CoGrammar

DS PORTFOLIO SESSION 6

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Data Science 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**

Data Science Session 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 dictionary method can be used to return the value of a key?

A. pop()
B. .keys()
C. .get()
D. .show

Recap of Week 5: Sequences

Managing data using strings and Lists

- 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

String building

- ***

Recap of Week 5: 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 of Week 5: 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'
```

Library Management System

- **Background:** In an effort to move away from their physical management system, the library wants a digital solution to manage book checkouts, returns, and reservations efficiently.

- **Challenge:** You are tasked with creating this new system using lists and dictionaries.

- **Objective:** Once the user has inputted the desired text into the document, the following features will be offered to the user:
  - The program should use lists and dictionaries to keep track of various details.
  - Handle common library operations like checking out a book, which would decrease the number of available copies and add the book to the user's borrowed list.
  - Prevent users from borrowing more books than allowed and calculate fines for late returns.

Library Management System

- **Programming Needs:**
  - String Handling
  - Implementing Dictionary & List Manipulation
  - Functions for calculations/computations

CoGrammar

New Features

- **User Interface**: A user-friendly interface where members can register, login, and navigate through various options.
- **Book Management**: Using lists and dictionaries, manage book checkouts, returns, reservations, and inventory.
- **Member Management**: Track how many books each user has borrowed, reserved, and any associated fines.
- **DNA Sequence Analyser Integration**: As a unique feature, members can input their DNA sequence to get book recommendations based on genetic traits (a fun fictional feature for the case study).
- **Defensive Programming**: Ensure the system is robust against potential errors and misuse. Handle errors gracefully with custom exceptions.
- **Advanced String Handling**: Search for books, authors, or genres using advanced string functions. Provide features like "similar books" based on string matching.
- **Iterative Processes**: For tasks like sending reminders to members with overdue books or calculating monthly fines, use loops effectively.

Demo: Checking Out

```
# Simple example to demonstrate a library checkout
books_inventory = {'Book1': {'copies': 5, 'current_borrowers': []},
                   'Book2': {'copies': 3, 'current_borrowers': []}}

users_info = {'User1': {'borrowed_books': [], 'fines': 0},
              'User2': {'borrowed_books': [], 'fines': 0}}

def checkout_book(book_title, user_name):
    if books_inventory[book_title]['copies'] > 0 and
len(users_info[user_name]['borrowed_books']) < 3:
        books_inventory[book_title]['copies'] -= 1
        books_inventory[book_title]['current_borrowers'].append(user_name)
        users_info[user_name]['borrowed_books'].append(book_title)
        print(f"Book '{book_title}' checked out successfully.")
    else:
        print("Error: Book not available or user has reached the maximum
borrow limit.")

# Call the function
checkout_book('Book1', 'User1')
```

Demo: Returning Books

Here we're using if statements combined with the split() method to Identify specific sequences and their associated traits:

```
def return_book(book_title, user_name):
  if book_title in users_info[user_name]['borrowed_books']:
    books_inventory[book_title]['copies'] += 1
    books_inventory[book_title]['current_borrowers'].remove(user_name)
    users_info[user_name]['borrowed_books'].remove(book_title)
    print(f"Book '{book_title}' returned successfully.")
    else:
        print("Error: User has not borrowed this book.")
```

Library Management

In an effort to move away from their physical management system the library wants a digital solution to manage book checkouts, returns, and reservations efficiently.

Here is a list of some of the methods for your program and potential user-defined functions.

.remove()
.pop()
.append()
calculate_fine()
return_book()
checkout_book()

Extended Concepts:

1. Gathering and processing user input and displaying relevant outputs.
2. Implementing data structures like strings, lists, and dictionaries for data management.
3. Applying defensive programming techniques to ensure system reliability.
4. Utilising loops for iterative processes and tasks.
5. Advanced string handling for search and recommendation features.
6. Professional development: Integrating all learned skills into a comprehensive project.

Summary

Dictionaries

- ★ Dictionaries are the perfect data structure to store user and inventory information.

Combining Lists and Dictionaries

- ★ Remember that lists and dictionaries can both be stored as items within one another.

Methods

- ★ Lists and dictionaries have useful methods to allow you to remove, update, and delete elements.

Which exception would you need to handle to avoid accessing a element that does not exist in a List?

A. DivisionByZeroError
B. TypeError
C. IndexError
D. All of the above.

Questions and Answers

Questions around the Case Study