PYTHON DATATYPES

BUILT IN DATA TYPES

Variables can store data of different types, and different types can do different things.

Python has the following data types built-in by default, in these categories:

In [1]:
```
'''Text Type:   str
Numeric Types:  int, float, complex
Sequence Types: list, tuple, range
Mapping Type:   dict
Set Types:      set, frozenset
Boolean Type:   bool
Binary Types:   bytes, bytearray, memoryview
None Type:      NoneType'''
```

Out[1]:
```
'Text Type:\tstr\nNumeric Types:\tint, float, complex\nSequence Types:\tlist, tuple, range\nMapping Type:\tdict\nSet Types:\tset, frozenset\nBoolean Type:\tbool\nBinary Types:\tbytes, bytearray, memoryview\nNone Type:\tNoneType'
```

In [2]:
```
x=10.0
print(type(x))
```

```
<class 'float'>
```

In [3]:
```
xy=20
print(type(xy))
```

```
<class 'int'>
```

In [4]:
```
y="umesh"
print(type(y))
```

```
<class 'str'>
```

In [5]:
```
z=True
print(type(z))
```

```
<class 'bool'>
```

In [7]:
```
l=["umesh","ramesh","chinni"]
print(type(l))
```

```
<class 'list'>
```

In [ ]:
```
# t=("umesh","ramesh","chinni")
print(type(t))
```

In [14]:
```
s={"apple","bannana","cherry"}
print(type(s))
```

```
<class 'set'>
```

In [15]:
```
d={"name" : "umesh","age":20}
print(type(d))
```

```
<class 'dict'>
```

In [9]:
```
c=2+3j
print(type(c))
```

```
<class 'complex'>
```

In [13]:
```
x=range(6)
print(x)
```

```
range(0, 6)
```

Python Numbers

THERE ARE THREE NUMERIC TYPES IN PYTHON 1.INTEGER 2.FLOAT 3.COMPLEX

In [16]:
```
x=1
y=1.3
z=3+2j
```

In [18]:
```
print(type(x))
```

```
<class 'int'>
```

In [19]:
```
print(type(y))
```

```
<class 'float'>
```

In [20]:
```
print(type(z))
```

```
<class 'complex'>
```

integer

Int, or integer, is a whole number, positive or negative, without decimals, of unlimited length.

In [21]:
```
x = 1
y = 35656222554887711
z = -3255522

print(type(x))
print(type(y))
print(type(z))
```

```
<class 'int'>
<class 'int'>
<class 'int'>
```

FLOAT

Float, or "floating point number" is a number, positive or negative, containing one or more decimals.

In [22]:
```
x = 1.10
y = 1.0
z = -35.59

print(type(x))
print(type(y))
print(type(z))
```

```
<class 'float'>
<class 'float'>
<class 'float'>
```

COMPLEX

Complex numbers are written with a "j" as the imaginary part:

In [23]:
```
x = 3+5j
y = 5j
z = -5j

print(type(x))
print(type(y))
print(type(z))
```

```
<class 'complex'>
<class 'complex'>
<class 'complex'>
```

TYPE CONVERSION

WE CAN CONVERT ONE DATATYPE TO ANOTHER DATATYPE BY USING TYPE CONVERSION

Convert from one type to another:

In [25]:
```
x=1
y=2.4
z=2+3j
```

In [28]:
```
#converting from int to float:
a=float(x)
print(a)
print(type(a))
```

```
1.0
<class 'float'>
```

In [29]:
```
#converting float in to int:
b=int(y)
print(b)
print(type(b))
```

```
2
<class 'int'>
```

In [31]:
```
#converting from int to complex
c=complex(x)
print(c)
print(type(c))
```

```
(1+0j)
<class 'complex'>
```

RANDOM NUMBER

Python does not have a random() function to make a random number, but Python has a built-in module called random that can be used to make random numbers:

In [33]:
```
#EXAMPLE
```

In [41]:
```
import random
print(random.randrange(0,20))
```

```
7
```

Python Casting

Specify a Variable Type There may be times when you want to specify a type on to a variable. This can be done with casting. Python is an object-orientated language, and as such it uses classes to define data types, including its primitive types.

Casting in python is therefore done using constructor functions:

- int() - constructs an integer number from an integer literal, a float literal (by removing all decimals), or a string literal (providing the string represents a whole number)
- float() - constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer)
- str() - constructs a string from a wide variety of data types, including strings, integer literals and float literals

In [43]:
```
#EXAMPLE
#Integers:

x = int(1)   # x will be 1
y = int(2.8) # y will be 2
z = int("3") # z will be 3
```

In [44]:
```
print(x,y,z)
```

```
1 2 3
```

In [45]:
```
#Example
#Floats:

x = float(1)     # x will be 1.0
y = float(2.8)   # y will be 2.8
z = float("3")   # z will be 3.0
w = float("4.2") # w will be 4.2
```

In [46]:
```
print(x,y,z)
```

```
1.0 2.8 3.0
```

In [47]:
```
#Example
#Strings:

x = str("s1") # x will be 's1'
y = str(2)    # y will be '2'
z = str(3.0)  # z will be '3.0'
```

In [48]:
```
print(x,y,z)
```

```
s1 2 3.0
```

In [ ]: