CoGrammar

SE PORTFOLIO SESSION 4

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Software Engineering Lecture Housekeeping

- The use of disrespectful language is prohibited if asking a question. This is a supportive, learning environment for all – please engage accordingly! **(FBV: Mutual Respect.)**
- No question is ‘silly’ – **ask away!**
- 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**

Progression Criteria

✅ **Criterion 1: Initial Requirements**

- Complete 15 hours of Guided Learning Hours and the first four tasks within two weeks.

✅ **Criterion 2: Mid-Course Progress**

- Software Engineering: Finish 14 tasks by week 8.
- Data Science: Finish 13 tasks by week 8.

✅ **Criterion 3: Post-Course Progress**

- Complete all mandatory tasks by 24th March 2024.
- Record an Invitation to Interview within 4 weeks of course completion, or by 30th March 2024.
- Achieve 112 GLH by 24th March 2024.

✅ **Criterion 4: Employability**

- Record a Final Job Outcome within 12 weeks of graduation, or by 23rd September 2024.

Which Python function converts all characters in a string to uppercase?

A. capitalize()
B. upper()
C. lower()
D. startswith()

What does the replace() method do in a Python string?

A. Removes a specified substring.
B. Reverses the string.
C. Replaces a specified substring with another substring.
D. Splits the string into a list.

How can you concatenate two strings in Python?

A. Using the + operator.
B. Using the append() method.
C. Using the join() method.
D. Using the concatenate() function.

Recap: Sequences

String manipulation methods

- built-in modules of code that manipulate and transform textual data(strings)
- essential for data processing and text analysis
- they save time since there is no need to write the code over and over again to perform certain operations.

Escape characters

- **‘\n’** - add new line
- **‘\t’** - add tab space

CoGrammar

Recap: Sequences

Lists

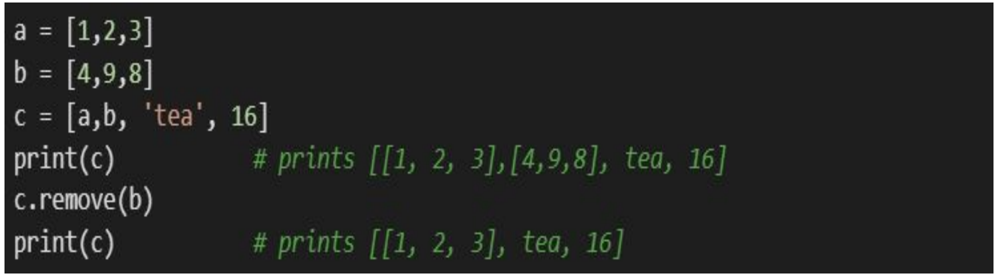
- a data structure that is a changeable, ordered sequence of elements (items)

List methods

- extend(), insert(), remove(), pop(), index(), count(), sort(), reverse()

Nested Lists

- Lists can include other lists as elements

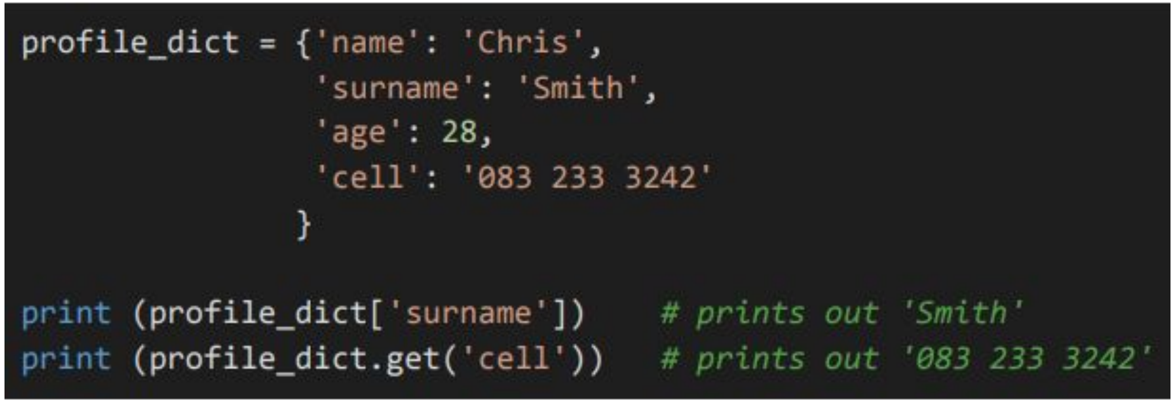


```
a = [1,2,3]
b = [4,9,8]
c = [a,b, 'tea', 16]
print(c)          # prints [[1, 2, 3],[4,9,8], tea, 16]
c.remove(b)
print(c)          # prints [[1, 2, 3], tea, 16]
```

Recap: Sequences

Dictionaries

- a data structure that is unordered and elements are accessed via their keys and not their index positions the way lists are.
- While we use indexing to access elements in a list, dictionaries use keys. Keys can be used to access values by placing them inside square brackets [ ]



```
profile_dict = {'name': 'Chris',
                'surname': 'Smith',
                'age': 28,
                'cell': '083 233 3242'
               }

print (profile_dict['surname'])    # prints out 'Smith'
print (profile_dict.get('cell'))   # prints out '083 233 3242'
```

Simple Scribe

- **Background:** Effective document creation and text editing can be achieved with a word-processing tool that enhances productivity and communication.

- **Challenge:** Create a simple word-processing tool called Simple Scribe that allows users to create and edit documents.

- **Objective:** Once the user has inputted the desired text into the document, the following features will be offered to the user:
  - text formatting (allowing the user to make some text bold, underlined, italicised, capitalised),
  - styles (allows the user to have a set style for titles, headings and bodies in the document),
  - a find-and-replace tool (allows user to find a specific word/character and replace it with another).

Simple Scribe

- **Programming Needs:**
    - String Handling
    - String Manipulation
    - Formatting strings for output
    - Built-in Python string functions

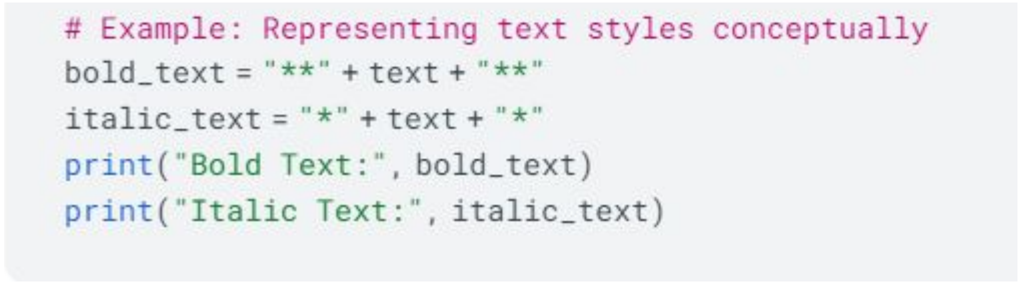
String Manipulation with Built-In Functions

```python
# Example: Formatting text for a document
text = "Welcome to Simple Scribe!"
formatted_text = text.upper() # Convert text to uppercase
print(formatted_text)
```

We initialise a string called ‘text’ containing a sentence. We then format the sentence to upper case using the **.upper()** in-built method. This will give us the output “WELCOME TO SIMPLE SCRIBE!”.

Demo: Developing the Simple Scribe Tool

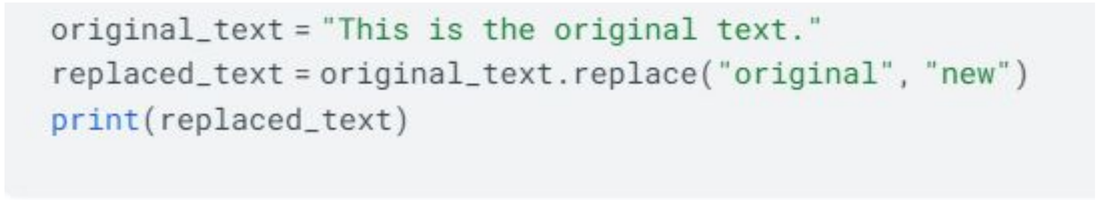
- We use string manipulation techniques to implement text formatting features such as bold and italicising(We represent bold text by putting it between two pairs of “**” we can do the same for italic by using two pairs of “*”:



```python
# Example: Representing text styles conceptually
bold_text = "**" + text + "**"
italic_text = "*" + text + "*"
print("Bold Text:", bold_text)
print("Italic Text:", italic_text)
```

Demo: Developing the Simple Scribe Tool

- We use the in-built method **.replace()** to implement the find and replace feature of our Simple Scribe tool. We can find a given substring and replace it with a new one. This will produce the output “This is the new text.”



```python
original_text = "This is the original text."
replaced_text = original_text.replace("original", "new")
print(replaced_text)
```

Simple Scribe Tool

Your challenge is to develop a Simple Scribe tool using in-built Python string functions and string manipulation techniques.

Here is a list of string methods you might find useful:

replace()
split()
lower()
upper()
“x”.join()
find()

Step-by-Step Tasks:

1. **String Manipulation:** Examples of these include splitting strings into a character array, checking if a string starts with or ends with a specific sequence. Splitting a string using delimiters.
2. **Searching & Extracting:** Search for specific sub-strings and extracting these (storing them separately or temporarily).
3. **Formatting and Data Cleaning:** Standardise your strings (user provided strings or externally sourced) by ensuring that the format won’t present any issues within your code’s functionality.
4. **Applying String Functions:** Simply applying the appropriate string methods/functions to accomplish all of the above.

Advanced Challenge:

- Allow the user to store their data in a text file.

CoGrammar

Summary

String manipulation

★ We can manipulate strings in many different ways by using string methods such as split, strip, remove and more.

Lists and Dictionaries

★ We can use lists and dictionaries to add structure and organization to our data.

2D Lists

★ Using our knowledge of lists we can build 2D list by having a list of lists. We can use this to our advantage by representing a spreadsheet table within a 2D list.

In the Simple Scribe word-processing tool, how might you use the split() function?

A. To change the case of the text.
B. To separate text into lines or paragraphs.
C. To find a specific word in the text.
D. To replace words in the text.

Why is it important to validate user input in a word-processing application?

A. To ensure the text is grammatically correct.
B. To prevent program errors and handle unexpected input.
C. To automatically translate the text.
D. To increase the speed of text processing.

Questions and Answers

Questions around the Case Study