|  |  |  |  |
| --- | --- | --- | --- |
| Datatypes | description | Mutable/immutable | example |
| int | We can use to represent the whole/integral numbers | immutable | a=123  Print(type(a))  <class int> |
| float | We can use to represent the decimal/floating point numbers | immutable | A=10.5  Print(type(a)  <class float> |
| complex | We can use to represent the complex numbers | immutable | a= 10+5j  Print(type(a))  <class complex> |
| bool | We can use to represent the logical values (Only allowed values are True and False) | immutable | flag=True flag=False type(flag)  <class ‘bool’> |

**Task8 - Make a table of all the In-built data structures and Point out the differences one by one**

|  |  |  |  |
| --- | --- | --- | --- |
| byte | To represent a sequence of byte values from 0-255 | immutable | list=[1,2,3,4] b=bytes(list)  type(b) |
| bytearray | To represent a sequence of byte values from 0-255 | mutable | list=[1,2,3,4] ba=bytearray(list) type(ba) |
| range | To represent a range of values | Immutable | r=range(10) r1=range(0,10) r2=range(0,10,2) |
| string | To represent sequence of Characters | Immutable | s=‘vamsi'  type(s)  s=“sai is good"  type(s) |

|  |  |  |  |
| --- | --- | --- | --- |
| list | To represent an ordered collection of objects | Mutable | l=[10,11,12,13,14] type(l) |
| tuple | To represent an ordered collections of objects | immutable | t=(1,2,3,4,5)  type(t) |
| set | To represent an unordered collection of unique objects | Mutable | s={1,2,3,4,5}  type(s) |
| dict | To represent a group of key value pairs | Mutable | d = {1:‘a', 2:‘b', 3:‘c'} type(d) |

Note:

* **Immutable Objects :**  In simple words, an immutable object can’t be changed after it is created.

**Ex:**

|  |
| --- |
| # Python code to test that  # tuples are immutable    tuple1 = (0, 1, 2, 3)  tuple1[0] = 4  print(tuple1) |

Error :

traceback (most recent call last):

File "e0eaddff843a8695575daec34506f126.py", line 3, in

tuple1[0]=4

TypeError: 'tuple' object does not support item assignment

* **Mutable Objects :**These are of type [**list**](https://www.geeksforgeeks.org/python-list/)**,**[**dict**](https://www.geeksforgeeks.org/python-dictionary/)**,**[**set**](https://www.geeksforgeeks.org/sets-in-python/). Custom classes are generally mutable.

|  |
| --- |
| # Python code to test that  # lists are mutable  color = ["red", "blue", "green"]  print(color)    color[0] = "pink"  color[-1] = "orange"  print(color) |

Output:

['red', 'blue', 'green']

['pink', 'blue', 'orange']