mindX
Technology & Startup School

Session 6

List - Tuple

The “collection” data types

After having mastered basic data types such as int, float, string and bool, we can organize them into “collections” - that is data types that can hold multiple values

TABLE OF CONTENTS

mindX
Technology & Startup School

01

List in Python

What is a “list”?

- A type of collection which allows storing multiple values, either homogeneous or mixed
- Sort of like the “array” data type in other languages
- Initialize with the syntax **[**item1, item2,...**]** or the **list((**item1, item2,...**))** function
- Items start at index **0**

```
scores = [5, 6, 6.5, 10]
players = list(('john',
'anna', 'tom', 'kelly'))
```

Common list methods

list[position]

Access item by index, also re-assigning

len(list)

Get the item size of the list

item in list

Check if item exists in a list

in is a **Boolean operator!!**

Common list methods: Add element(s)

list.insert(item)

Add a new element at specific position

list.append(item)

Add **1 new item** to the end of the list

list.extend(item)

Add **several items** to the end of the list

Joining 2 lists

mindX
Technology & Startup School

The + operator

```
list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]

list3 = list1 + list2
print(list3)
```

extend() method

```
list1 = ["a", "b" , "c"]
list2 = [1, 2, 3]

list1.extend(list2)
print(list1)
```

Common list methods: Delete element(s)

del(list[index])

Delete element(s) by index

del is a **keyword!!**

list.remove(item)

Delete an element by value

list.pop(index)

Delete an element by index & **return** that elements value
<u>Default</u>: the last element in the list

list.clear()

Clear the whole list

Common list methods: Organize list

list.sort(item)

Sort the list in alphabetical order (**mutate** the list)

sorted(list)

Builds a new sorted list from an **interable** (**NOT mutate** the list)

list.reverse()

Reverse the original order of a list

sorted is a **built-in function!!**

for <item> in list:

Loop through the whole list, each item is represented by the **item** variable

for i in range(len(list)):

Loop through the whole list with the index, access the element with **list[i]**

Some note about mutability vs immutability

A list can add and remove values, which is why it’s called a **mutable** data type

Unlike int, float, string,... where if you want to change the value of the variable, you have to reassign it, the value itself can’t be changed, meaning it’s **immutable**

Copying a list

When assigning a list variable to a second variable, the second variable will hold a **reference** to the list, not the **copy** the list. Any modification to the second variable will reflect back the original list

To copy the list to another variable and preserve the original list, use the **copy()** method

TRY IT YOURSELF

[redacted] **Seeing the World:** Think of at least five places in the world you'd like to visit.

- Store the locations in a list. Make sure the list is not in alphabetical order.
- Print your list in its original order. Don't worry about printing the list neatly, just print it as a raw Python list.
- Use `sorted()` to print your list in alphabetical order without modifying the actual list.
- Show that your list is still in its original order by printing it.
- Use `sorted()` to print your list in reverse alphabetical order without changing the order of the original list.
- Show that your list is still in its original order by printing it again.
- Use `reverse()` to change the order of your list. Print the list to show that its order has changed.
- Use `reverse()` to change the order of your list again. Print the list to show it's back to its original order.
- Use `sort()` to change your list so it's stored in alphabetical order. Print the list to show that its order has been changed.
- Use `sort()` to change your list so it's stored in reverse alphabetical order. Print the list to show that its order has changed.

TRY IT YOURSELF

4-1. Pizzas: Think of at least three kinds of your favorite pizza. Store these pizza names in a list, and then use a `for` loop to print the name of each pizza.

- Modify your `for` loop to print a sentence using the name of the pizza instead of printing just the name of the pizza. For each pizza you should have one line of output containing a simple statement like *I like pepperoni pizza.*
- Add a line at the end of your program, outside the `for` loop, that states how much you like pizza. The output should consist of three or more lines about the kinds of pizza you like and then an additional sentence, such as *I really love pizza!*

4-2. Animals: Think of at least three different animals that have a common characteristic. Store the names of these animals in a list, and then use a `for` loop to print out the name of each animal.

- Modify your program to print a statement about each animal, such as *A dog would make a great pet.*
- Add a line at the end of your program stating what these animals have in common. You could print a sentence such as *Any of these animals would make a great pet!*

TRY IT YOURSELF

4-3. Counting to Twenty: Use a `for` loop to print the numbers from 1 to 20, inclusive.

4-4. One Million: Make a list of the numbers from one to one million, and then use a `for` loop to print the numbers. (If the output is taking too long, stop it by pressing CTRL-C or by closing the output window.)

4-5. Summing a Million: Make a list of the numbers from one to one million, and then use `min()` and `max()` to make sure your list actually starts at one and ends at one million. Also, use the `sum()` function to see how quickly Python can add a million numbers.

4-6. Odd Numbers: Use the third argument of the `range()` function to make a list of the odd numbers from 1 to 20. Use a `for` loop to print each number.

4-7. Threes: Make a list of the multiples of 3 from 3 to 30. Use a `for` loop to print the numbers in your list.

4-8. Cubes: A number raised to the third power is called a *cube*. For example, the cube of 2 is written as `2**3` in Python. Make a list of the first 10 cubes (that is, the cube of each integer from 1 through 10), and use a `for` loop to print out the value of each cube.

Documents

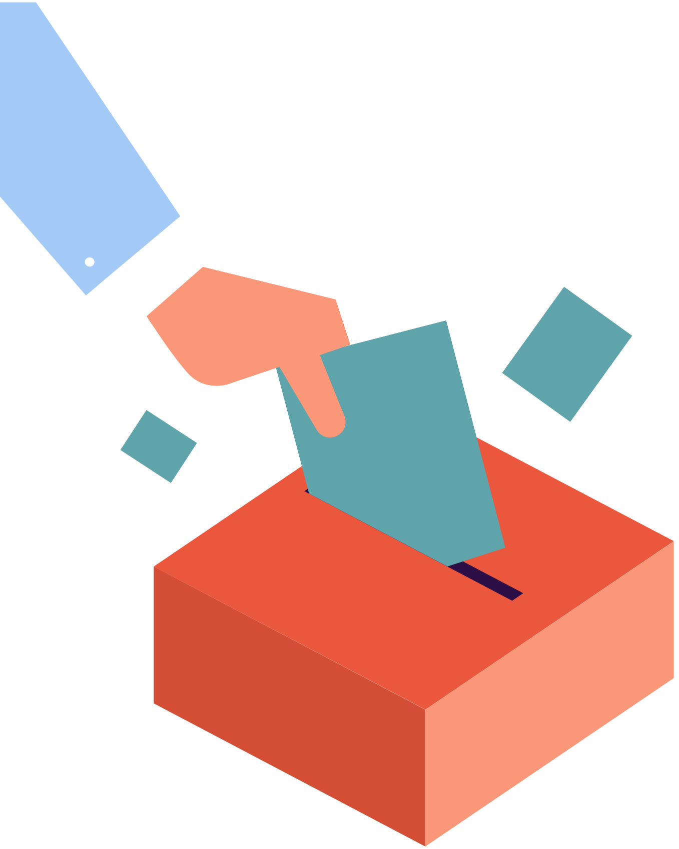
[5. Data Structures — Python 3.10.0 documentation]

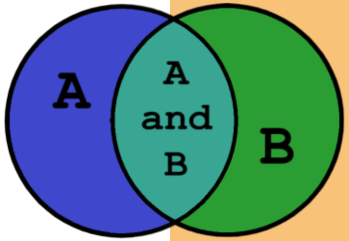
[Python Tutor - Visualize Python, Java, JavaScript, C, C++, Ruby code execution]

mindX
Technology & Startup School

02

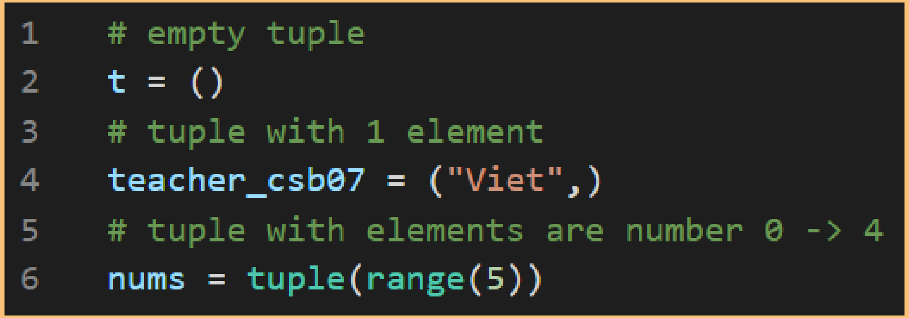
Tuple in Python





What is a “tuple”?

- Sort of like list, set allows storing multiple items, either homogeneous or mixed
- However, tuple does not allow changing items in it



```python
# empty tuple
t = ()
# tuple with 1 element
teacher_csb07 = ("Viet",)
# tuple with elements are number 0 -> 4
nums = tuple(range(5))
```

TRY IT YOURSELF

4-13. Buffet: A buffet-style restaurant offers only five basic foods. Think of five simple foods, and store them in a tuple.

- Use a `for` loop to print each food the restaurant offers.
- Try to modify one of the items, and make sure that Python rejects the change.
- The restaurant changes its menu, replacing two of the items with different foods. Add a block of code that rewrites the tuple, and then use a `for` loop to print each of the items on the revised menu.

THANKS!

See you in the next lesson!