* **Container:**
  + Container is an entity that contains multiple data items.
  + Also known as a collection or a compound data type
* **Container Data Types:**
  + **Lists**
  + **Tuples**
  + **Sets**
  + **Dictionaries**
* **Tuples**
  + Commonly used for storing dissimilar data (and that means it can store similar data as well).
  + A tuple is defined by writing Comma-separated elements within **( )**.
    - num = ( 18, 2 , 72, 26, 4, 74 )
    - names = (‘Darshit’, ‘Ragi’)
    - stud\_data = (‘Darshit’, 18, 2 , 72 )
  + Items in a tuple can be repeated. i.e. a tuple may contain duplicate items.
    - ages = (50, 48, 50)
    - Ages = 50, 48, 50 🡨 While initializing a tuple, we may drop ( ).
    - tpl = ( ) 🡨 empty tuple, valid
    - a = (11,) 🡨 Tuple with one item. After 11 a comma is necessary.
    - num = (10,) \* 5 🡨 stores (10, 10, 10, 10, 10)
    - num = (10) \* 5 🡨 Stores 50 and type of num would be <int>.
* **Accessing Tuple Elements**
  + print (num) 🡨 entire tuple can be printed by just using the name of the tuple.
  + print (names[0]) 🡨 ‘Darshit’ 🡨 Individual elements can also be accessed using

indices like we used earlier in strings and lists.

* + print (num[1:4]) 🡨 2, 72, 26 🡨 like strings and lists, tuples can also be sliced.
* **Looping in Tuples**
  + Using **for** and **while**
  + #using for loop #using enumerate() to keep track of index

for i in num: for index, a in enumerate(num):

print (i) print (index, a)

* + #using while loop

i = 0

while i < len(num):

print(num[i])

i = i + 1

* **Basic Tuple Operations**
  + **Tuples are immutable (unchangeable).** 
    - Name[0] = ‘Aashna’ 🡨 Error
    - Num[0:3] = (15 , 8, 6 ) 🡨 Error
  + **Since a tuple is immutable, operations like append, remove and insert do not work with a tuple.**
  + **One tuple can be concatenated (appended) at the end of another tuple.**
    - tpl1 = ( 15, 8 , 6)
    - tpl2 = tpl1 + (70, 80, 90)
  + **Two tuples can be merged to create a new tuple.**
    - tpl3 = tpl1 + tpl2
  + **Though a tuple itself is immutable, it can contain mutable objects like lists.**
    - t = ([1,2,3,4], [4,5], ’Drisha’)
  + **If a tuple contains a list, the list can be modified as list is a mutable object.**
    - t [1][1] = 45 🡨 5 will be replaced with 45 in above tuple.
  + **Conversion:** 
    - **A string/list/set can be converted into a tuple using the tuple() conversion function.**
      * l = tuple(‘PDPU’) 🡪 l(‘P’,’D’,’P’,’U’)
  + **Aliasing:** 
    - **On assigning one tuple to another, both refer to the same tuple. Changing one changes the other.**
    - **Also known as shallow copy or aliasing.**
      * tpl1 = ( 15, 8 , 6)
      * tpl2 = tpl1 🡨 doesn’t copy tuple. tpl2 refers to the same tuple as tpl1.
      * print (tpl1 is tpl2) 🡨 True
  + **Cloning:** 
    - **This involves copying contents of one tuple into another.**
    - **After copying both refer to different tuples, although both contain same values.**
    - **Changing one tuple doesn’t change another.**
    - **Also known as deep copy.**
      * tpl1 = ( 15, 8 , 6)
      * tpl2 = ( )
      * tpl2 = tpl2 + tpl1
      * print (tpl1 is tpl2) 🡨 False
  + **Searching:**
    - **An element can be searched in a tuple using the (in) membership operator.**
      * print (15 **in** tpl1) 🡨 True
  + **Identity:**
    - **Use (is) operator to check whether the two variables are referring to the same tuple.**
      * tpl1 = (10 , 20 , 30 , 40 , 50 )
      * tpl2 = (10 , 20 , 30 , 40 , 50 )
      * tpl3 = tpl1
      * print (tpl1 is tpl2, tpl1 is not tpl2, tpl1 is tpl3) 🡨 False, True, True
  + **Comparison:**
    - **We can compare contents of two tuples.**
    - **Comparison is done item by item till there is a mismatch.**
      * A = (1, 2, 3, 4)
      * B = (1, 2, 5)
      * print ( A < B) 🡨 True
  + **Emptiness:** 
    - **We can check if a tuple is empty using not operator.**
      * tpl = ( )
      * if not tpl:
        + print (“Empty tuple.”)
      * print (bool(tpl))
  + **Built-in functions on Tuples**
    - len(tpl) - max(tpl) - min(tpl)
    - sum(tpl) - any(tpl) - all(tpl)
    - sorted(tpl) - reversed(tpl)
    - Examples of sorted( ) and reversed( )
      * sorted ( ) and reversed ( ) return a **List** and not tuple.
        + tpl = (72, 60, 58 , 52, 54, 56 )
        + tpl1 = reversed(tpl)
        + print(tpl1) 🡨 [56, 54, 52, 58, 60, 72]
        + tpl2 = sorted(tpl)
        + print(tpl2) 🡨 [52, 54, 56, 58, 60, 72]
        + tpl3 = sorted(tpl, reverse = True)
        + print(tpl) 🡨 [72, 60, 58, 56, 54, 52]
  + **Tuple Methods**
    - **Any tuple is an object of type tuple.**
    - **Its methods can be accessed using the syntax tpl.method( )**
      * tpl = (72, 60, 58 , 52, 54, 56 )
      * ~~tpl.append(74)~~ ##########################################
      * ~~tpl.remove(58)~~ # #
      * ~~tpl.remove(50)~~ # Tuples are immutable, so append, remove, #
      * ~~tpl.pop()~~ # pop, delete, insert, not going to work on them. #
      * ~~tpl.pop(4)~~ # #
      * ~~tpl.insert(3,6)~~ ##########################################
      * tpl.count(72) # counts frequency of 72 in the tuple.
      * i = tpl.index(72) # returns index of item 72 in the tuple.
      * i = tpl.index(50) # returns valueError as 50 is not there in the tuple.
  + **Tuple Varieties**
    - **Nested Tuple is allowed. We can create a tuple of tuples.**
      * a = ( 1, 3, 5, 7, 9)
      * b = (2, 4, 6, 8, 10)
      * c = (a, b) 🡨 ((1, 3, 5, 7, 9), (2, 4, 6, 8, 10))
      * print( c(0) (0), c (1) (2) ) #0th element of 0th tuple and 2nd element of 1st tuple.
    - **A tuple may be embedded in another tuple**
      * a = ( 1, 3, 5, 7, 9)
      * b = (2, 4, a, 6, 8, 10)
      * print (b) 🡨 (2, 4, (1, 3, 5, 7, 9), 6, 8, 10)
      * **# unpack a string or tuple within a tuple using \* operator**
      * b = (2, 4, \*a, 6, 8, 10)
      * print (b) 🡨 (2, 4, 1, 3, 5, 7, 9, 6, 8, 10)
    - **It is possible to create a list of tuples, or a tuples of lists.**
      * lst = [ (‘Darshit’, 51, 18,2,72), (“Ragi”, 48, 26,4,74)]
      * tpl = ( [‘Darshit’, 51, 18,2,72], [“Ragi”, 48, 26,4,74])
    - **Sorting either a list of tuples or tuple of lists:**
      * print (sorted(lst)) 🡨 by default, sorted() sorts by first item in list/tuple i.e. name.
      * print (sorted(tpl)
      * **import operator**
      * print (sorted(lst, key = operator.itemgetter(1))) 🡨 will sort the data based on
      * print (sorted(tpl, key = operator.itemgetter(1))) 🡨 age.
      * print (sorted(tpl, key = operator.itemgetter(1), reverse = True))

**Few Examples of Tuple:**

1. **Pass a tuple to divmod() and obtain the quotient and the remainder.**

result = divmod(17, 3) Output:

print (result) (5, 2)

t = (17,3)

result = divmod(\*t) 🡨 Unpack t into two distinct value and then pass to divmod().

print (result) 🡨 (5, 2) , divmod() returns a tuple consisting of quotient and remainder.

1. **Unpack the tuple in 5 variables, each holding 1 value.**

tpl = (10 , 20 , 30 , 40 , 50)

a, b, c, d, e = tpl

1. **Unpack the tuple in 5 variables such that 1st value gets stored in 1st variable x and last value in y and all other values in between into disposable variables.**

x, \_, \_, \_, y = tpl

1. **Unpack the tuple in 5 variables such that 1st value gets stored in 1st variable x and last value in y and all other values in between into a single disposable variables.**

x, \*\_, y = tpl

1. **Unpack the tuple in 5 variables such that 1st value gets stored in 1st variable x and the second last value in y and all other values in between into a single disposable variables.**

x, \*\_, y , \_ = tpl

1. **Check whether a list element is tuple of not.**

lst = [ ‘a’, ‘b’,’c’,(‘d’), (‘e’) ]

for e in lst:

if isinstance(e, tuple):

….

* **Write following programs considering tuple in mind:**

1. A list contains names of boys and girls as its elements. Boys’ names are stored as tuples. Write a program to find out number of boys and girls in the list. (Hint: use isinstance(ele,tuple))
2. A list contains tuples containing roll no., name and age of student. Write a python program to create three lists separately for roll no., name and age
3. Suppose a date is represented as a tuple (d, m, y). Create two date tuples and find the number of days between the two dates.
4. Create a list of tuples containing a food item and its price. Sort the tuples in descending order by price.
5. Remove empty tuple(s) from the list of tuples.
6. Modify an element of a tuple.
7. Delete an element of a tuple.

What will be the output of the following code?

1. lst = [ (‘X’, ‘Y’, ‘Z’), 40, 50 , 60]

a = lst[0]

print (a)

1. X b) 0 c) (x, y, z) d) (‘x’, ‘y’, ‘z’)