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

|  |  |
| --- | --- |
| 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 |

# Text Type  
print("----------------------------")  
txtType1 = 'Manoj Veroni'  
txtType2 = "Manoj Verma"  
print(txtType1, type(txtType1))  
print(txtType2, type(txtType2))  
  
# Numeric Type - int, float, complex  
print("----------------------------")  
num1 = 6  
num2 = 6.2  
num3 = 1j  
print(num1, type(num1))  
print(num2, type(num2))  
print(num3, type(num3))  
  
# Sequence Type - List, Tuple and Range  
# Python List  
print("----------------------------")  
list1 = ["apple", "banana", "cherry"]  
print(list1, type(list1))  
  
# Python Tuple  
print("----------------------------")  
tuple1 = ("apple", "banana", "cherry")  
print(tuple1, type(tuple1))  
  
# Python Range  
print("----------------------------")  
ran = range(6)  
print(ran, type(ran))  
  
# Mapping Type  
print("----------------------------")  
dictMap = {"name": "Johnna", "age": 26}  
print(dictMap, type(dictMap))  
  
# Set Type - set , frozenset  
  
# Set Type  
print("----------------------------")  
set1 = {"Thorivakkam", "Thanjvaur", "Delhi"}  
print(set1, type(set1))  
  
# FrozenSet Type  
print("----------------------------")  
fset1 = frozenset({"Thorivakkam", "Thanjvaur", "Delhi"})  
print(fset1, type(fset1))  
  
# Boolean Type  
print("----------------------------")  
boolType = True  
print(boolType, type(boolType))  
  
# None Type  
print("----------------------------")  
noneType = None  
print(noneType, type(noneType))  
  
# Binary Type  
print("----------------------------")  
byteType = b"Hello"  
bytearrayType = bytearray(5)  
bytearrayTypeMemoryView = memoryview(bytes(5))  
print(byteType, type(byteType))  
print(bytearrayType, type(bytearrayType))  
print(bytearrayTypeMemoryView, type(bytearrayTypeMemoryView))