**CSF 2113 Lab 5.2: Containers (Tuple & Dictionaries)**

1. **Using Tuples in Python**

**Tuples are a collection of data items. They may be of different types. Tuples are immutable (like strings). Python optionally uses brackets () to denote tuples**

**We could have also used () for the above tuple**

**If we have only one item, we need to use a comma to indicate it's a tuple: e.g. (“Bat”,)**

1. **Creation:** Create a touple of following information about a student. Name the tuple std.

“Fatima”, 3.4, 78,”Sharjah”



|  |
| --- |
| std = ("Fatima", 3.4 , 78 , "Sharjah")  print(std)  ('Fatima', 3.4, 78, 'Sharjah') |

1. Create a tuple where one element is a list and rest are numbers:



|  |
| --- |
| tuple = (4, 8, 9, ["Ali", 2.3])  print(tuple)  (4, 8, 9, ['Ali', 2.3]) |

1. **Indexing**: Tuple elements can be accessed by index: try out following element in above created lists:

tup[0]

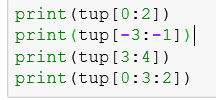
tup[-1]

tup[3]

tup[-4]

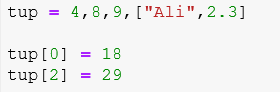
|  |
| --- |
| tuple = (4, 8, 9, ["Ali", 2.3])  print(tuple)  print(tuple[0])  print(tuple[-1])  print(tuple[3])  print(tuple[-4])  (4, 8, 9, ['Ali', 2.3])  4  ['Ali', 2.3]  ['Ali', 2.3]  4 |

1. **Slicing:** All slice operations return a new tuple containing the requested elements. This means that the following slice returns a new (shallow) copy of the tuple. Perform following slicing operations on above created tuple:



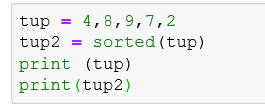
|  |
| --- |
| tuple = (4, 8, 9, ["Ali", 2.3])  print(tuple)  print(tuple[0:2])  print(tuple[-3:-1])  print(tuple[3:4])  print(tuple[0:3:2])  (4, 8, 9, ['Ali', 2.3])  (4, 8)  (8, 9)  (['Ali', 2.3],)  (4, 9) |

1. **Mutable**: Tuples are immutable. We cannot change the value of an index. Try out following.



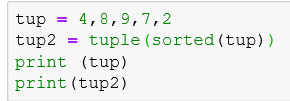
|  |
| --- |
| tup = 4,8,9,["Ali",2.3]  tup[0]=18  tup[2]=29  **---------------------------------------------------------------------------**  **TypeError** Traceback (most recent call last)  **<ipython-input-14-ce7b28534c0f>** in <module>  1 tup **=** **4,8,9,["Ali",2.3]**  2  **----> 3** tup**[0]=18**  4 tup**[2]=29**  **TypeError**: 'tuple' object does not support item assignment |

1. **Sorting a Tuple**: As tuple are immutable sorting a tuple is not possible however we can sort by using sorted function which not change the tuple will rather return a sorted list of elements of tuple.



|  |
| --- |
| tup = 4,8,9,7,2  tup2 = sorted(tup)  print(tup)  print(tup2)  (4, 8, 9, 7, 2)  [2, 4, 7, 8, 9] |

1. **Sorting a Tuple**: We can use this sorted list to create another tuple.



|  |
| --- |
| tup= 4,8,9,7,2  tup2= tuple(sorted(tup))  print(tup)  print(tup2)  (4, 8, 9, 7, 2)  (2, 4, 7, 8, 9) |

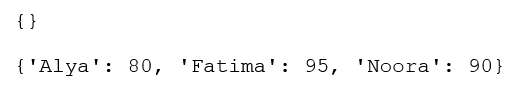
1. **Other functions of tuple**: We can use variety of functions with tuple.

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **purpose** | **code** | **output** |
| length | Total length of tuple |  | 5 |
| concatenation | Concatenate two or more tuples |  | (4, 8, 9, 'sharjah', 'Dubai') |
| Repetition | Repeat the sequence of element |  | (4, 8, 9, 4, 8, 9, 4, 8, 9) |
| Membership | Check the element is present in tuple |  | True  False |
| Max | Find maximum value in tuple |  | 9 |
| Min | Find minimum value in tuple |  | 4 |

1. **Using Dictionaries in Python**

**A dictionary is like a list, but more general. In a list, the indices have to be integers; in a dictionary they can be (almost) any type. Keys must be *unique* within a dictionary: No *duplicates.* Simply put, a dictionary is a list of key-value pairs.**

1. Create two dictionary variables: One an empty dictionary and one with student’s marks as shown in the image below. Than Display the contents of dictionaries.



|  |
| --- |
| empty = {}  students = {"Alya": 80, "Fatima": 95, "Noora":90}  print(empty)  print(students)  {}  {'Alya': 80, 'Fatima': 95, 'Noora': 90} |

1. Write down the output of the following python code segment.

|  |
| --- |
|  |
| 80  95  **---------------------------------------------------------------------------**  **KeyError** Traceback (most recent call last)  **<ipython-input-19-1fafc9861bc2>** in <module>  2 display**(**students\_marks**["Alya"])**  3 display**(**students\_marks**["Fatima"])**  **----> 4** display**(**students\_marks**["Noura"])**  **KeyError**: 'Noura' |

1. Create a dictionary with three items as per given screenshot.



Display following:

* List of all items in the dictionary “student\_marks”
* List of all keys in the dictionary “student\_marks”
* List of all the values in the dictionary “student\_marks”

|  |
| --- |
| {'Alya': 80, 'Fatima': 95, 'Noora': 90}  dict\_keys(['Alya', 'Fatima', 'Noora'])  dict\_values([80, 95, 90]) |

1. Write down the output of the following code segment.

|  |
| --- |
|  |
| True  True  True  False |

1. Write down the output of the following code segment.

|  |
| --- |
|  |
| 80 95 90 |

1. Write down the output of the following code segment.

|  |
| --- |
|  |
| {'Alya': 80, 'Fatima': 95, 'Noora': 90} 3  {'Alya': 85, 'Fatima': 95, 'Noora': 90} 3  {'Alya': 85, 'Fatima': 95, 'Noora': 75} 3 |

1. Write down the output of the following code segment.

|  |
| --- |
|  |
| {'Alya': 80, 'Fatima': 95, 'Noora': 90} 3  {'Alya': 67, 'Fatima': 95, 'Noora': 90} 3  {'Alya': 67, 'Fatima': 95, 'Noora': 90, 'Sara': 82} 4 |

**End of Lab**