication or repetation)
   D. "/" --> Division(Answer will be in float by default)
   E. "//" --> Floor Division(Integer of Division)
   F. "%" --> Modulus
   G. "**" --> Exponent or power