|  |  |  |  |
| --- | --- | --- | --- |
| DATATYPE | DESCRIPTION | IS IMMUTABLE? | EXAMPLE |
| Int | We can use to Represent the whole / integral  Numbers. | Immutable | >>> a=10  >>> type(a)  <class ‘int’> |
| Float | We can use to represent  Decimal / Floating  Point numbers. | Immutable | >>> b=10.5  >>> type(b)  < class ‘float’> |
| Complex | We can use to represent  The complex numbers. | Immutable | >>> c=10+5j  >>> type(c) |
| Bool | We can use to represent the logical values ( Only allowed values are True and False ) | Immutable | >>> flag = True  >>> flag = Flase  >>> type (flag)  <class’bool’> |
| Str | To represent sequence  Of Characters. | Immutable | >>> s=’shivansh’  >>> type(s)  <class ‘str’>  >>>s=”Shivansh”  >>> s=’’’not fikarr ,No fear. When Shivansh Gupta  Is here ‘’’  >>>type(s)  <class ‘str’> |
| Bytes | To represent a sequence of BYTES  Values from 0-255 | Immutable | >>> list = [1,2,3,4]  >>> ba = bytearray(list)  >>> type(ba)  <class ‘bytearray’> |
| Bytearray | To represent a sequence of BYTE  Value from 0-255 | Mutable | >>> list=[10,20,30,40]  >>>ba=bytearray(list)  >>> type(ba)  <class ‘bytearray’> |
| Range | To represent a range of  Values | Immutable | >>> r=range(10)  >>> r1 = range(0,10)  >>> r2 = range(0,10,2) |

|  |  |  |  |
| --- | --- | --- | --- |
| list | To represent an ordered collection of  Objects. | Mutable | >>> l=[10,20,30,40,50,60]  >>> type(l)  <class ‘list’> |
| Tuple | To represent an ordered collections of objects. | Immutable | >>> t=(1,2,3,4,5)  >>> type(t)  <class ‘tuple’> |
| Set | To represents an unordered collection  of UNIQUE Objects. | Mutable | >>> s=(10,20,30,40,50)  >>> type(s)  <class ’set’> |
| Frozement | To represent an unordered collection  Of unique Objects. | Immutable | >>> s=( 11,2,3,’Shiva’,100,’Gupta’)  >>> fs=frozenset(s)  >>> type(fs)  <class ‘ frozensets’> |
| Dict | To represents a group of key values  Pairs. | Mutable | >>>  d=(101:’shiva’,102:’gupta’,103:’ramu’)  >>> type(d)  <class ‘dict’> |