

Lecture 2 - Prepared

January 23, 2023

1 Outline:

Data structure in Python:

- Lists:
- Define
- Simple operations
- Practice problem
- Tuple:
- Define
- Simple operations
- Practice problem

2 Lists

Lists are a collection of items in a specific order. They are mutable, which means that they can be modified after they are created.

```
[1]: # Creating a list
my_list = [1, 2, 3, 4, 5]
print(my_list) # [1, 2, 3, 4, 5]

# Adding items to a list
# Using the append() method
my_list.append(6)
print(my_list) # [1, 2, 3, 4, 5, 6]

# Using the extend() method
new_list = [7, 8, 9]
my_list.extend(new_list)
print(my_list) # [1, 2, 3, 4, 5, 6, 7, 8, 9]

# Using the insert() method
my_list.insert(3, 10)
print(my_list) # [1, 2, 3, 10, 4, 5, 6, 7, 8, 9]

# Removing items from a list
# Using the remove() method
```

```

my_list.remove(10)
print(my_list) # [1, 2, 3, 4, 5, 6, 7, 8, 9]

# Using the pop() method
my_list.pop()
print(my_list) # [1, 2, 3, 4, 5, 6, 7, 8]

# Sorting and reversing a list
# Using the sort() method
my_list.sort()
print(my_list) # [1, 2, 3, 4, 5, 6, 7, 8]

# Using the reverse() method
my_list.reverse()
print(my_list) # [8, 7, 6, 5, 4, 3, 2, 1]

# Accessing items in a list
# Using indexing
print(my_list[0]) # 8

# Using slicing
print(my_list[1:4]) # [7, 6, 5]

```

```

[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 10, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8]
[8, 7, 6, 5, 4, 3, 2, 1]
8
[7, 6, 5]

```

3 Practice Problems

Problem 1: 1. Create a list of your favorite fruits. 2. Add a new fruit to the list. 3. Remove a fruit from the list. 4. Print out the second to last elements

Problem 2: Create a program that takes a list of words and returns a new list with only the words that have a length of 5 or more.

```

[2]: # Solution for Problem 2
# Create a list of words
words = ["apple", "banana", "cherry", "date", "elderberry", "fig", "grape"]

# Create an empty list to store the words with a length of 5 or more

```

```

long_words = []

# Iterate over the list of words
for word in words:
    # Check if the length of the word is 5 or more
    if len(word) >= 5:
        # If it is, add the word to the long_words list
        long_words.append(word)

# Print the new list of long words
print(long_words)

```

```
['apple', 'banana', 'cherry', 'elderberry', 'grape']
```

4 List Comprehension

List comprehension is a quick way to generate a list from another list or iterable (such as range, etc).

```

[3]: squares = [x**2 for x in range(10)]
print(squares) # [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

list_to_n = [i for i in range(1001)]
sum(list_to_n)

```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
[3]: 500500
```

5 Tuples

- Tuples are similar to lists in that they are a collection of items, but unlike lists, they are immutable, meaning the elements cannot be modified once created.
- This makes them useful for situations where you want to store data that should not be changed, such as coordinates, dates, or other types of data with a fixed number of elements.

```

[4]: # Creating a tuple
my_tuple = (1, 2, 3)

# Accessing elements of a tuple
print(my_tuple[0]) # Output: 1

# Tuple length
print(len(my_tuple)) # Output: 3

# Tuple concatenation

```

```
new_tuple = my_tuple + (4, 5)
print(new_tuple) # Output: (1, 2, 3, 4, 5)

# Tuple repetition
print(my_tuple * 3) # Output: (1, 2, 3, 1, 2, 3, 1, 2, 3)
```

```
1
3
(1, 2, 3, 4, 5)
(1, 2, 3, 1, 2, 3, 1, 2, 3)
```

6 Practice problem:

Problem 1: Create a function that takes in a string and returns a tuple of the string and its length.

```
string_length("hello")
# Output: ("hello", 5)
```

Problem 2:

1. Create a tuple called `zoo` that contains the animals 'lion', 'tiger', 'giraffe', 'elephant', and 'monkey'.
2. Print the third element of the tuple.
3. Add the animal 'rhinoceros' to the tuple. Hint: You can't add elements to a tuple, so you'll need to create a new tuple.
4. Convert the `new_zoo` tuple to a list and save it as `zoo_list`

7 Check out ticket

Work on all practice problems!