



CoGrammar

DS PORTFOLIO SESSION 4



**SKILLS
FOR LIFE**

SKILLS BOOTCAMPS



Department
for Education

Data Science Session 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**

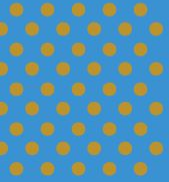

- Complete all mandatory tasks by 24th March 2024.
- Record an Invitation to Interview within 4 weeks of course completion, or by 30th March 2024.
- Achieve 112 GLH by 24th March 2024.

✓ **Criterion 4: Employability**

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


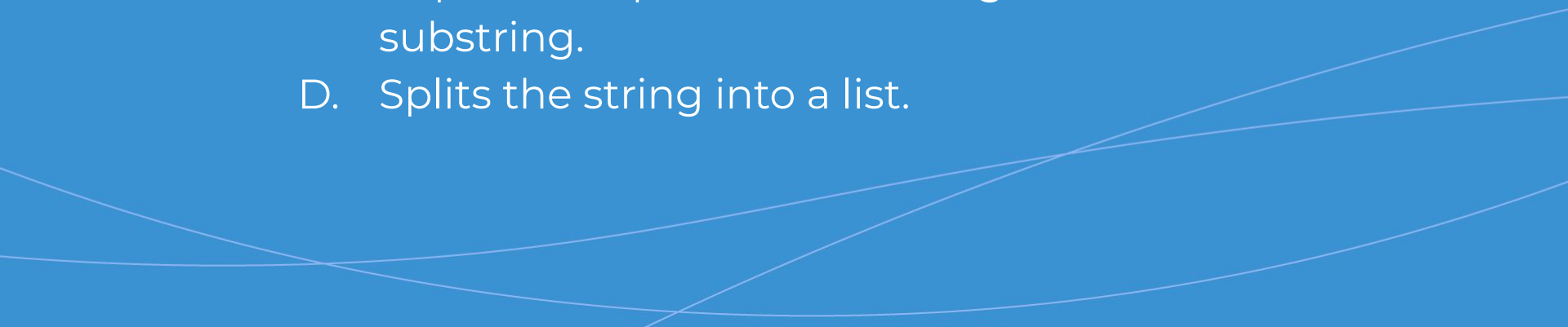


Which Python function converts all characters in a string to uppercase?

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


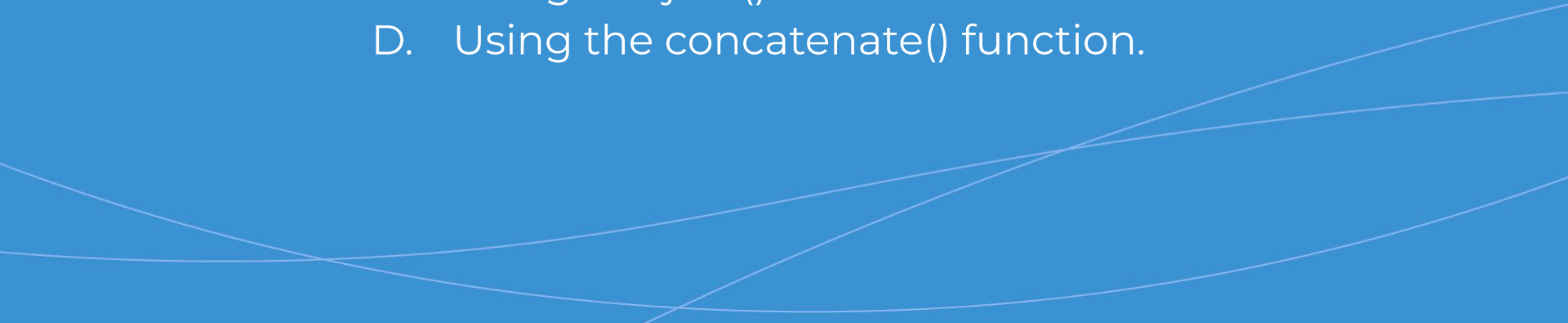


What does the `replace()` method do in a Python string?

- 
- 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 of Week 4: 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

String building

- `***`

Recap of Week 4: 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 4: 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'
```

Geno Tech

- **Background:** Using advanced string functions, the program should identify specific patterns, match them to known genetic markers, and even predict possible traits based on these sequences
- **Challenge:** Your task is to develop a DNA Sequence Analyser.
- **Objective:** Once the user has inputted the desired text into the document, the following features will be offered to the user:
 - Allow the program to be capable of reading long DNA sequences, which are essentially strings of nucleotides represented by the letters **A**, **T**, **C**, and **G**.
 - Identify specific patterns, match them to known genetic markers, and predict possible traits based on these sequences.
 - Additionally, the program should offer functionalities like converting the entire sequence to uppercase for uniformity, replacing specific nucleotide patterns, and splitting sequences based on certain markers.

Geno Tech

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

String Manipulation

```
# Example: Converting a DNA sequence to uppercase and replacing a nucleotide
dna_sequence = "atcgctagctag"
uppercase_sequence = dna_sequence.upper()
replaced_sequence = uppercase_sequence.replace("A", "T")

print("Original:", dna_sequence)
print("Uppercase:", uppercase_sequence)
print("Replaced A with T:", replaced_sequence)
```

Here we see an example on how to format a DNA sequence in Python using the `upper()` and `lower()` methods.

Demo: Predicting Traits

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

```
genetic_marker = "TCGA"
if genetic_marker in uppercase_sequence:
    print("Genetic marker found:", genetic_marker)

segments = uppercase_sequence.split("G")
print("Segments split by 'G':", segments)

trait_sequence = "TACG"
if trait_sequence in uppercase_sequence:
    print("Trait for blue eyes likely present.")
```

Example Logic

1. DNA Sequence Cleaning

2. Finding Genetic Markers

3. Segmenting DNA Sequences

(Splitting sequence into segments for detailed analysis)

4. Identifying Patterns and Predicting Traits

5. Advanced Pattern Analysis

(Checking for multiple traits)

6. Error Checking

Geno Tech

Your task is to develop a DNA Sequence Analyser. Using advanced string functions, the program should identify specific patterns, match them to known genetic markers, and even predict possible traits based on these sequences

Here is a list of some of the methods for your program.

<code>replace()</code>
<code>split()</code>
<code>lower()</code>
<code>upper()</code>
<code>"x".join()</code>
<code>find()</code>

Important Concepts:

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:

- Provide optional context to the user regarding the traits discovered after examining their DNA sequences.

Summary

String Manipulation

- ★ String manipulation involves performing various operations on strings, such as creating, modifying, searching, or transforming them.

Predicting Traits

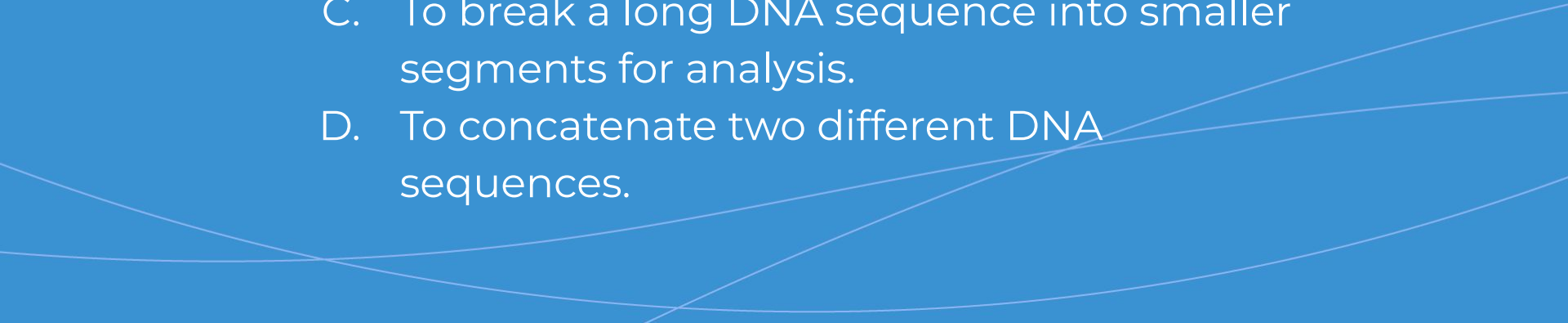
- ★ Using string manipulation to identify specific patterns, match them to known genetic markers, and even predict possible traits based on these sequences.


Formatting & Standardisation

- ★ The process of inserting a custom string or variable in predefined text. This sounds similar to string concatenation but without using “+” or concatenation methods.

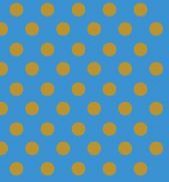


In the GenoTech DNA Sequence Analyser, why might you use the `split()` function?

- A. To convert a DNA sequence to uppercase.
 - B. To replace certain nucleotides with others.
 - C. To break a long DNA sequence into smaller segments for analysis.
 - D. To concatenate two different DNA sequences.
- 



How can you check if a specific genetic marker, like "AGTC," exists in a DNA sequence?



- A. Using the find() function to locate the index of the marker.
- B. Applying a for loop to iterate through each nucleotide.
- C. Checking if the marker is in the string using an if statement.
- D. Splitting the DNA sequence at each occurrence of the marker.



Questions and Answers

Questions around the Case Study

