CoGrammar

Welcome to this session: Task Walkthrough - Tasks 6 - 9

The session will start shortly...

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

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

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

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

CoGrammar HyperionDev

Skills Bootcamp Data Science

- 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]

Skills Bootcamp Data Science

- 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***

- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

Learning Outcomes

- ❖ **Define and use functions** to structure reusable data processing workflows.
- ❖ **Perform file I/O operations** to read and write structured data.
- ❖ **Work with Python collections and sequences** (lists, dictionaries, sets, and tuples) to store, manipulate, and retrieve data.
- ❖ **Integrate functions, collections, and I/O operations** to automate tasks like data extraction, transformation, and summary generation.

Lecture Overview

→ Presentation of the Task
→ Functions
→ Sequences
→ IO Operations
→ Task Walkthrough

CoGrammar

Task Walkthrough

Imagine you've just joined the data science team at an e-commerce company. Your boss hands you a messy file filled with thousands of customer transactions and asks you to analyze and clean the data, extract insights, and generate a sales report. But who has time to do all of that manually? That's where your Python skills come in!

- ❖ **Read sales data from a file and structure it properly.**
- ❖ **Process customer transactions using Python lists and dictionaries.**
- ❖ **Use functions to clean and organize the data efficiently.**
- ❖ **Write a final report that highlights top customers and sales trends.**

What is the main advantage of using functions in Python?

A. It makes the program execute faster
B. It reduces code duplication and improves reusability
C. It automatically optimizes memory usage
D. It forces Python to execute code in order

Which of the following is an example of an immutable data structure in Python?

A. List
B. Tuple
C. Dictionary
D. String

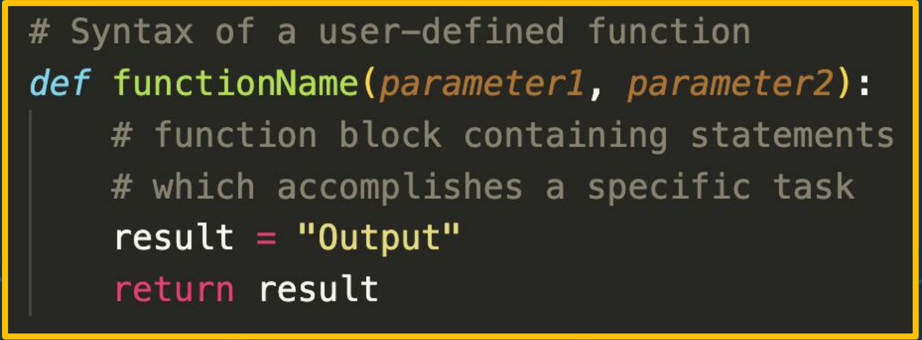
CoGrammar

Functions

HyperionDev

Functions

- To declare a function in Python, we use the **def** keyword.
- We have to provide a **name** for our function (using variable naming conventions), a list of **parameters** (placeholders for function inputs) in brackets, a colon and **body** of the function indented.
- We also need to add a **return statement** for functions that return a value. This is not necessary for all functions e.g. functions that modify a state.

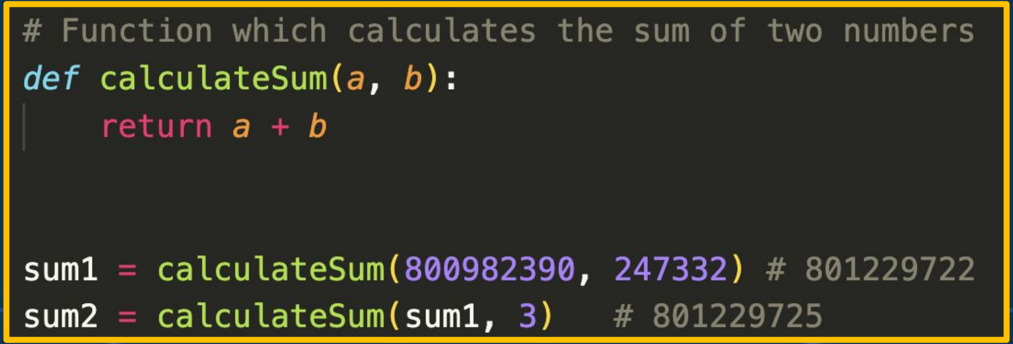


```python
# Syntax of a user-defined function
def functionName(parameter1, parameter2):
    # function block containing statements
    # which accomplishes a specific task
    result = "Output"
    return result
```

HyperionDev

Functions

- After defining a function, we **call or invoke** it to use it in our code.
- We call a function with its name followed by a list of **arguments** enclosed in brackets, if required by the functions.
- **Arguments** are the input values provided to the function and take the place of the **parameters** defined in the function in the **same position**.



```python
# Function which calculates the sum of two numbers
def calculateSum(a, b):
    return a + b


sum1 = calculateSum(800982390, 247332) # 801229722
sum2 = calculateSum(sum1, 3)    # 801229725
```

HyperionDev

Sequences

HyperionDev

Lists

Ordered, mutable collections of data.

- Items in a list are known as **elements**.
- Elements do not have to be unique nor of the same type.
- Lists are **mutable**, meaning that elements in the list can be changed.
- Use **square brackets** to create lists and separate values with **commas**:

```
my_list = [1, "two", "buckle", True]
```

- We can access elements in a list using indexing, which is based on the element's position in the list:

```
print(my_list[0])  # 1
print(my_list[3])  # True
```

HyperionDev

Lists

❖ The most commonly used list functions are:

➢ Adding an element

```
my_list.append("three")
my_list.insert(0, "zero")
```

➢ Deleting an element

```
my_list.remove("zero")
my_list.pop(3)
```

➢ Manipulating the list: sorting, reversing etc.

```
my_list.sort()
my_list.reverse()
```

Strings

- Strings are considered to be **immutable** collections of sequences in Python.
- We can access characters in our strings the same way we can access elements in a list:

```
string = "hello"
print(string[0]) # h
```

- We can also manipulate strings using the same methods that we use on lists:

```
string.find("h")
```

HyperionDev

Dictionaries

Collections of key-value pairs, where each key is unique.

- ❖ Unlike lists, dictionaries distinguish each element in the collection using a **key** instead of an **index**.
- ❖ When we use dictionaries to study languages, we look up definitions of a given word by looking up the word in the dictionary.
- ❖ In the data structure, we can access the **value associated with a key** value by looking up the key in the dictionary.
- ❖ Each element in a dictionary is a **key-value pair**.
- ❖ To create a dictionary, keys and values are **separated by colons (:)** and pairs are **separated by commas** and **enclosed in curly brackets {}**.

HyperionDev

Dictionaries

- We can also use the **dict** function to create dictionaries:

```
my_dict = {"name": "Zahra", "age": 24}
my_dict = dict(name = "Zahra", age = 24)
```

- To access values in a dictionary:

```
my_name = my_dict["name"]
```

- To add elements to a dictionary:

```
my_dict["bday"] = "13 November"
```

- To delete elements in a dictionary:

```
my_dict.pop("bday")
```

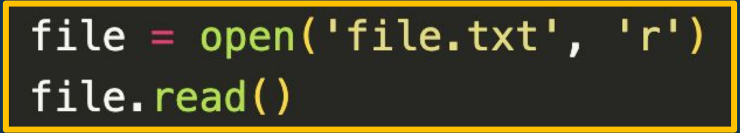
HyperionDev

IO Operations

HyperionDev

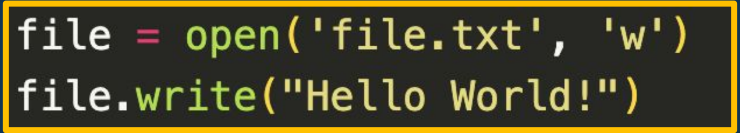
File Modes

- **Read** text from a file with the **mode 'r'**



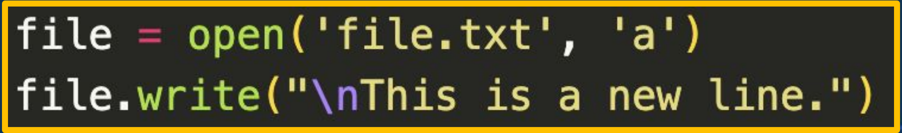
```python
file = open('file.txt', 'r')
file.read()
```

- **Write** text to a file with the **mode 'w'**



```python
file = open('file.txt', 'w')
file.write("Hello World!")
```

- **Append** text to an existing file with the **mode 'a'**



```python
file = open('file.txt', 'a')
file.write("\nThis is a new line.")
```

File Handling (Reading)

Read from a File Python
Methods

read()
Reads the entire contents of the file and returns it as a string.

readline()
Reads a single line from the file and returns it as a string.

readlines()
Reads all lines from the file and returns them as a list of strings.

CoGrammar

File Handling (Writing)

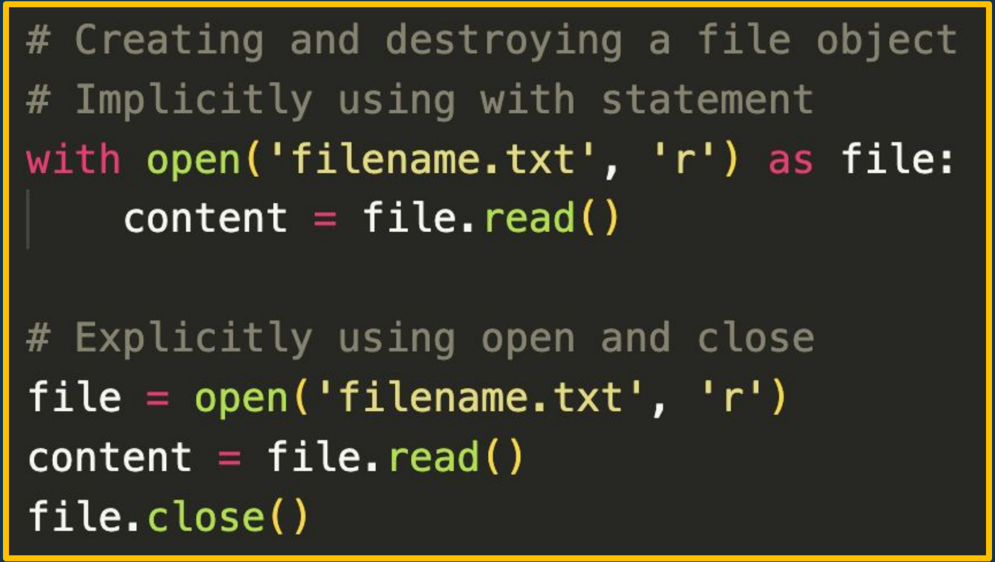
Write to a File Python
Methods

write()
This method is used to write data to the file. It takes a string argument and adds it to the end of the file.

writelines()
This method writes a sequence of strings to the file. It takes a list of strings as an argument and writes each string to the file.

CoGrammar

Resource Management



```python
# Creating and destroying a file object
# Implicitly using with statement
with open('filename.txt', 'r') as file:
    content = file.read()

# Explicitly using open and close
file = open('filename.txt', 'r')
content = file.read()
file.close()
```

HyperionDev

Task Walkthrough

Imagine you’ve just joined the data science team at an e-commerce company. Your boss hands you a messy file filled with thousands of customer transactions and asks you to analyze and clean the data, extract insights, and generate a sales report. But who has time to do all of that manually? That’s where your Python skills come in!

- **Read sales data from a file and structure it properly.**
- **Process customer transactions using Python lists and dictionaries.**
- **Use functions to clean and organize the data efficiently.**
- **Write a final report that highlights top customers and sales trends.**

Which function should be used to write data to a text file in Python?

A. writefile()

B. file.write()

C. text.output()

D. printfile()

CoGrammar

What will the .get() method return if the key is not found in a dictionary?

A. It raises an error
B. It returns None or a specified default value
C. It deletes the dictionary entry
D. It automatically adds the key to the dictionary

Summary

★ **User-Defined Functions:**
Modularizing operations like data validation and analysis.

★ **Strings:**
Validating email formats and formatting output.

★ **Lists and Dictionaries:**
Storing participant objects and managing survey questions and responses.

★ **File I/O:**
Reading and saving data to/from files for persistence.

CoGrammar

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education