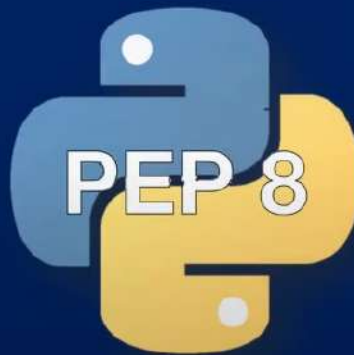


11

What is PEP8?



- Python Enhancement Proposal
- It is a set of rules that specify how to format Python code for maximum readability

11th question what is pep8 Pep stands for python enhancement



8:57 / 41:56 Looking for Python Online Training? Call us at IN: 9606058406 / US: 18338555775 or visit www.edureka.co



SUBSCRIBE

12

What is PYTHONPATH?

- Environment variable used when a module is imported
- The interpreter uses it to determine which module to use



13

What are Python modules? Name some commonly used built-in modules in Python

- Files containing Python code
- Can be functions, classes or variables
- It is a .py file containing executable code

14

What is break and continue in Python?

Break

The break statement is used to exit the loop prematurely when a certain condition is met

14

What is break and continue in Python?

Continue

The continue statement is used to skip the current iteration of the loop and continue with the next iteration.

15

Can we pass a function as an argument in Python?

"YES" in Python you can pass a function as an argument to another function because functions in Python are first-class objects

17

What is type conversion in Python?

Type conversion refers to the conversion of one data type into another

17

What is type conversion in Python?

int()

Converts any data type into integer type

float()

Converts any data type into float type

ord()

Converts characters into integers

hex()

Converts integers into hexadecimals

17

What is type conversion in Python?

oct()

Converts integers into octal

tuple()

This function is used to convert to a tuple

set()

This function returns the type after converting to set

list()

This function is used to convert any data type to a list type

Returns the type after converting to set
list this function is



18

What is docstring in Python?

Enclosed in triple quotes(""" """)
and can span multiple lines

```
def greet(name):
```

```
    """
```

```
    This function greets the user with the provided name.
```

```
    Parameters:
```

```
    - name (str): The name of the person to greet.
```

```
    Returns:
```

```
    - str: A greeting message including the provided  
    name.
```

```
    """
```

```
    return f "Hello, {name}! Welcome to our program."
```

```
# Accessing the docstring  
print(greet.__doc__)
```

203M

19

What is slicing in Python?

[start : end : step]

271M

19

What is slicing in Python?

[start : end : step]

[start : end]

20

What is pickling and unpickling ?

Pickling

The Pickle module accepts any Python object converts it into a string representation and dumps it into a file by using the dump function

20

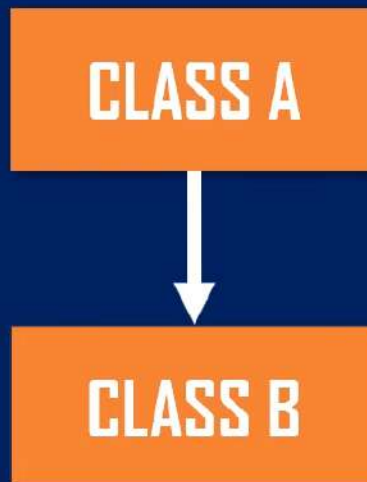
What is pickling and unpickling ?

Unpickling

The process of retrieving original python objects from the string representation

21

Explain Inheritance in Python?



It allows one class to gain all the members (attributes and methods) of another class

21

Explain Inheritance in Python?



21

Explain Inheritance in Python?

Single Inheritance

Multi-level Inheritance

Hierarchical Inheritance

Multiple Inheritance

22

Does Python make use of access specifiers?

In Python you can control how attributes and methods of objects are accessed



209M



22

Does Python make use of access specifiers?

`' _`



Protected

`—`

`' _`



Private

257M

23

Discuss different ways of deleting an element from a list?

remove()

```
list1 = [1, 2, 3, 4]
```

```
list1.remove(2)
```

```
print(list1)
```



[1,3,4]



23

Discuss different ways of deleting an element from a list?

pop()

```
list1.pop(1)
```

```
print(list1)
```



[1,4]

24

How are classes created in Python?

Classes are created using
'class' keyword

Class ClassName:
def function()

25

Write a program in Python to execute the bubble sort algorithm

Bubble sort is a straightforward sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order

28

What is the difference between a shallow copy and a deep copy

Shallow Copy

- Creates a new object by copying their references
- Changes in copied objects will affect the original

28

What is the difference between a shallow copy and a deep copy

Deep Copy

- Creates a new object by copying all the object it contains
- Changes will not affect the original object

34

What is the process of compilation and linking in Python?

Compilation

- For Python code there is no separate compilation step
- Compilation takes place for C extensions
- Handled by compilers such as GCC on Unix-like systems or MSVC on windows

34

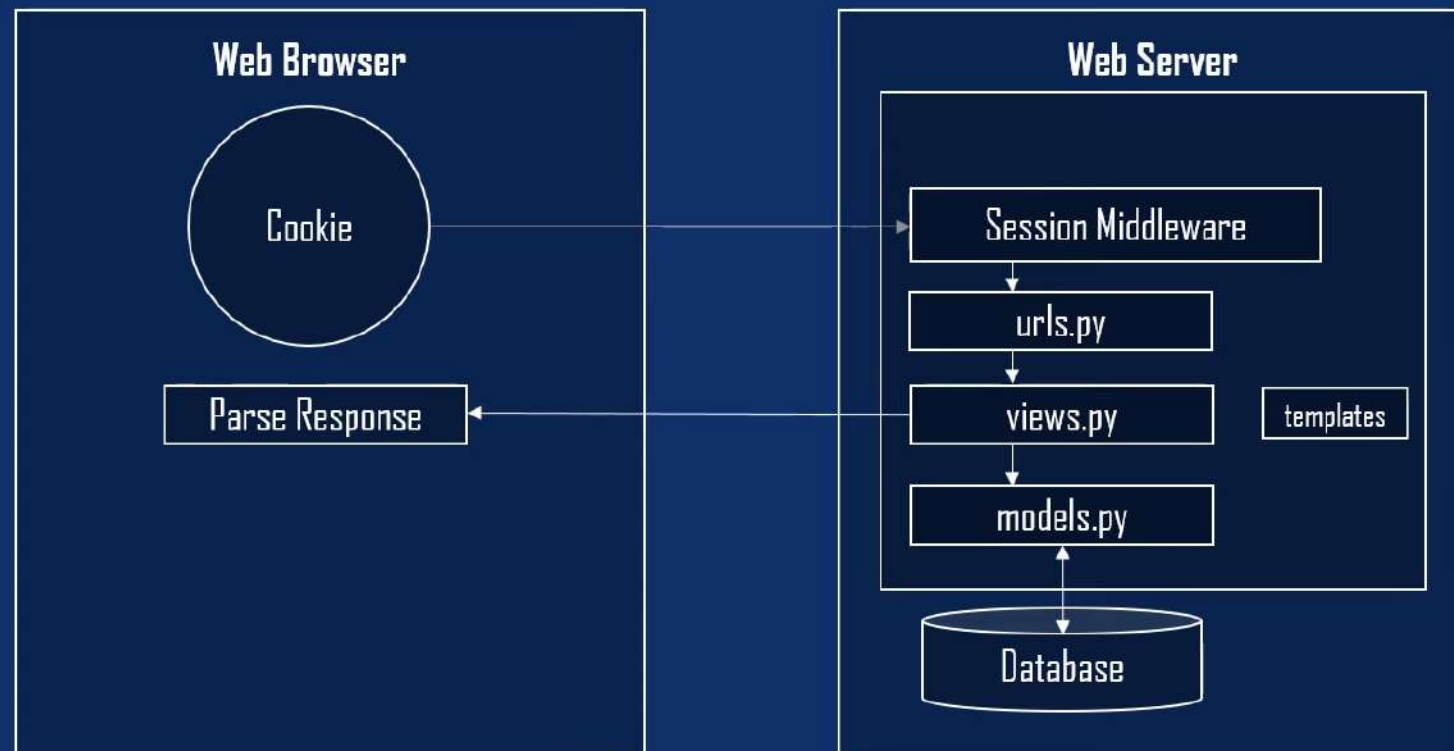
What is the process of compilation and linking in Python?

Linking

- References between different object files are resolved
- Necessary libraries are linked to the final executables
- Handled by the linker such as LD on unix-like systems or link.exe on windows

41

What is the use of sessions in the Django Framework?



41

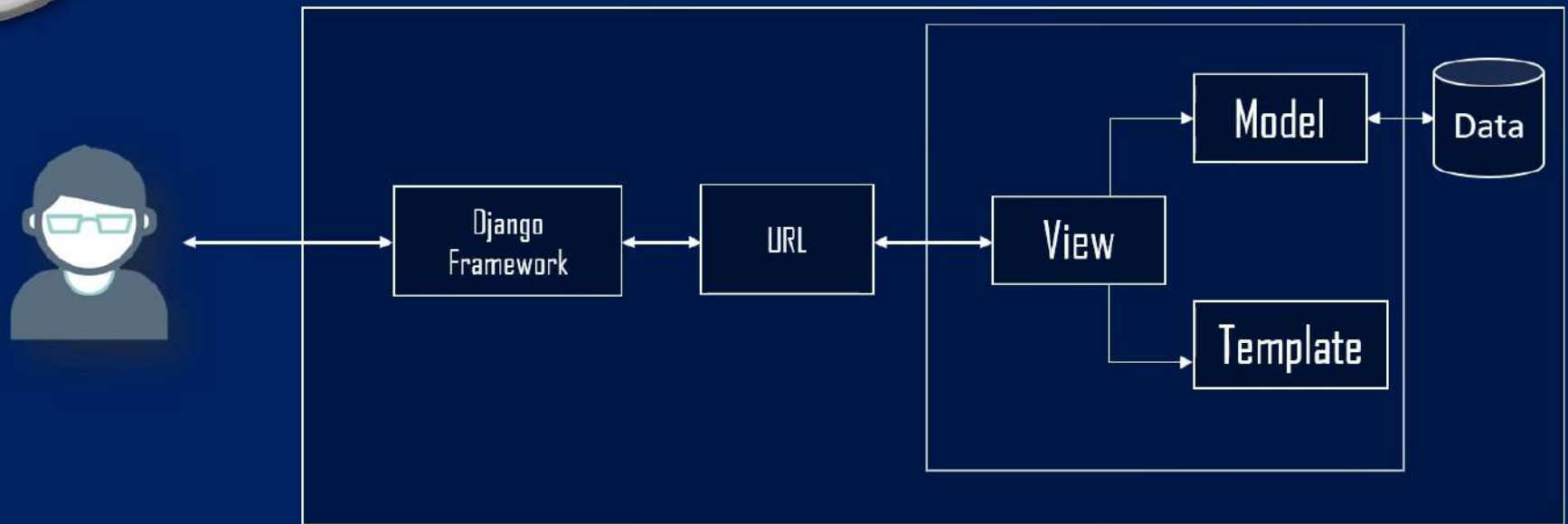
What is the use of sessions in the Django Framework?

Enables the server to remember data about a user's activity and preferences



43

Discuss Django Architecture



MVT Architecture

43

Discuss Django Architecture

Model Layer

- Represents the application's data structure
- Interacts with the database using Django's ORM
- Defines the data schema



43

Discuss Django Architecture

View Layer

- Handles business logic and user interaction
- Process data from model layer
- Use templates to render dynamic HTML content for the user

43

Discuss Django Architecture

Template Layer

- Manages the presentation of data to the user
- Generate dynamic content based on data passed from view layer

44

List out the inheritance styles in Django

Abstract Base Classes

203M

44

List out the inheritance styles in Django

Abstract Base Classes

They don't create their own tables in the database, instead they are like blueprints that other models can use

44

List out the inheritance styles in Django

Multi-table Inheritance

It's when a new model is created based on an existing one, and each new model gets its own table in the database



Subscribe

44

List out the inheritance styles in Django

Multi-table Inheritance

Allows to add more features to an existing model without changing it directly

44

List out the inheritance styles in Django

Proxy Models

They have the same field and methods, but their working can be changed without affecting the database

44

List out the inheritance styles in Django

Proxy Models

They don't create new tables, they just act as alternative views of the original data

44

List out the inheritance styles in Django

Proxy Models

Can create a proxy model by setting a special attribute to True in the model's meta class

46

How to save an image locally using Python whose URL address I already Know?

```
import requests
```

```
def save_image_from_url(url, filename):
```

```
    try:
```

```
        # Get the image from the URL
```

```
        response = requests.get(url)
```

```
        # Check if the image was fetched successfully  
        (status code 200)
```

```
        if response.status_code == 200:
```

```
            # Save the image to a file on your computer  
            with open(filename, 'wb') as f:
```

```
                f.write(response.content)
```

```
            print(f"Image saved as {filename}")
```

```
    else:
```

```
        print("Couldn't download the image. Status code:",  
              response.status_code)
```

```
        except Exception as e:
```

```
            print("An error occurred:", str(e))
```

```
# Example usage
```

```
url = "https://example.com/image.jpg" # Put the web  
address (URL) of the image here
```

```
filename = "image.jpg" # Choose the name for the image  
file you want to save
```

```
save_image_from_url(url, filename)
```

48

How to send an email in Python?

```
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
```

```
# Email configuration
sender_email = 'your_email@gmail.com'
receiver_email = 'recipient_email@gmail.com'
password = 'your_password'
```

```
# Create a message object
message = MIMEMultipart()
message['From'] = sender_email
message['To'] = receiver_email
message['Subject'] = 'Test Email from Python'
```

```
# Email body
body = 'Hello, this is a test email sent from Python!'
message.attach(MIMEText(body, 'plain'))
```