|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | List | Dictionary | Tuple | Set |
|  | lists are written with square brackets  List1 =[1,True,”Spain”,89.80 ] | dictionaries are written with curly brackets and they have keys and values separated by commas  Dict1 = {“OS”:”Windows”,”Model”:”OnePlus”} | Tuples are written with round brackets.  Tup1 = (1,3,”hello”,True,5,6)  to create a tuple with one element make use of comma(,) after the value | sets are written with curly brackets.  Set1 = {1,2,6,”demo”,True} |
| Stores different data types | Yes | Yes | Yes | Yes |
| Duplicate values | allowed | Keys should be unique | Allowed | Not allowed |
| order | retained | Not retained | retained | Not retained |
| mutable | Yes | Yes | No | Yes, items can be added or removed but not updated |
| Indexing and slicing | Yes, indexing starts with zero | Can access items using keys  Proper indexing and slicing not possible | Possible | Not possible |
| adding elements | 1 *.append(val)*2. *insert(pos,element)* *3 .copy()*  *4 .extend()* | dict1[key] = value  Updates the value if key already exists, if not creates a new key and value | Adding can be done by converting tuple into list and again changing it to tuple | 1 .add(value)  2 .update(list) - used to add multiple items at a time |
| Removing elements | .pop(index) | 1. pop(key)  2. popitem() removes the last inserted item  3. del keyword  del dict1['year']  4. clear() method - deletes the dictionary | Removing can be done by converting tuple into list and again changing it to tuple | 1.remove(item) - if item doesnot exists throws an error  2.Discard(item) - similar as remove but doesnot throw an error if item doesnot exist 3.pop() - as set is unordered we cannot know which item is removed. |