Welcome to this CoGrammar session:

Functions and APIs

The session will start shortly…

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

CoGrammar

Software Engineering 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**

Software Engineering 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**

Skills Bootcamp
8-Week Progression Overview

✅ Criterion 3: Course Progress

- ***Completion:*** All mandatory tasks, including Build Your Brand and resubmissions by study period end
- ***Interview Invitation:*** Within 4 weeks post-course
- ***Guided Learning Hours:*** Minimum of 112 hours by support end date (10.5 hours average, each week)

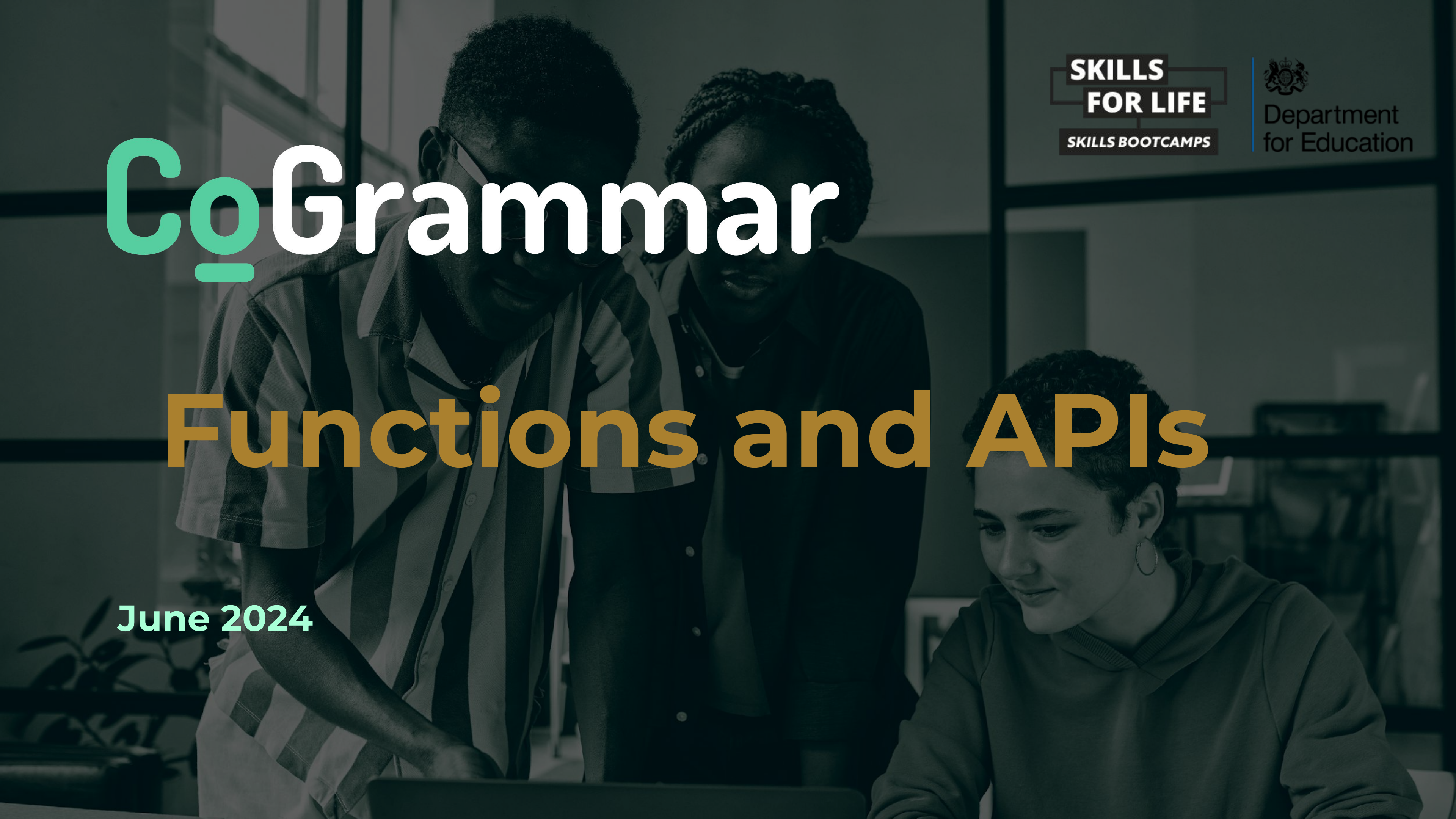
✅ Criterion 4: Demonstrating Employability

- ***Final Job or Apprenticeship Outcome:*** Document within 12 weeks post-graduation
- ***Relevance:*** Progression to employment or related opportunity

CoGrammar

Learning Outcomes

- Explain what functions are.
- Explain what user defined functions are.
- Use built-in functions within their projects.
- Create user-defined function within their projects.
- Explain what APIs are.
- Explain what the python requests module is.
- Utilise the requests module to make calls to API endpoints.
- Utilise the data from API calls within their programs.



SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

Functions and APIs

June 2024

What are Functions?

- Functions are reusable blocks of code that perform specific tasks.
- Called methods when used in OOP Classes.
- Functions help organise code, make it more readable, and facilitate debugging and maintenance.
- Useful for abstraction.
- Similarity to functions in maths, f(x) takes input x and produces some output.

Calling Functions?

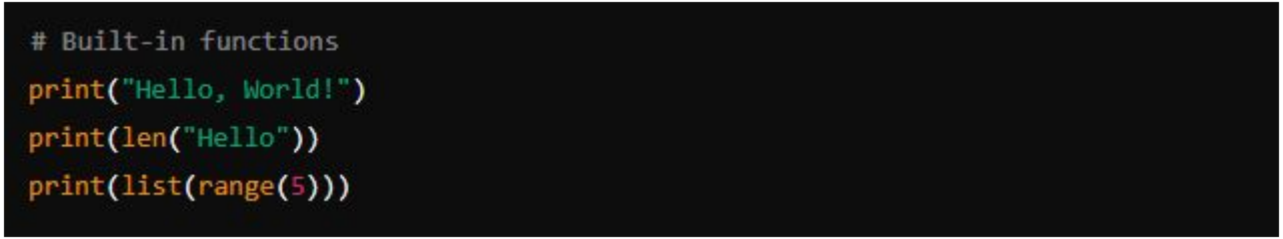
- Functions with one required positional input:
  - my_function(input1)
- Functions with two required positional inputs:
  - my_function(input1, input2)
- Functions with one required positional input and one optional keyword input:
  - my_function(input1, **keyword_arg**=input2)

Why Use Functions?

- Code Reusability: Write once, use multiple times.
- Modularity: Break down complex problems into simpler pieces.
- Maintainability: Easier to update and fix issues in a modular codebase.
- Abstraction: Hide complexity and expose simple interfaces.
- Error checking/validation: Makes this easier, as you can define all rules in one place.

Using Built-In Functions

- Python provides numerous built-in functions for common tasks.
- Examples: print(), len(), and range()



```python
# Built-in functions
print("Hello, World!")
print(len("Hello"))
print(list(range(5)))
```

More Python Functions

- The list of functions that you can use in Python doesn’t just stop with what is built in.
- Using Pip (python package manager), you can install various packages containing modules.
- To search for packages, visit https://pypi.org/
- Some packages are already installed by default in Python, such as the Math package.
- These modules can be imported into your script using an import statement.

More Python Functions

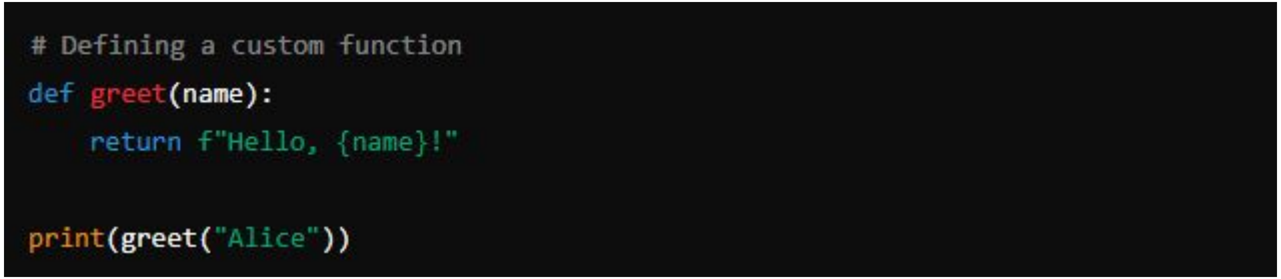
- Let’s take a look at the maths module. Let’s say that you want to use round(), which rounds a number off.
- There are multiple ways to access this:
  - import math –or- from math import *

    my_result = math.round(my_num, 2)
  - from math import round

    my_result = round(my_num, 2)

Creating Custom Functions

- Use the def keyword to define a function.
- Define a function to greet a user.

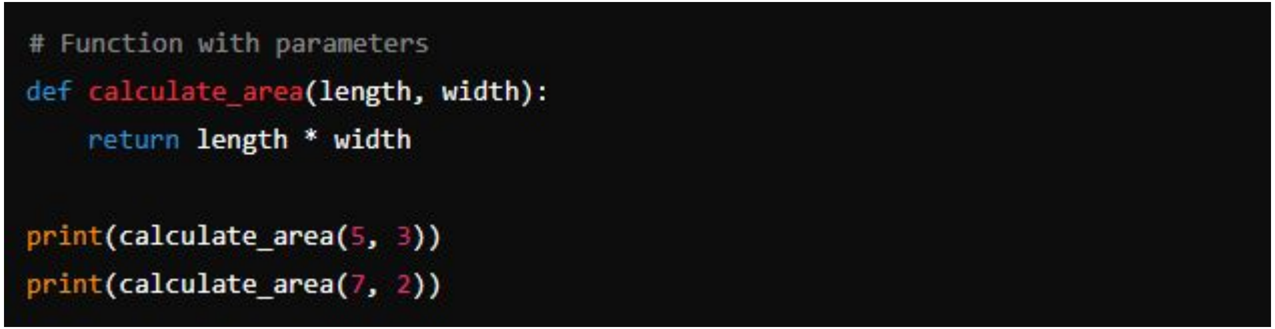


```python
# Defining a custom function
def greet(name):
    return f"Hello, {name}!"

print(greet("Alice"))
```

Functions with Parameters

- Functions can accept inputs through parameters.
- Example: Calculate the area of a rectangle.

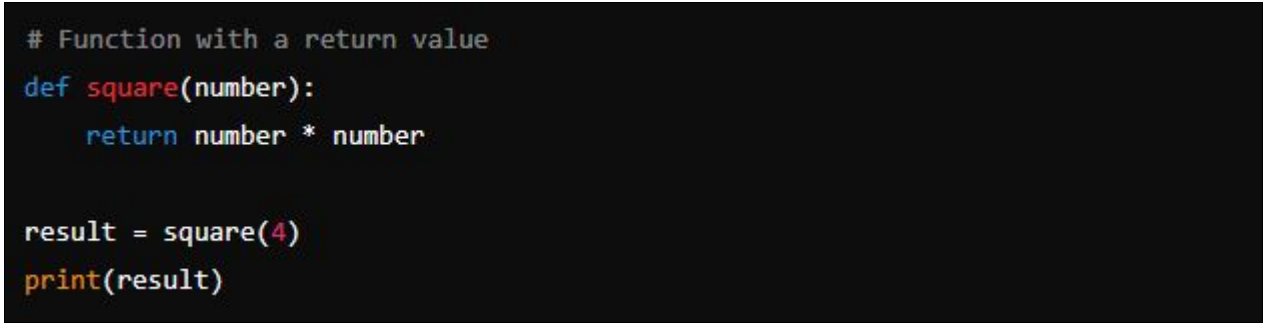


```python
# Function with parameters
def calculate_area(length, width):
    return length * width

print(calculate_area(5, 3))
print(calculate_area(7, 2))
```

Functions with Return Values

- Use the return statement to return a result from a function.
- Example: Return the square of a number.



```
# Function with a return value
def square(number):
    return number * number

result = square(4)
print(result)
```

APIs

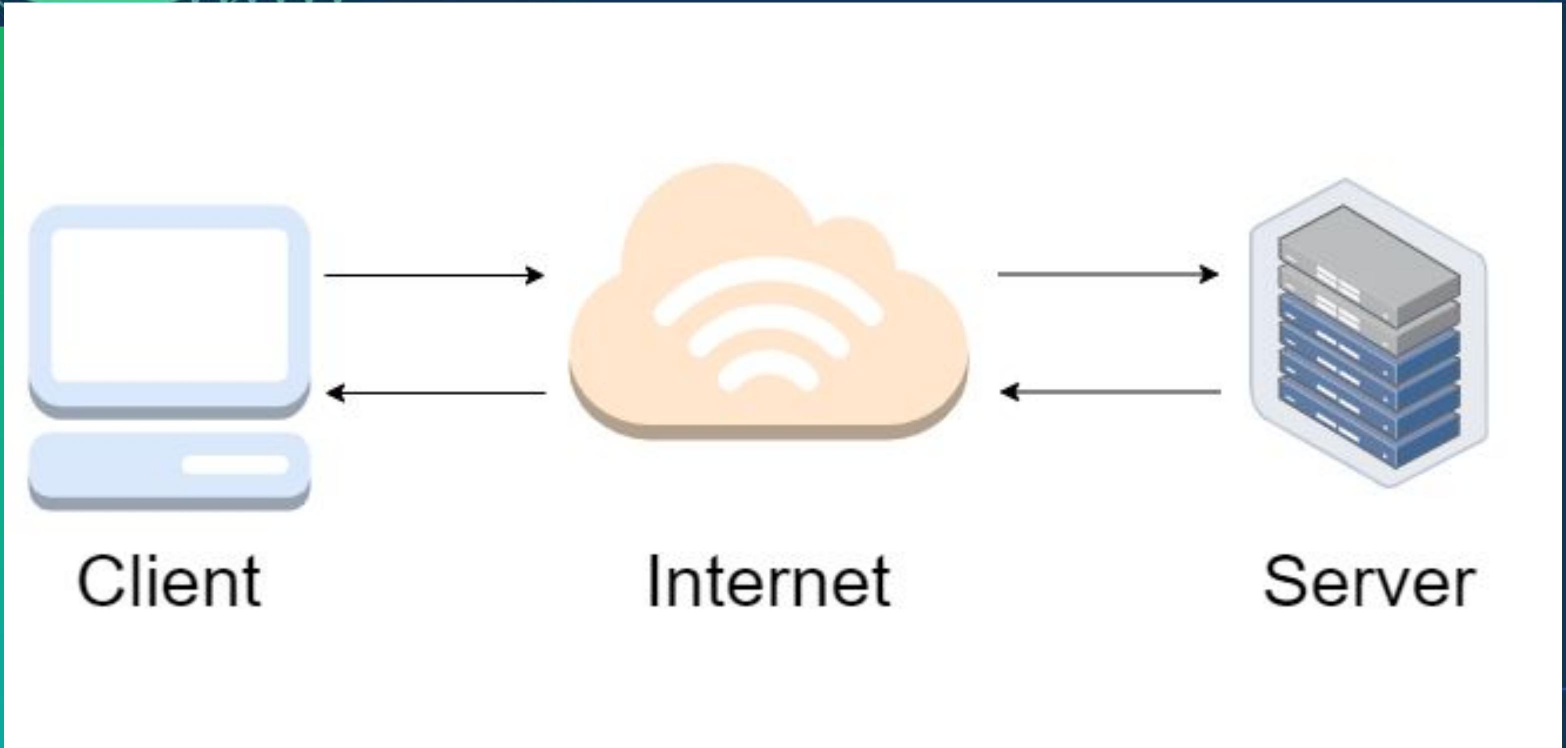
CoGrammar

APIs

- **API:** Application Programming Interface – Interface that provides a set of functions to a user where the underlying mechanism of that function is hidden.

- **Web API:** It is an API that travels through the internet. Usually called client-server architecture.

Web APIs



CoGrammar

API Terminologies

Integration: API integration refers to the process of **connecting** two or more **applications** or **systems** by using **APIs** (Application Programming Interfaces) to **exchange data** and **perform actions.**

Call: The process of **sending** a **request** to your API after setting up the right **endpoints**

Monetization: API monetization is a process by which a business can create **revenue** from its **APIs**.

HTTP and Status Codes

HTTP

HTTP : **H**yper**T**ext **T**ransfer **P**rotocol

https://www.google.com/search/q=python

200

400

500

REST

{REST}

Representational State Transfer

Endpoint

Method

Header

Body

https://dummy.restapiexample.com/

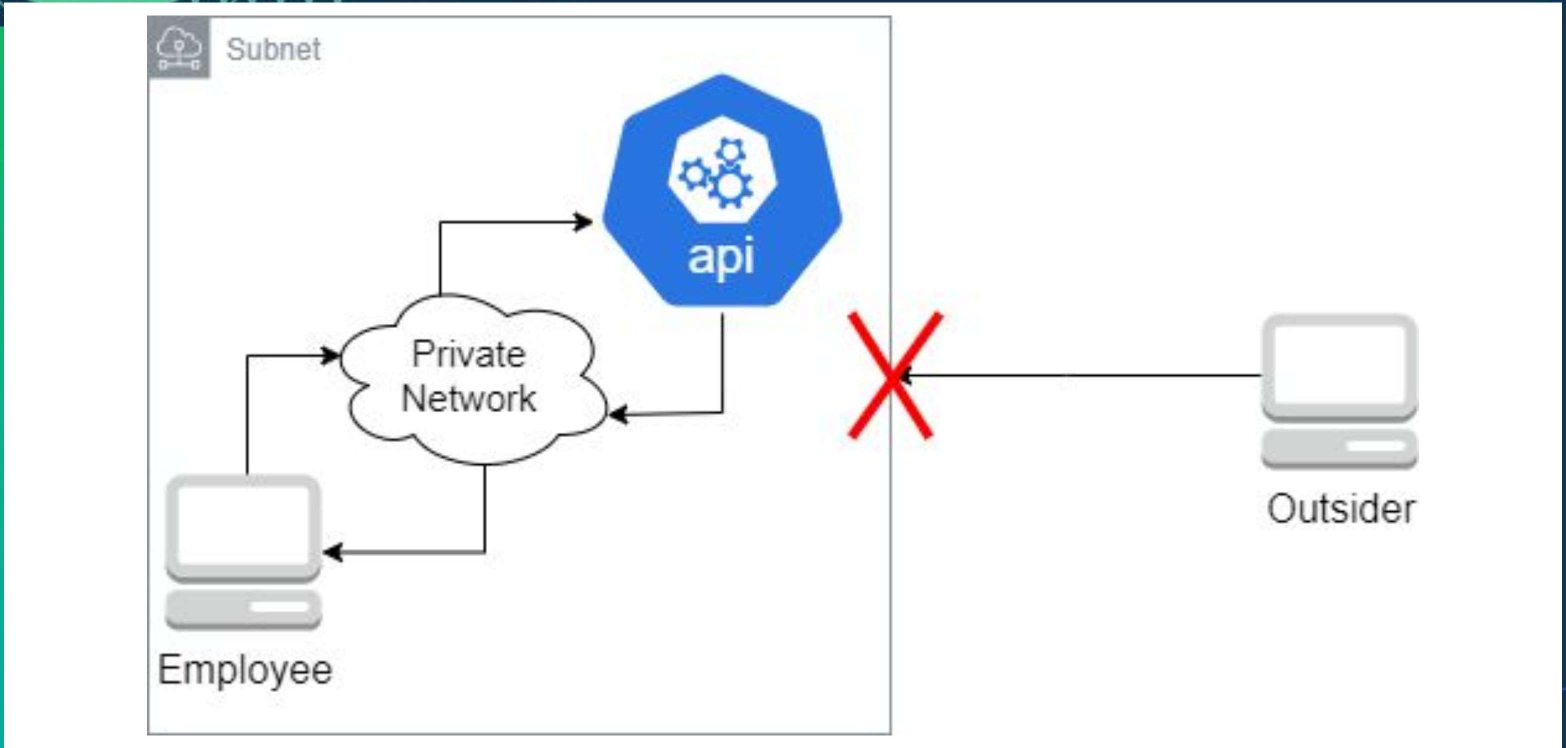
HTTP Payload Format

xml Payload {.json}

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <name>Tim</name>
    <age>23</age>
    <salary>50000</salary>
</root>
```

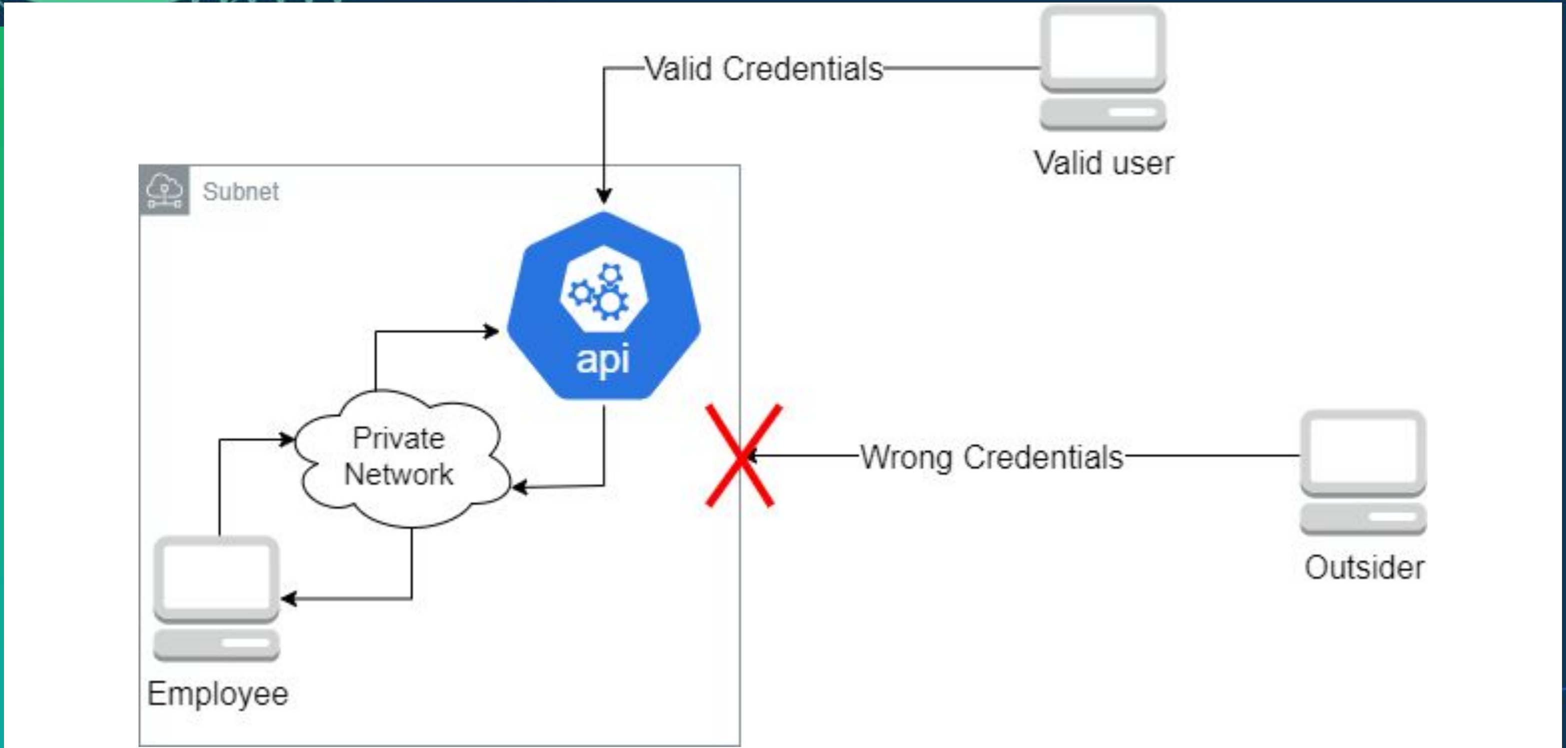
```
{
    "name: "Tim",
    "age: "23",
    "salary: "50000"
 }
```

API Types



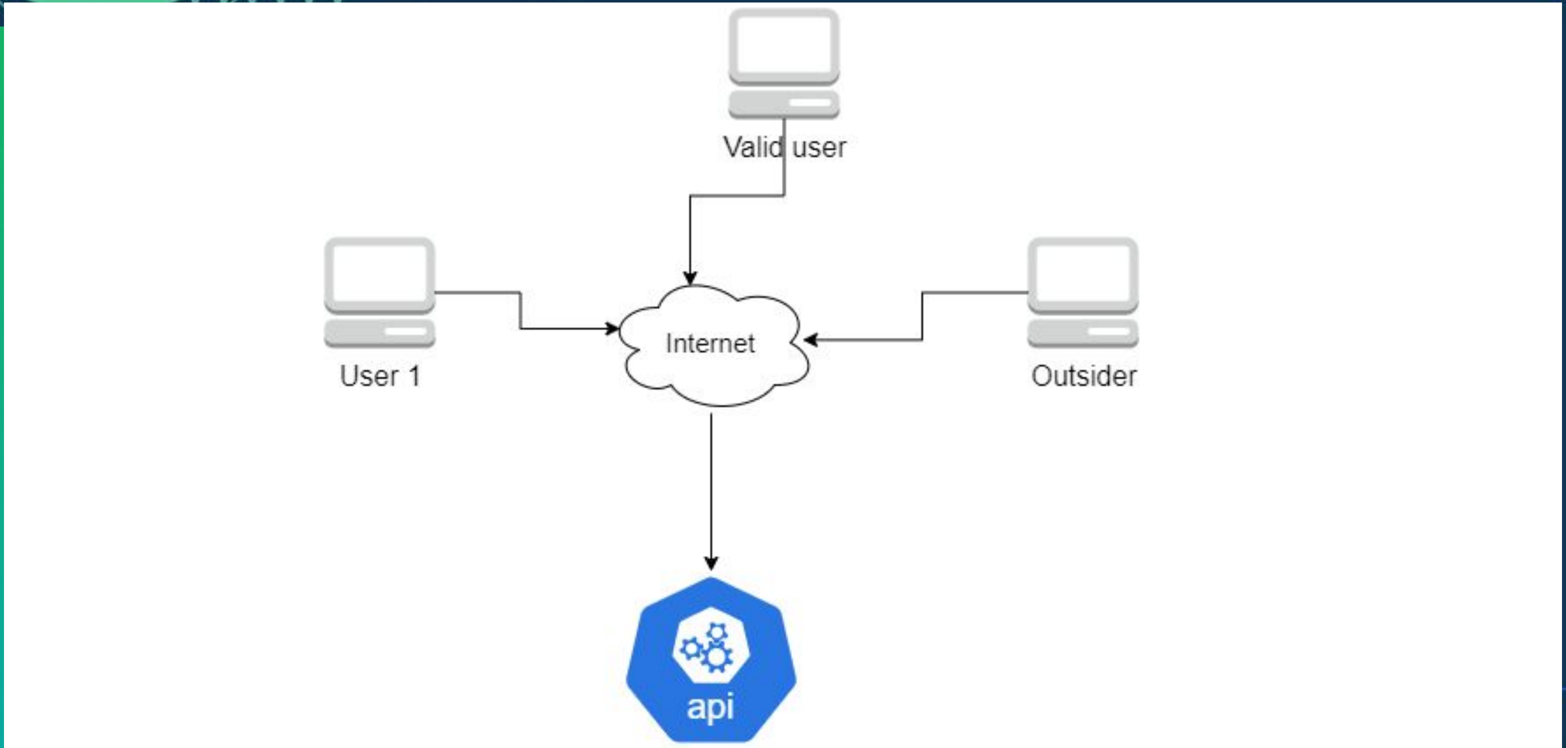
CoGrammar

API Types



CoGrammar

API Types



Let’s get coding!

CoGrammar

Final Assessment

CoGrammar

Polls

- *Refer to the polls section to vote for you option.*

What is the primary purpose of the Python requests module?

a. To send HTTP requests

b. To manage file I/O operations

c. To create graphical user interfaces

d. To perform mathematical calculations

Polls

- *Refer to the polls section to vote for you option.*

Which attribute of the response object contains the status code of the HTTP response?

a. response.status

b. response.code

c. response.status_code

d. response.http_code

CoGrammar

Lesson Conclusion and Recap

Lesson Conclusion and Recap

- Functions: We can group code together and call to it whenever we need it. Benefits code reuse, organisation and maintenance as well as allow for abstraction.
- APIs: The Glue of the Digital World: APIs (Application Programming Interfaces) act as bridges between software systems, enabling seamless communication, data exchange, promoting interoperability and integration.
- RESTful Foundations: REST (Representational State Transfer) is a common architectural style for APIs, using HTTP methods (GET, POST, PUT, DELETE) to interact with resources.
- Consuming APIs: With Python’s requests library, we can send HTTP requests to interact with APIs, retrieving data with GET requests and sending data with POST requests.

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar