CoGrammar

Sequences: Lists

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Software Engineering Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(FBV: Mutual Respect.)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: **Open Class Questions**

Software Engineering Lecture Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**
- We would love your **feedback** on lectures: **Feedback on Lectures**

Lecture Objectives

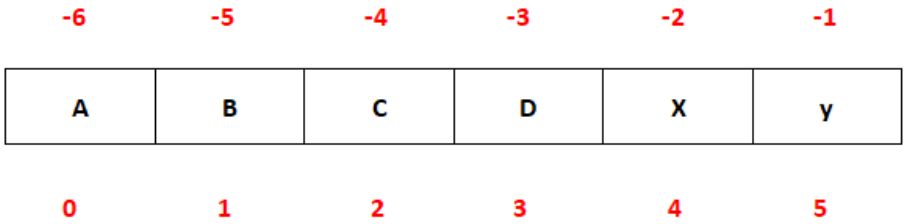
1. Recall the fundamental characteristics of Lists.
2. Explain the concept of indexing in a list.
3. Apply knowledge of lists to manipulate elements.

CoGrammar

Recap on Strings

Lists

- ★ A list is a data type that allows us to store multiple values of any type together and a list can contain duplications.
- ★ We can access individual values using indexing and multiple values using slicing.
- ★ We can iterate over lists using a for loop.



Lists

- ★ Lists are mutable.
- ★ This means the values inside a list can be changed and unlike a string won't return a new list when changes have been made.
- ★ We can apply methods to our lists without having to restore them inside our variables.

Question:

What list method can you use to remove an item using its index?

Lists

- ★ To create a list we can surround comma separated values with square brackets. []
- ★ E.g. my_list = [value1, value2, value3]

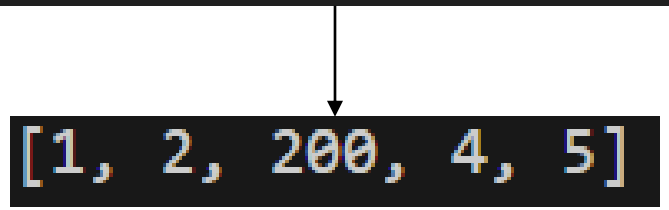
- ★ Adding Elements: append(), insert()
- ★ Removing Elements: remove(), pop() and 'del'
- ★ Manipulating elements: sorting, reversing and slicing

List Example

```
num_list = [1,2,3,4,5]
word_list = ["Word1", "Word2", "Word3"]
```

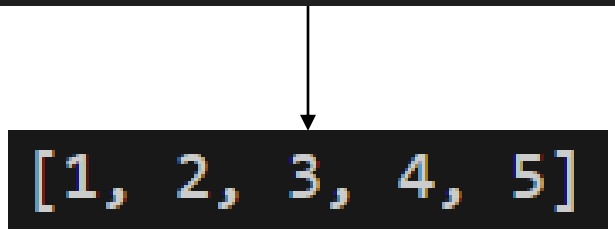
List Example

```
num_list = [1,2,3,4,5]
new_num_list = num_list

new_num_list[2] = 200
print(num_list)
```



List Example

```
num_list = [1,2,3,4,5]
new_num_list = num_list.copy()

new_num_list[2] = 200
print(num_list)
```



Question:

How can I reverse a list using slicing?

CoGrammar

Questions around Lists

Wrapping Up

Lists

LIsts in Python offer a powerful mechanism for organizing and manipulating data in a structured manner.

Indexing

We can access elements in our list with indexing and can use slicing to grab multiple values

List Methods

List methods allow us to manipulate the data within our list very easily and efficiently.

CoGrammar

Thank you for joining