How can you implement an authorization interceptor in JavaScript for an Angular application?

By modifying the Angular source code directly

By defining authorization logic in the router guards

By using third-party libraries only

By creating a custom HTTP interceptor using Angular's HttpClient and injecting it into the app's HTTP interceptor chain

Which of the following is NOT a valid HTTP status code returned by Django views?

200 (OK)

303 (See Other)

1000 (Custom)

401 (Unauthorized)

What command is used to run the Django development server?

django-admin startserver

python runserver

python manage.py runserver

python manage.py server

How do you remove duplicates from a list in Python?

By using the distinct() function on the list object.

By using the remove_duplicates() method on the list object.

By converting the list to a set and then back to a list.

By using the unique() method on the list object.

What is the purpose of a Python decorator?

Decorators are used to extend the functionality of a function or method without modifying its code directly.

Decorators are used to create graphical user interfaces (GUIs) in Python.

Decorators are used to perform mathematical operations on numeric data.

Decorators are used to define classes and objects in Python.

How can you create a serializer that includes all fields from a model in Django REST Framework?

By defining each field manually within the serializer class

By passing the model class as an argument to the serializer class constructor

By using the @all_fields decorator on the serializer class

By using the fields = '__all__' attribute in the serializer class

How do you define a class-based view in Django?

By using the @staticmethod decorator

By using the @classmethod decorator

By defining a Python function

By subclassing Django's built-in generic view classes

What is the main difference between class-based views and function-based views in Django?

Class-based views provide an object-oriented way of organizing view code, while function-based views are based on Python functions

Function-based views provide better flexibility and reusability than class-based views

Class-based views are faster than function-based views

There is no significant difference between them

What is the purpose of Django's ForeignKey field?

The ForeignKey field in Django is used to define a many-to-many relationship between models.

The ForeignKey field in Django is used to define a one-to-one relationship between models.

The ForeignKey field in Django is used to define a one-to-many relationship between models.

The ForeignKey field in Django is used to define a reverse relationship between models.

What is the purpose of the related_name attribute in Django models?

The related_name attribute in Django models is used to define the name of the database table.

The related_name attribute in Django models is used to define a custom name for the reverse relationship from the related model back to the current model.

The related_name attribute in Django models is used to specify the verbose name of the model.

The related_name attribute in Django models is used to handle multiple models in a single database query.

How can you specify HTTP methods for function-based views in Django?

By defining them in the urls.py file

By using function decorators like @http_method on the view function

By modifying the Django source code directly

By using conditional statements based on the request method inside the view function

What is a serializer in Django REST Framework?

A component used to convert complex data types like querysets and model instances into native Python datatypes that can then be easily rendered into JSON, XML, or other content types

A decorator used to define custom API endpoints

A function used to handle HTTP requests and generate responses

A class used to manage authentication in Django REST Framework

What is the difference between Python's == and is operators?

The == operator checks for object identity, whereas the is operator checks for the equality of values.

The == operator checks for the equality of values, whereas the is operator checks for object identity, i.e., whether two variables refer to the same object in memory.

Both the == and is operators check for object identity.

Both the == and is operators check for the equality of values.

<div class="relative max-w-[var(--user-chat-width,70%)] rounded-3xl bg-token-message-surface px-5 py-2.5">What is the output of print(3 * '7')

77

777

21

7

What does the range() function in Python return?

A list of integers

A dictionary of integers

A tuple of integers

An iterator of integers

What is a Python dictionary and how do you create one?

A Python dictionary is an ordered collection of key-value pairs enclosed in parentheses (()). You can create a dictionary by specifying key-value pairs separated by colons inside the parentheses, like ('key1': 'value1', 'key2': 'value2').

A Python dictionary is an unordered collection of key-value pairs enclosed in curly braces ({}). You can create a dictionary by specifying key-value pairs separated by colons inside the braces, like {'key1': 'value1', 'key2': 'value2'}.

A Python dictionary is an ordered collection of key-value pairs enclosed in square brackets ([]). You can create a dictionary by specifying key-value pairs separated by commas inside the brackets, like [('key1', 'value1'), ('key2', 'value2')].

A Python dictionary is a collection of ordered, mutable elements enclosed in square brackets ([]). You can create a dictionary by specifying key-value pairs separated by commas inside the brackets, like [1, 2, 3].

What is a Python list and how do you create one?

A Python list is a collection of ordered, immutable elements enclosed in parentheses (()). You can create a list by specifying elements separated by commas inside the parentheses, like (1, 2, 3).

A Python list is a collection of unordered, mutable elements enclosed in parentheses (()). You can create a list by specifying elements separated by commas inside the parentheses, like (1, 2, 3).

A Python list is a collection of unordered, immutable elements enclosed in curly braces ({}). You can create a list by specifying key-value pairs inside the braces, like {1: 'a', 2: 'b'}.

A Python list is a collection of ordered, mutable elements enclosed in square brackets ([]). You can create a list by simply separating the elements with commas inside the brackets, like [1, 2, 3].

What is the purpose of authentication and permissions in Django REST Framework?

To control access to API endpoints based on user authentication and permissions

To handle serialization logic for API endpoints

To manage database migrations in Django projects

To define URL patterns for API endpoints

How do you use generic views in Django?

By defining them in the settings.py file

By subclassing Django's GenericView class

By using decorators like @generic_view on view functions

By importing them from rest_framework.generics and including them in URL patterns

What is the difference between a regular serializer and a model serializer in Django REST Framework?

There is no difference between them

A model serializer can only be used with model instances, whereas a regular serializer can be used with any type of data

A model serializer automatically generates serializer fields based on the corresponding model fields, whereas a regular serializer requires manual definition of fields

A regular serializer is faster than a model serializer