Welcome to the CoGrammar

Lists, Sets and Tuples

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.

Cyber Security Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- 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 Academic Sessions. You can submit these questions here: **Questions**

Cyber Security Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- We would love your **feedback** on lectures: **Feedback on Lectures**
- Find all the lecture **content** in you **Lecture Backpack** on GitHub.

Safeguarding & Welfare

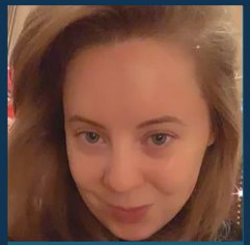
We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Rafiq Manan



Charlotte Witcher

Nurhaan Snyman

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Stay Safe Series:

Mastering Online Safety One week at a Time

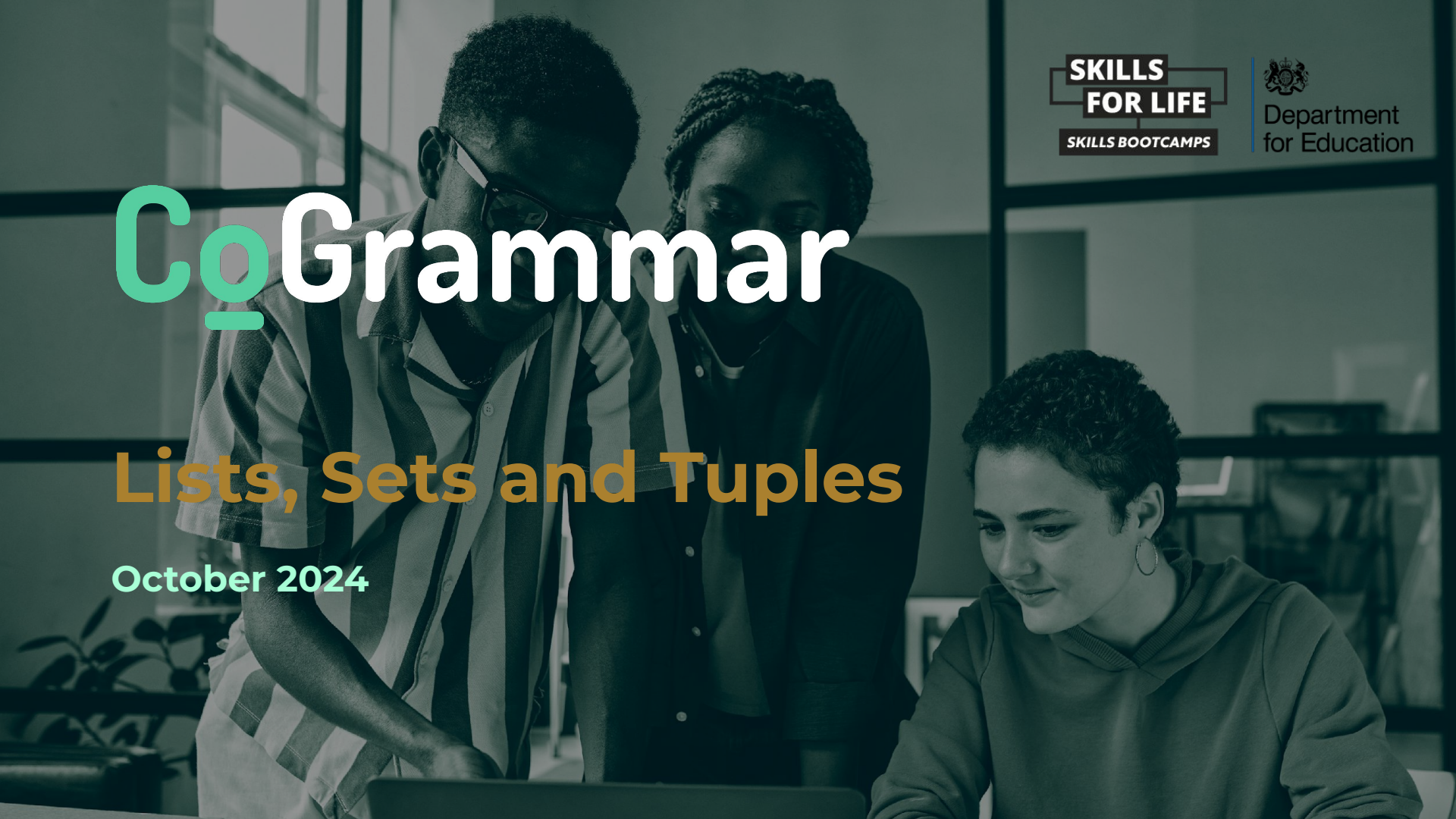
While the digital world can be a wonderful place to make education and learning accessible to all, it is unfortunately also a space where harmful threats like online radicalization, extremist propaganda, phishing scams, online blackmail and hackers can flourish.

As a component of this BootCamp the *Stay Safe Series* will guide you through essential measures in order to protect yourself & your community from online dangers, whether they target your privacy, personal information or even attempt to manipulate your beliefs.

Digital Decorum:

The Importance of Respect in Online Spaces

From safeguarding personal data to promoting positive online behavior, mutual respect plays a crucial role in preventing security breaches, protecting privacy, and fostering a respectful digital world.



Lists, Sets and Tuples

October 2024

Learning Objectives & Outcomes

- Define **lists**, **sets**, and **tuples** in Python.
- Identify the differences between these data structures and when to use each one.
- Perform basic operations like adding, removing, and accessing elements in **lists**, **sets**, and **tuples**.
- Understand the immutability of tuples and how it affects data manipulation.

Non-Primitive Data Structures.

Imagine you have a playlist of your favorite songs or a collection of your favorite books. How would you like to organize this collection? Would you prefer the flexibility to easily add, remove, or rearrange items as your tastes evolve, or would you prefer to keep it exactly as it is, preserving its original form?

Polls

Please have a look at the poll notification and select an option.

What do you think a list is in programming?

A. A collection of items that can change
B. A way to store only unique items
C. A fixed collection of items that cannot be changed
D. A type of error in code.

Polls

Please have a look at the poll notification and select an option.

If you have a list of students and want to ensure each student can only appear once, what would be the best way to manage this?

A. Use a list and check for duplicates each time.
B. Use a set to automatically handle uniqueness
C. Use a tuple to store student names
D. Use a dictionary to map students to their grades

Python Collections

- Collections are ways to group multiple elements together.
- They help in managing large amounts of data efficiently
- Three main types of collections:
  - Lists
  - Sets
  - Tuples

Lists

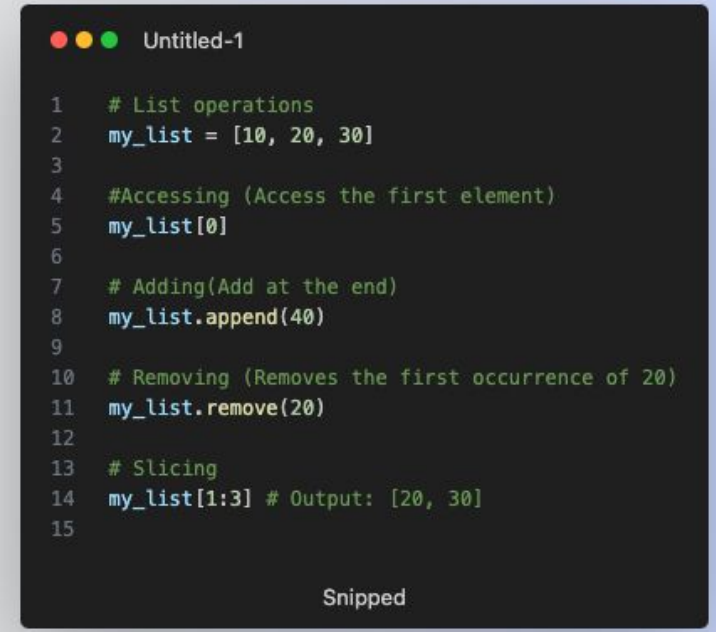
- Lists are ordered, mutable collections that allow duplicate elements. (Dynamic arrays)

- Key Features:
  - Ordered: Elements maintain the order in which they were added.
  - Mutable: Elements can be changed.
  - Supports duplicate items: Supports duplicate items.

Lists Operations

- Common List operations:
  - Accessing elements
  - Adding elements
  - Removing elements
  - Slicing

Lists


Untitled-1
1 # List operations
2 my_list = [10, 20, 30]
3
4 #Accessing (Access the first element)
5 my_list[0]
6
7 # Adding(Add at the end)
8 my_list.append(40)
9
10 # Removing (Removes the first occurrence of 20)
11 my_list.remove(20)
12
13 # Slicing
14 my_list[1:3] # Output: [20, 30]
15
Snipped

Sets

- Sets are unordered collections that only allow unique elements
- Sets are like mathematical sets (no duplicates)
- Key features:
  - Unordered: No indexing or ordering of elements
  - Mutable: You can add or remove elements
  - Cannot contain duplicate items

Sets Operations

- Common Set Operations
  - Adding elements
  - Removing elements
  - Union: Combines two sets
  - Intersection: Common elements in sets

Sets

Untitled-1

```python
# Sets Operations
my_set = { 1, 2, 3 }

my_set2 = {4, 5, 6}

# Adding elements
my_set.add(4)

# Removing elements
my_set.remove(2)

# Union: Combines two sets
my_set.union(my_set2)

# Intersection: Outputs the common elements in sets
my_set.intersection(my_set2)
```

Snipped

CoGrammar

Tuples

- Tuples are ordered, immutable collections that allow duplicate elements.
- Once created, tuples cannot be changed
- Key features:
    - Ordered
    - Immutable
    - Supports duplicate items

Tuples Operations

- Common Tuple Operations:
    - Accessing Elements
    - Slicing
    - Unpacking: Assigns tuple elements to variables

Tuples

Untitled-1

```python
# Tuples
my_tuple = ( 1, 2, 3 )

# Accessing elements
my_tuple[0]

# Slicing
my_tuple[1:3] # Output: (2, 3)

# Unpacking
a, b, c = my_tuple # Output: a=1, b=2, c=3
```

Snipped

CoGrammar

When to use each?

- Lists: Use when you need an ordered collection that can change over time and may contain duplicates
- Sets: Use when you need a collection of unique elements and don’t care about order
- Tuples: Use when you need an ordered, unchangeable collection, such as coordinates or fixed data.

Summary

- Lists are for ordered, mutable collections with duplicates.
- Sets are for unique, unordered collections.
- Tuples are for ordered, immutable collections.
- Each collection type serves different purposes, depending on the task.

Polls

What will be the result of the following code?

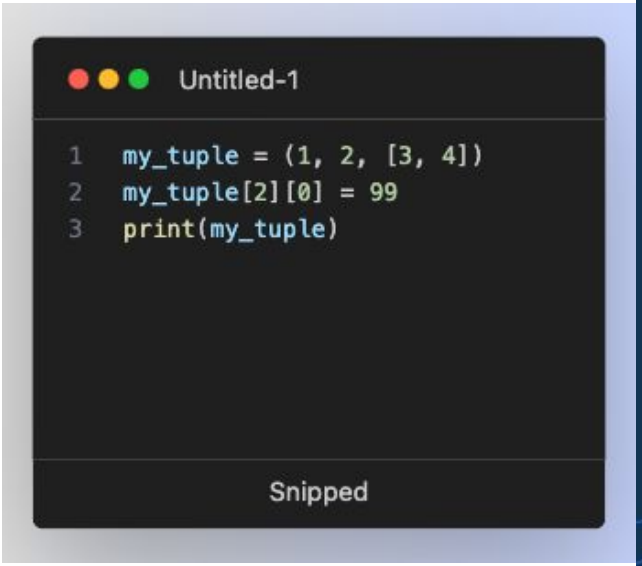
A. { 1. 2. 3. [4, 5] }
B. TypeError
C. { 1, 2, 3, 4, 5 }
D. [1, 2, 3, [4, 5]]

Polls

Please have a look at the poll notification and select an option.

What will be the output of this code?

A. (1, 2, [99, 4])
B. TypeError: 'tuple' object does not support item assignment
C. (1, 2, 99, 4)
D. (1, 2, [3, 4])



Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar