1. **مبانی پایتون**

2. **ساختار داده‌ها**

3. **الگوریتم‌ها**

4. **برنامه‌نویسی شی‌گرا**

5. **توابع و ماژول‌ها**

6. **مدیریت خطا و استثنا**

7. **عملیات بر روی فایل‌ها**

8. **مدیریت حافظه و کارایی**

9. **تست و دیباگ**

10. **کتابخانه‌های استاندارد و شخصی**

هر دسته شامل سوالات از سطح آسان تا سخت خواهد بود. اینجا شروع می‌کنیم:

---

1. مبانی پایتون

سطح آسان

1. **متغیر** سوال: متغیر `a` را به 10 تغییر دهید و مقدار آن را در پایتون تعریف کنید.

پاسخ:

```python
a = 10
```

2. **نوع داده‌ی** سوال: نوع داده‌ی `True` در پایتون چیست؟

پاسخ: نوع داده‌ی `True` در پایتون از نوع `bool` است.

سطح متوسط

3. **تفاوت بین `==` و `is`** سوال: تفاوت بین `==` و `is` چیست؟

پاسخ: `==` برای مقایسه مقدارها استفاده می‌شود، در حالی که `is` برای مقایسه اشیاء و بررسی هویت آنها استفاده می‌شود.

در پایتون به چه معنی است؟ `None` **:سوال** .4

.دارد `NoneType` نمایانگر عدم وجود مقداری است و نوع داده‌ای خاص به نام `None` **:پاسخ**

سطح سخت ####

سوال:** چگونه می‌توانید متدهایی را به یک کلاس اضافه کنید که بعد از ساخت نمونه‌ی آن کلاس به وجود آمده‌اند؟** .5

.برای اضافه کردن متد به یک نمونه استفاده کنید `types.MethodType` پاسخ:** می‌توانید از تابع**

```python

from types import MethodType

class MyClass:

pass

def new_method(self):

print("This is a new method")

instance = MyClass()

instance.new_method = MethodType(new_method, instance)

instance.new_method()

```

وجود دارد؟ `copy` و `deepcopy` سوال:** در پایتون چه تفاوتی بین** .6

تمام زیرشی‌ها را نیز کپی می‌کند و تغییرات در کپی‌ها بر روی شیء `deepcopy` تنها یک کپی سطحی از شیء ایجاد می‌کند، در حالی که `copy` **:پاسخ**
.اصلی تاثیری ندارد

ساختار داده‌ها .2 ###

سطح آسان ####

سوال:** چگونه یک لیست در پایتون تعریف می‌کنید؟** .1

:پاسخ

```python
```

```
my_list = [1, 2, 3, 4, 5]
```

چیست؟ `list` و `tuple` سوال:** تفاوت بین** 2.

قابل تغییر و برای ذخیره داده‌های قابل تغییر استفاده `list` غیرقابل تغییر است و برای ذخیره داده‌های ثابت استفاده می‌شود، در حالی که `tuple` **:پاسخ**
.می‌شود

سطح متوسط

تبدیل کنید؟ `list` را به یک `set` سوال:** چگونه می‌توانید یک** 3.

:پاسخ

```python
my_set = {1, 2, 3}
my_list = list(my_set)
```

چیست؟ `list` و `deque` سوال:** تفاوت بین** 4.

برای این عملیات بهینه `list` برای افزودن یا حذف عناصر از هر دو انتهای صف بهینه‌تر است، در حالی که (double-ended queue) `deque` **:پاسخ**
.نیست

سطح سخت

سوال:** چگونه می‌توانید یک لیست را بدون استفاده از توابع داخلی معکوس کنید؟** 5.

:پاسخ

```python
def reverse_list(lst):
    reversed_list = []
    for item in lst:
        reversed_list.insert(0, item)
    return reversed_list
```

استفاده کنید و مزایای آن چیست؟ `heapq` سوال:** چه زمانی باید از** 6.

(کپه) در پایتون استفاده می‌شود و مزایای آن شامل عملیات سریع بر روی کوچک‌ترین (یا بزرگ‌ترین) heap برای مدیریت ساختار داده‌ی `heapq` **:پاسخ**
.عنصر است

---

3. الگوریتم‌ها

سطح آسان

1. **سوال:** الگوریتم مرتب‌سازی حبابی چیست و چگونه کار می‌کند؟**

پاسخ: الگوریتم مرتب‌سازی حبابی با مقایسه و جابجایی عناصر مجاور برای مرتب کردن یک لیست استفاده می‌شود. این عملیات را تا زمانی که هیچ جابجایی دیگری نیاز نباشد تکرار می‌کند.**

2. **سوال:** الگوریتم جستجوی دودویی چیست و چگونه کار می‌کند؟**

پاسخ: الگوریتم جستجوی دودویی برای جستجو در یک لیست مرتب شده استفاده می‌شود. با مقایسه عنصر میانه با عنصر جستجو شده و کاهش دامنه جستجو به نصف هر بار، این الگوریتم سریع‌تر از جستجوی خطی است.**

سطح متوسط

3. **سوال:** الگوریتم `quick sort` چیست و چگونه عمل می‌کند؟**

پاسخ: الگوریتم `quick sort` بر اساس زیرلیست به دو لیست تقسیم و تسخیر است که با انتخاب یک عنصر به عنوان 'پارتیشن' و تقسیم مقایسه با پارتیشن عمل می‌کند.**

4. **سوال:** توضیح دهید که چگونه الگوریتم `Dijkstra` برای یافتن کوتاه‌ترین مسیر در گراف عمل می‌کند.**

پاسخ: الگوریتم `Dijkstra` با استفاده از یک ساختار داده‌ی priority queue برای انتخاب و به روز رسانی مسیر کوتاه‌ترین مسیر از منبع به تمام نقاط دیگر در گراف عمل می‌کند.**

سطح سخت

5. **سوال:** توضیح دهید که چگونه الگوریتم `A*` مشابه `Dijkstra` چیست؟ `Dijkstra` برای یافتن کوتاه‌ترین مسیر استفاده می‌شود و تفاوت آن با الگوریتم `A*` algorithm

پاسخ: الگوریتم `A*` مشابه `Dijkstra` است اما علاوه بر هزینه مسیر تا کنونی، از تابع تخمینی (heuristic) برای برآورد هزینه کل تا هدف استفاده می‌کند و بنابراین می‌تواند سریع‌تر عمل کند.**

6. **سوال:** چگونه می‌توانید الگوریتم `Knapsack` مسئله‌ی انتخاب را برای بیشترین ارزش در کوله‌پشتی با ظرفیت معین پیاده‌سازی کنید؟**

پاسخ: الگوریتم `Knapsack` با استفاده از برنامه‌نویسی پویا (dynamic programming) برای هر ظرفیت ممکن از کوله‌پشتی و هر آیتم ممکن، حداکثر ارزش قابل انتخاب را محاسبه می‌کند.**

برنامه‌نویسی شی‌گرا .4 ###

سطح آسان ####

سوال:** در پایتون، چگونه می‌توانید یک کلاس تعریف کنید؟** .1

:پاسخ

```python
class MyClass:
def __init__(self, value):
self.value = value
```

سوال:** تفاوت بین ارث‌بری ساده و ارث‌بری چندگانه چیست؟** .2

.پاسخ:** ارث‌بری ساده شامل ارث‌بری از یک کلاس پایه است، در حالی که ارث‌بری چندگانه شامل ارث‌بری از چندین کلاس پایه است**

سطح متوسط ####

.سوال:** توضیح دهید که چگونه می‌توانید متدهای سازنده و تخریب‌کننده را در پایتون تعریف کنید** .3

.در پایتون تعریف می‌شوند `__del__` و متد تخریب‌کننده با نام `__init__` پاسخ:** متد سازنده با نام**

```python
class MyClass:
def __init__(self, value):
self.value = value

def __del__(self):
print("Object is being destroyed")
```

را در پایتون پیاده‌سازی کنید و از آنها استفاده کنید؟ (special methods) سوال:** چگونه می‌توانید متدهای خاص** .4

.با نام‌های خاص تعریف می‌شوند و برای سفارشی کردن رفتارهای کلاس استفاده می‌شوند (`__repr__`, `__str__` مثل) پاسخ:** متدهای خاص**

```python
class MyClass:
def __str__(self):
return "MyClass instance"
```

```

سطح سخت ####

برای ارث‌بری از کلاس‌های پایه در پایتون استفاده کنید؟ `super()` سوال:** چگونه می‌توانید از متد** .5

.برای فراخوانی متدهای کلاس پایه از کلاس‌های مشتق شده استفاده می‌شود `super()` **:پاسخ**

```python

class Base:

def __init__(self):

print("Base init")

class Derived(Base):

def __init__(self):

super().__init__()

print("Derived init")

```

را در پایتون پیاده‌سازی کنید؟ Singleton سوال:** چگونه می‌توانید الگوی** .6

پاسخ:** با استفاده از یک کلاس که تنها یک نمونه از آن ایجاد**

.را پیاده‌سازی کنید Singleton شود، می‌توانید الگوی

```python

class Singleton:

_instance = None

def __new__(cls, *args, **kwargs):

if cls._instance is None:

cls._instance = super(Singleton, cls).__new__(cls, *args, **kwargs)

return cls._instance

```

```

5. توابع و ماژول‌ها

سطح آسان

1. **سوال:** چگونه می‌توانید یک تابع تعریف کنید و آن را صدا بزنید؟**

پاسخ:

```python
def my_function(param):
    return param * 2

result = my_function(5)
```

2. **سوال:** چگونه می‌توانید یک ماژول را در پایتون وارد کنید؟**

پاسخ:

```python
import math
```

سطح متوسط

3. **سوال:** توضیح دهید که چگونه می‌توانید توابع با تعداد متغیرهای ورودی متغیر (varargs) را تعریف کنید.

پاسخ: با استفاده از `*args` و `**kwargs` می‌توانید توابعی با تعداد متغیرهای ورودی متغیر تعریف کنید.

```python
def func(*args, **kwargs):
    print(args)
    print(kwargs)
```

4. **سوال:** چگونه می‌توانید از ماژول‌های خارجی استفاده کنید و آن‌ها را نصب کنید؟**

پاسخ: از `pip` برای نصب ماژول‌های خارجی استفاده می‌شود.

```bash
pip install requests
```

سطح سخت ####

سوال:** چگونه می‌توانید یک تابع داخلی را از یک ماژول شخصی به یک ماژول دیگر صادر کنید؟** .5

.می‌توانید توابع داخلی را صادر کنید `__init__.py` پاسخ:** با استفاده از**

```python

# mymodule/__init__.py

from .submodule import my_function

```

استفاده کنید؟ (unit tests) سوال:** چگونه می‌توانید عملکردهای یک ماژول را تست کنید و از تست‌های واحد** .6

.می‌توانید تست‌های واحد بنویسید `unittest` پاسخ:** با استفاده از ماژول**

```python

import unittest

class TestMyModule(unittest.TestCase):

def test_function(self):

self.assertEqual(my_function(2), 4)

```

مدیریت خطا و استثنا .6 ###

سطح آسان ####

سوال:** چگونه می‌توانید یک استثنا را در پایتون شناسایی کنید و آن را مدیریت کنید؟** .1

.می‌توانید استثناها را شناسایی و مدیریت کنید `except` و `try` پاسخ:** با استفاده از**

```python

try:

1 / 0

except ZeroDivisionError:

print("Cannot divide by zero")

```

سوال:** چگونه می‌توانید چندین نوع استثنا را با هم مدیریت کنید؟** 2.

.بلوک می‌توانید انواع مختلف استثناها را مدیریت کنید `except` پاسخ:** با استفاده از چندین**

```python

try:

# some code

except (TypeError, ValueError):

print("TypeError or ValueError occurred")

```

سطح متوسط

سوال:** چگونه می‌توانید استثنای سفارشی ایجاد کنید؟** 3.

.ارث‌بری می‌کند، می‌توانید استثنای سفارشی ایجاد کنید `Exception` پاسخ:** با ایجاد یک کلاس جدید که از**

```python

class CustomError(Exception):

pass

```

استفاده کنید؟ `try-except` را در بلوک‌های `finally` سوال:** چگونه می‌توانید** 4.

.برای اجرای کدهایی است که باید در هر صورت اجرا شوند، حتی اگر استثنا رخ دهد `finally` **:پاسخ**

```python

try:

# some code

finally:

print("This will always execute")

```

سطح سخت

سوال:** چگونه می‌توانید در پایتون خطاهای زمان اجرا را در قالب یک گزارش دقیق ثبت کنید؟** 5.

.می‌توانید خطاها را با جزئیات ثبت کنید `logging` پاسخ:** با استفاده از ماژول**

```python

import logging
```

```
logging.basicConfig(filename='app.log', level=logging.ERROR)

try:

1 / 0

except ZeroDivisionError as e:

logging.error("Exception occurred", exc_info=True)
```

6. **سوال:** توضیح دهید که چگونه می‌توانید با استفاده از `contextlib` یک context manager سفارشی ایجاد کنید.

پاسخ: با استفاده از `contextlib.contextmanager` می‌توانید یک context manager سفارشی ایجاد کنید.

```python
from contextlib import contextmanager

@contextmanager

def my_context():

print("Entering context")

try:

yield

finally:

print("Exiting context")
```

---

7. عملیات بر روی فایل‌ها

سطح آسان

1. **سوال:** چگونه می‌توانید یک فایل متنی را در پایتون بخوانید؟

پاسخ:

```python
with open('file.txt', 'r') as file:

content = file.read()
```

```

سوال:** چگونه می‌توانید متنی به یک فایل متنی اضافه کنید؟** .2

**:پاسخ**

```python
with open('file.txt', 'a') as file:
    file.write('New content')
```

سطح متوسط

سوال:** چگونه می‌توانید محتویات یک فایل را به صورت خط به خط بخوانید؟** .3

:پاسخ

```python
with open('file.txt', 'r') as file:
    lines = file.readlines()
```

سوال:** چگونه می‌توانید یک فایل باینری را در پایتون بنویسید؟** .4

:پاسخ

```python
with open('file.bin', 'wb') as file:
    file.write(b'\x00\x01\x02')
```

سطح سخت

سوال:** چگونه می‌توانید با استفاده از `csv` ماژول داده‌ها را به یک فایل CSV بنویسید؟** .5

:پاسخ

```python
import csv

with open('file.csv', 'w', newline='') as file:
    writer = csv.writer(file)
```
```

writer.writerow(['Name', 'Age'])

writer.writerow(['Alice', 30])

```

بنویسید و از آن بخوانید؟ JSON ماژول داده‌ها را به یک فایل `json` سوال:** چگونه می‌توانید با استفاده از** .6

**:پاسخ**

```python

import json

data = {'name': 'Alice', 'age': 30}

with open('file.json', 'w') as file:

json.dump(data, file)

with open('file.json', 'r') as file:

loaded_data = json.load(file)

```

مدیریت حافظه و کارایی .8 ###

سطح آسان ####

سوال:** چگونه می‌توانید یک لیست را در پایتون حذف کنید؟** .1

:پاسخ

```python

my_list = [1, 2, 3]

del my_list

```

ماژول برای مدیریت حافظه استفاده کنید؟ `gc` سوال:** چگونه می‌توانید از** .2

.می‌توانید جمع‌آوری زباله‌ها را به صورت دستی انجام دهید `gc` پاسخ:** با استفاده از ماژول**

```python

```
import gc

gc.collect()
```

سطح متوسط

3. **سوال:** چگونه می‌توانید حافظه را در پایتون با استفاده از ابزارهای پروفایلینگ اندازه‌گیری کنید؟**
پاسخ: با استفاده از ماژول‌هایی مانند `memory_profiler` می‌توانید مصرف حافظه را پروفایل کنید.

```python
from memory_profiler import profile

@profile
def my_function():
    # code
```

4. **سوال:** چگونه می‌توانید از `timeit` برای اندازه‌گیری زمان اجرای کد استفاده کنید؟
پاسخ:

```python
import timeit

execution_time = timeit.timeit('code_to_test()', setup='from __main__ import code_to_test', number=1000)
```

سطح سخت

5. **سوال:** چگونه می‌توانید با استفاده از `tracemalloc` برای تجزیه و تحلیل مصرف حافظه استفاده کنید؟
پاسخ: با استفاده از `tracemalloc` می‌توانید مصرف حافظه را در طول اجرای برنامه بررسی کنید.

```python
import tracemalloc

tracemalloc.start()

# code
```

```
snapshot = tracemalloc.take_snapshot()

top_stats = snapshot.statistics('lineno')

for stat in top_stats[:10]:

print(stat)
```

سوال:** چگونه می‌توانید حافظه غیر ضروری را در پایتون آزاد کنید و کارایی برنامه را بهینه کنید؟** .6

.پاسخ:** با استفاده از مدیریت بهینه منابع و حذف مراجع به اشیاء غیر ضروری می‌توانید حافظه را آزاد کنید و کارایی را بهینه کنید**

---

9. تست

و دیباگ

سطح آسان

ساده بنویسید؟ (unit test) سوال:** چگونه می‌توانید یک تست واحد** .1

.می‌توانید تست‌های واحد بنویسید `unittest` پاسخ:** با استفاده از ماژول**

```python
import unittest

class TestMyFunction(unittest.TestCase):

def test_addition(self):

self.assertEqual(1 + 1, 2)
```

برای بررسی شرایط استفاده کنید؟ `assert` سوال:** چگونه می‌توانید از** .2

:پاسخ

```python
assert 1 + 1 == 2
```

سطح متوسط ####

برای نوشتن و اجرای تست‌ها استفاده کنید؟ `pytest` سوال:** چگونه می‌توانید از** .3

.می‌توانید تست‌ها را با استفاده از توابع ساده و بدون نیاز به کلاس‌ها بنویسید `pytest` پاسخ:** با استفاده از**

```python
def test_addition():
    assert 1 + 1 == 2
```

برای شبیه‌سازی اشیاء در تست‌ها استفاده کنید؟ `mock` سوال:** چگونه می‌توانید از** .4

:پاسخ

```python
from unittest.mock import Mock

mock_object = Mock()
mock_object.method.return_value = 'value'
```

سطح سخت ####

میزان پوشش تست‌های خود را اندازه‌گیری کنید؟ `coverage` سوال:** چگونه می‌توانید با استفاده از** .5

.می‌توانید میزان پوشش کد را اندازه‌گیری کنید `coverage` پاسخ:** با استفاده از ابزار**

```bash
coverage run -m unittest discover
coverage report
```

را مدیریت کنید؟ (breakpoints) برای دیباگ کردن کد استفاده کنید و نقاط توقف `pdb` سوال:** چگونه می‌توانید از** .6

.می‌توانید نقاط توقف را تنظیم و کد را به صورت خط به خط اجرا کنید `pdb` پاسخ:** با استفاده از ماژول**

```python
import pdb; pdb.set_trace()
```

---

10. کتابخانه‌های استاندارد و شخصی

سطح آسان

1. **سوال:** چگونه می‌توانید از کتابخانه `math` برای محاسبه ریشه مربع استفاده کنید؟

پاسخ:

```python
import math
result = math.sqrt(16)
```

2. **سوال:** چگونه می‌توانید از کتابخانه `datetime` برای گرفتن تاریخ و زمان فعلی استفاده کنید؟

پاسخ:

```python
from datetime import datetime
now = datetime.now()
```

سطح متوسط

3. **سوال:** چگونه می‌توانید از کتابخانه `requests` برای ارسال درخواست HTTP استفاده کنید؟

پاسخ:

```python
import requests
response = requests.get('https://example.com')
```

4. **سوال:** چگونه می‌توانید از کتابخانه `beautifulsoup4` برای استخراج داده‌ها از HTML استفاده کنید؟

پاسخ:

```python
from bs4 import BeautifulSoup
soup = BeautifulSoup('<html></html>', 'html.parser')
```

```

سطح سخت ####

تبدیل کنید و تحلیل‌های ابتدایی انجام دهید؟ DataFrame داده‌ها را به یک `pandas` سوال:** چگونه می‌توانید با استفاده از** .5

**:پاسخ**

```python

import pandas as pd

data = {'Name': ['Alice', 'Bob'], 'Age': [30, 25]}

df = pd.DataFrame(data)

```

برای انجام عملیات ماتریسی و جبر خطی استفاده کنید؟ `numpy` سوال:** چگونه می‌توانید با استفاده از** .6

:پاسخ

```python

import numpy as np

matrix1 = np.array([[1, 2], [3, 4]])

matrix2 = np.array([[5, 6], [7, 8]])

result = np.dot(matrix1, matrix2)

:از سخت به آسان همراه با پاسخ‌ها آماده شده است Django در اینجا سوالات مصاحبه

**سوالات سخت** ###

را توضیح دهید و تفاوت‌های آنها چیست؟ Django در `prefetch_related` و `select_related` سوال:** نحوه استفاده از** .1

**:پاسخ**

- در سطح پایگاه داده برای مدل‌های مرتبط به کار می‌رود. این متد به خصوص برای مدل‌هایی join برای بهینه‌سازی کوئری‌ها با استفاده از `select_related`
.استفاده می‌کنند مفید است `ForeignKey` که از

- مناسب است. این متد به خصوص برای مدل‌هایی Python برای بهینه‌سازی کوئری‌ها با استفاده از دو کوئری جداگانه و ترکیب نتایج در `prefetch_related`
.استفاده می‌کنند مفید است `reverse ForeignKey` یا `ManyToManyField` که از

**:مثال**

```python

# select_related
```

```
books = Book.objects.select_related('author').all()

# prefetch_related
authors = Author.objects.prefetch_related('books').all()
```

2. **سوال:** چگونه می‌توانید الگوهای سفارشی در Django ایجاد کنید و از آنها استفاده کنید؟

پاسخ:

برای ایجاد الگوهای سفارشی در URL در Django، می‌توانید از ماژول `re_path` یا `path` در `urls.py` الگوهای مورد نظر خود را تعریف کنید و استفاده کنید.

مثال:

```python
from django.urls import path
from . import views

urlpatterns = [
path('article/<int:id>/', views.article_detail, name='article_detail'),
]
```

3. **سوال:** چگونه می‌توانید یک مدل Django را برای استفاده از پایگاه داده‌های چندگانه پیکربندی کنید؟

پاسخ:

در کوئری‌های خود برای مشخص کردن پایگاه داده‌های مختلف را پیکربندی کنید و از Django برای استفاده از پایگاه داده‌های چندگانه، باید در تنظیمات `using` پایگاه داده استفاده کنید.

مثال:

```python
# تنظیمات در settings.py
DATABASES = {
'default': {
'ENGINE': 'django.db.backends.postgresql',
'NAME': 'default_db',
```

},

'other': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'other_db',

},

}

استفاده از پایگاه داده در کوئری #

MyModel.objects.using('other').all()

```

پیاده‌سازی کنید؟ Django سوال:** چگونه می‌توانید عملکردها را به صورت همزمان در** .4

**:پاسخ**

.برای مدیریت کارهای پس‌زمینه استفاده کنید RabbitMQ/Redis و Celery برای پیاده‌سازی عملکرد همزمان، می‌توانید از

**:مثال**

- **Celery:** تنظیم**

```python

# tasks.py

from celery import shared_task

@shared_task

def my_task(param):

عملکردی که به صورت همزمان اجرا می‌شود #

```

چیست و هر کدام در چه مواقعی باید استفاده شود؟ Django در `Form` و `ModelForm` سوال:** تفاوت‌های بین** .5

:پاسخ

.به شما امکان می‌دهد فرم‌هایی را برای مدل‌ها بسازید که به طور خودکار فیلدها و اعتبارسنجی‌ها را از مدل به ارث می‌برند `ModelForm` -

.برای فرم‌های سفارشی که ممکن است به مدل خاصی مرتبط نباشند یا نیاز به فیلدهای سفارشی دارند، مناسب است `Form` -

:مثال
```

```python
# ModelForm
from django.forms import ModelForm
from .models import MyModel

class MyModelForm(ModelForm):
class Meta:
model = MyModel
fields = ['field1', 'field2']

# Form
from django import forms

class CustomForm(forms.Form):
field1 = forms.CharField(max_length=100)
field2 = forms.EmailField()
```

سوالات متوسط**

6. **سوال:** چگونه می‌توانید از سیگنال‌ها در Django استفاده کنید و برای چه مقاصدی از آنها بهره می‌برید؟**

پاسخ:
سیگنال‌ها به شما این امکان را می‌دهند که به رویدادهای خاصی مانند ذخیره یا حذف یک مدل پاسخ دهید. از سیگنال‌ها می‌توانید برای انجام کارهایی مانند ارسال ایمیل پس از ذخیره یک شی استفاده کنید.

مثال:
```python
from django.db.models.signals import post_save
from django.dispatch import receiver

@receiver(post_save, sender=MyModel)
def my_callback(sender, instance, **kwargs):
```

عملکرد مورد نظر پس از ذخیره مدل #

```

محدودیت‌های سفارشی تعریف کنید؟ Django سوال:** چگونه می‌توانید برای یک مدل** .7

**:پاسخ**

.در مدل برای تعریف محدودیت‌های سفارشی استفاده کنید که اعتبارسنجی‌های پیچیده‌تر را انجام دهد `clean` می‌توانید از متد

**:مثال**

```python

from django.core.exceptions import ValidationError

class MyModel(models.Model):

field1 = models.IntegerField()

def clean(self):

if self.field1 < 0:

raise ValidationError('field1 must be non-negative')

```

استفاده کنید و برای چه مقاصدی می‌توانید از آن بهره ببرید؟ Django در Middleware سوال:** چگونه می‌توانید از** .8

:پاسخ

برای انجام کارهایی مانند Middleware پردازش کنید. می‌توانید از Django به شما این امکان را می‌دهد که فرآیند درخواست و پاسخ را در Middleware .احراز هویت، ثبت‌نام، و پردازش درخواست‌ها استفاده کنید

:مثال

```python

# custom_middleware.py

from django.utils.deprecation import MiddlewareMixin

class CustomMiddleware(MiddlewareMixin):

def process_request(self, request):

# پردازش درخواست قبل از رسیدن به view

pass

def process_response(self, request, response):

پردازش پاسخ قبل از ارسال به مرورگر #

return response

```

برای اضافه کردن فیلدهای سفارشی استفاده کنید؟ Django Admin سوال:** چگونه می‌توانید از** .9

:پاسخ

.برای افزودن فیلدهای سفارشی به صفحه ادمین استفاده کنید و تنظیمات خود را در آن قرار دهید `ModelAdmin` می‌توانید از کلاس

:مثال

```python

from django.contrib import admin

from .models import MyModel

class MyModelAdmin(admin.ModelAdmin):

list_display = ('field1', 'field2')

search_fields = ('field1',)

admin.site.register(MyModel, MyModelAdmin)

```

چیست؟ Django سوال:** روش‌های مختلف برای بهینه‌سازی عملکرد در** .10

:پاسخ

.برای بهینه‌سازی کوئری‌ها `prefetch_related` و `select_related` استفاده از -

.استفاده از کش برای ذخیره نتایج محاسبات سنگین -

.استفاده از بانک‌های داده‌ای بهینه و مناسب برای نیازهای شما -

.و استفاده از ایندکس‌ها SQL بهینه‌سازی کوئری‌های -

سوالات آسان ###
```

را راه‌اندازی کنید و اولین اپلیکیشن را ایجاد کنید؟ Django سوال:** چگونه یک پروژه** .11

:پاسخ

.ایجاد کنید `django-admin startproject projectname` را با استفاده از دستور Django ابتدا پروژه -

.اپلیکیشن جدیدی بسازید `python manage.py startapp appname` سپس وارد دایرکتوری پروژه شوید و با دستور -

ایجاد کنید و آن را در پایگاه داده ذخیره کنید؟ Django سوال:** چگونه می‌توانید یک مدل ساده در** .12

:پاسخ

آن را در `python manage.py migrate` و `python manage.py makemigrations` تعریف کنید و سپس با دستور `models.py` یک مدل را در - .پایگاه داده ذخیره کنید

:مثال

```python
from django.db import models


class MyModel(models.Model):
    name = models.CharField(max_length=100)
```

چیست؟ Django سوال:** نحوه ایجاد و استفاده از یک فرم ساده در** .13

:پاسخ

.خود استفاده کنید view تعریف کنید و آن را در `forms.py` فرم را در -

:مثال

```python
from django import forms


class MyForm(forms.Form):
    name = forms.CharField(max_length=100)
```

متصل کنید؟ URL تعریف کنید و آن را به Django ساده در view سوال:** چگونه می‌توانید یک** .14

:پاسخ

.مورد نظر متصل کنید URL به `urls.py` تعریف کنید و سپس آن را در `views.py` را در view یک -

:مثال

```python

# views.py

from django.http import HttpResponse

def my_view(request):

return HttpResponse('Hello, world!')

# urls.py

from django.urls import path

from .views import my_view

urlpatterns = [

path('hello/', my_view),

]

```

:در مصاحبه، از سوالات پایه‌ای تا پیشرفته، می‌توان به دسته‌بندی‌های زیر توجه کرد Django برای پوشش 100 سوال از

سوالات سخت

1. **سوال:** نحوه استفاده از `select_related` و `prefetch_related` در Django چیست؟ را توضیح دهید و تفاوت‌های آنها

پاسخ:

- `select_related` از join برای بهینه‌سازی کوئری‌ها با استفاده از سطح پایگاه داده برای مدل‌های مرتبط به کار می‌رود.

- `prefetch_related` برای بهینه‌سازی کوئری‌ها با استفاده از دو کوئری جداگانه و ترکیب نتایج در Python مناسب است.

2. **سوال:** چگونه می‌توانید یک مدل Django را برای استفاده از پایگاه داده‌های چندگانه پیکربندی کنید؟

پاسخ:

- در تنظیمات Django از `using` پایگاه داده‌های مختلف را پیکربندی کنید و از در کوئری‌های خود برای مشخص کردن پایگاه داده استفاده کنید.

3. **سوال:** تفاوت‌های بین `Form` و `ModelForm` در Django چیست و هر کدام در چه مواقعی باید استفاده شود؟

:پاسخ

- `ModelForm` برای فرم‌هایی است که به طور خودکار فیلدها و اعتبارسنجی‌ها را از مدل به ارث می‌برند.

- `Form` برای فرم‌های سفارشی که ممکن است به مدل خاصی مرتبط نباشند یا نیاز به فیلدهای سفارشی دارند، مناسب است.

4. **سوال:** چگونه می‌توانید از سیگنال‌ها در Django استفاده کنید و برای چه مقاصدی از آنها بهره می‌برید؟**

:پاسخ

- سیگنال‌ها به شما این امکان را می‌دهند که به رویدادهای خاصی مانند ذخیره یا حذف یک مدل پاسخ دهید.

5. **سوال:** چگونه می‌توانید عملکردها را به صورت همزمان در Django پیاده‌سازی کنید؟**

:پاسخ

- برای مدیریت کارهای پس‌زمینه استفاده کنید RabbitMQ/Redis و Celery برای پیاده‌سازی عملکرد همزمان، می‌توانید از.

6. **سوال:** نحوه استفاده از `django-debug-toolbar` را توضیح دهید و چه زمانی باید از آن استفاده کنید؟**

:پاسخ

- `django-debug-toolbar` برای اشکال‌زدایی و تحلیل عملکرد کوئری‌های SQL و زمان‌بندی درخواست‌ها استفاده می‌شود.

7. **سوال:** چگونه می‌توانید فیلترهای سفارشی برای Django Admin ایجاد کنید؟**

:پاسخ

- می‌توانید فیلترهای سفارشی اضافه کنید `RelatedOnlyFieldListFilter` و `SimpleListFilter` با استفاده از کلاس‌های.

8. **سوال:** چگونه می‌توانید مدل‌های Django را برای استفاده از پایگاه داده‌های متنوع پیکربندی کنید؟**

:پاسخ

- در کوئری‌ها و تنظیم پایگاه‌های داده در تنظیمات پروژه `using` با استفاده از گزینه.

9. **سوال:** مفهوم `DBRouter` در Django چیست و چگونه می‌توان از آن استفاده کرد؟**

:پاسخ

- `DBRouter` برای مسیریابی درخواست‌های پایگاه داده به پایگاه داده‌های مختلف بر اساس مدل‌های خاص یا اپلیکیشن‌ها استفاده می‌شود.

10. **سوال:** چگونه می‌توانید یک الگوی URL سفارشی برای view های Django ایجاد کنید؟**

:پاسخ

- الگوهای URL می‌توانید `urls.py` در `re_path` یا `path` با استفاده از سفارشی ایجاد کنید.

استفاده کنید؟ Django در پروژه CORS برای مدیریت `django-cors-headers` سوال:** چگونه می‌توانید از** 11.

:پاسخ

.برای کنترل دسترسی به منابع از دامنه‌های مختلف `django-cors-headers` با نصب و پیکربندی -

استفاده کنید؟ view برای سازماندهی کدهای `Class-based views` چگونه می‌توانید از Django سوال:** در** 12.

:پاسخ

.views برای مدیریت رفتارهای مختلف در `UpdateView` و `CreateView`, `ListView`, `DetailView` با استفاده از کلاس‌هایی مانند -

چیست؟ Django در `Session` و `Request` سوال:** تفاوت‌های بین** 13.

:پاسخ

برای ذخیره اطلاعات کاربر در طول چندین `Session` به داده‌های مربوط به درخواست کاربر در هر درخواست اشاره دارد، در حالی که `Request` - .درخواست استفاده می‌شود

انجام دهید؟ Django در `F` و `Q` سوال:** چگونه می‌توانید کوئری‌های پیچیده را با استفاده از** 14.

:پاسخ

.برای انجام عملیات‌های محاسباتی بر روی فیلدها `F` برای انجام جستجوهای پیچیده و ترکیب شروط و از `Q` با استفاده از -

را انجام دهید؟ Django سوال:** چطور می‌توانید کار با توکن‌ها و مدیریت دسترسی در** 15.

:پاسخ

.و پیکربندی توکن‌ها برای مدیریت احراز هویت و مجوزها `django-rest-framework` با استفاده از -

چیست و چه زمانی باید از آنها استفاده کرد؟ Django در `Custom Managers` سوال:** نحوه استفاده از** 16.

:پاسخ

.برای افزودن متدهای سفارشی به کوئری‌ها و انجام عملیات‌های خاص بر روی مدل‌ها -

چیست و چگونه می‌توان از آن برای ایجاد سیستم‌های احراز هویت سفارشی استفاده کرد؟ Django در `AbstractBaseUser` سوال:** مفهوم** 17.

:پاسخ

.پایه‌ای برای ایجاد مدل‌های کاربری سفارشی فراهم می‌کند که می‌توانید ویژگی‌های خاص خود را به آن اضافه کنید `AbstractBaseUser` -

چیست؟ Django برای ارسال ایمیل‌های فعال‌سازی در `Signals` سوال:** نحوه استفاده از** 18.

:پاسخ

.برای ارسال ایمیل به کاربر پس از ثبت‌نام `post_save` با استفاده از سیگنال‌های -

19. **سوال:** چگونه می‌توانید با استفاده از `Django Channels` و WebSockets برنامه‌های بلادرنگ را پیاده‌سازی کنید؟**

پاسخ:

- برای ارتباط دوطرفه WebSockets برای ایجاد برنامه‌های بلادرنگ و استفاده از `Django Channels` با استفاده از.

20. **سوال:** نحوه استفاده از `Django Rest Framework` برای ساخت APIهای RESTful چیست؟**

پاسخ:

- برای ایجاد و مدیریت serializers و viewsets با استفاده از APIهای RESTful.

سوالات متوسط

21. **سوال:** چگونه می‌توانید از `ModelForm` برای ایجاد و مدیریت فرم‌های مدل استفاده کنید؟**

پاسخ:

- برای ساخت فرم‌هایی که به طور خودکار فیلدها و اعتبارسنجی‌ها را از مدل به ارث می‌برند `ModelForm` با استفاده از.

22. **سوال:** نحوه تعریف و استفاده از `Custom Fields` در Django چیست؟**

پاسخ:

- با تعریف فیلدهای سفارشی که می‌توانند متدهای اضافی و اعتبارسنجی‌های خاص خود را داشته باشند.

23. **سوال:** چگونه می‌توانید از `Middleware` برای پردازش درخواست و پاسخ استفاده کنید؟**

پاسخ:

- را پیاده‌سازی می‌کند `process_response` و `process_request` که متدهای Middleware با ایجاد یک کلاس.

24. **سوال:** نحوه استفاده از `Django Signals` برای پردازش خودکار رویدادها چیست؟**

پاسخ:

- با استفاده از سیگنال‌ها برای اجرای کدهایی در پاسخ به رویدادهایی مانند ذخیره یا حذف مدل.

25. **سوال:** چگونه می‌توانید با استفاده از `Django Debug Toolbar` عملکرد کوئری‌ها و درخواست‌ها را تحلیل کنید؟**

پاسخ:

- و زمان‌بندی درخواست‌ها SQL برای مشاهده و تحلیل کوئری‌های `django-debug-toolbar` با نصب و پیکربندی.

26. **سوال:** نحوه تعریف `custom user models` در Django چیست و چرا باید از آنها استفاده کنید؟**

:پاسخ

.برای ایجاد مدل‌های کاربری سفارشی و افزودن ویژگی‌های خاص `BaseUserManager` و `AbstractBaseUser` با استفاده از -

27. **سوال:** چگونه می‌توانید از `Django Admin` برای سفارشی‌سازی رابط کاربری ادمین استفاده کنید؟

:پاسخ

.برای سفارشی‌سازی نمایش و مدیریت مدل‌ها در رابط کاربری ادمین `ModelAdmin` با استفاده از کلاس‌های -

28. **سوال:** نحوه پیاده‌سازی `pagination` در Django چیست؟

:پاسخ

.برای تقسیم نتایج کوئری‌ها به صفحات و نمایش تعداد معین از اشیاء در هر صفحه `Paginator` با استفاده از -

29. **سوال:** چگونه می‌توانید از `Django Templates` برای سفارشی‌سازی و نمایش داده‌ها استفاده کنید؟

:پاسخ

.HTML برای نمایش داده‌ها و اجرای منطق ساده در Django و تگ‌های قالب HTML با استفاده از قالب‌های -

Django معماری به عنوان یک چارچوب مبتنی بر معماری **MVC (Model-View-Controller)** یا همان **MTV (Model-Template-View)**
:وجود دارد که در ادامه به آن پرداخته می‌شود MVC برخی تفاوت‌های ظاهری با معماری سنتی Django عمل می‌کند. با این حال، در

Django: در MTV (Model-Template-View) معماری

1. **Model (مدل):**

نقش:** بخش داده‌ای برنامه است و به تعامل با پایگاه داده می‌پردازد. مدل‌ها ساختار داده‌ها را تعریف می‌کنند و شامل فیلدها و متدهایی هستند که مرتبط با ** -
.اطلاعات برنامه می‌باشند

استفاده می‌کنند. این یعنی شما می‌توانید ORM (Object Relational Mapping) کاربرد:** مدل‌ها به طور مستقیم با جداول پایگاه داده مطابقت دارند و از** -
.به تعامل با پایگاه داده بپردازید SQL بدون نیاز به نوشتن کوئری‌های

:مثال -

```python
class Product(models.Model):
name = models.CharField(max_length=100)
price = models.DecimalField(max_digits=10, decimal_places=2)
description = models.TextField()
```

2. **Template (قالب):**

"Template" به آن Django شناخته می‌شود، اما در "View" ، این بخش به عنوانMVC در معماری سنتی .HTML نقش:** نمایش داده‌ها به کاربر از طریق** - .می‌گویند

ها برای نمایش داده‌ها به شکل وب‌سایت استفاده می‌شوند. آنها می‌توانند شامل منطق نمایشی ساده مانند حلقه‌ها، شروط، و فیلترها باشند، Template **:کاربرد** - .اما نباید شامل منطق پیچیده برنامه‌نویسی باشند

:مثال -

```html

<h1>{{ product.name }}</h1>

<p>{{ product.description }}</p>

<p>Price: ${{ product.price }}</p>

```

3. **View (ویو):**

.نقش:** دریافت درخواست‌ها از کاربر، پردازش داده‌ها (احتمالاً از مدل) و ارسال داده‌ها به قالب برای نمایش** -

به کاربر (JSON یا HTML مثلاً) وظیفه دریافت درخواست از مرورگر کاربر، اعمال منطق برنامه‌نویسی لازم و ارسال پاسخ Django در View **:کاربرد** - .شناخته می‌شود "Controller" به عنوان MVC را دارد. این بخش در معماری

:مثال -

```python

from django.shortcuts import render

from .models import Product


def product_detail(request, product_id):

product = Product.objects.get(id=product_id)

return render(request, 'product_detail.html', {'product': product})

```

انواع معماری در Django:**

1. **Monolithic Architecture (معماری یکپارچه):**

قرار دارد Django در این نوع معماری، تمام بخش‌های برنامه به هم متصل هستند و یک واحد بزرگ تشکیل می‌دهند. به عنوان مثال، کل برنامه در یک پروژه - .ها به طور مستقیم در همان پروژه تعریف می‌شوندview و تمام اپلیکیشن‌ها، مدل‌ها و

:ویژگی‌ها -

سادگی پیاده‌سازی -

مناسب برای پروژه‌های کوچک -

پیچیدگی بیشتر در نگهداری پروژه‌های بزرگ -

2. **Modular Architecture (معماری ماژولار):**

- **ویژگی‌ها:**

قابلیت استفاده مجدد از اپلیکیشن‌ها -

توسعه و نگهداری آسان‌تر برای پروژه‌های بزرگ -

قابلیت توسعه پروژه به مرور زمان -

3. **Microservices Architecture (معماری میکروسرویس‌ها):**

- **ویژگی‌ها:**

جداسازی کامل بخش‌ها -

توانایی مقیاس‌پذیری هر سرویس به صورت مستقل -

پیچیدگی در مدیریت ارتباط بین سرویس‌ها -

مقایسه معماری‌های مختلف:**

معماری	**مزایا**	**معایب**
Monolithic	ساده برای پروژه‌های کوچک، توسعه سریع	مشکل در نگهداری و مقیاس‌پذیری پروژه‌های بزرگ
Modular	قابلیت استفاده مجدد از اپلیکیشن‌ها، مقیاس‌پذیری بهتر	نیاز به مدیریت پیچیدگی ارتباط بین اپلیکیشن‌ها
Microservices	توسعه و دیپلوی مستقل، مقیاس‌پذیری عالی	پیچیدگی بیشتر در ارتباط بین سرویس‌ها

توضیح MVC و MTV در Django:**

- **MVC (Model-View-Controller):** در این معماری، "Model" داده‌ها را مدیریت می‌کند، "View" داده‌ها را به کاربر نمایش می‌دهد و "Controller" را مدیریت می‌کند model و view منطق اصلی برنامه و تعامل بین.

- **MTV (Model-Template-View):** در Django، "Model" همچنان مسئول مدیریت داده‌ها است، اما "View" وظیفه نمایش داده‌ها را دارد و "Template" همچنان مسئول مدیریت داده‌ها است، اما "Model"، Django در MVC که در) به عنوان لایه‌ی کنترلی Controller به آن می‌شود (گفته عمل می‌کند.

:جمع‌بندی ###

به همراه پاسخ‌های آنها آورده شده است که می‌تواند برای Django REST Framework ، و(JWT (JSON Web Token، API در زیر سوالات دقیقی در مورد مصاحبه‌های مرتبط با این مفاهیم مفید باشد:

چیست؟ (Application Programming Interface) API .1 ###

:پاسخ

API ها اغلب به عنوان یک سرویس مبتنی برAPIیک رابط یا واسط است که به دو برنامه نرم‌افزاری اجازه می‌دهد با یکدیگر ارتباط برقرار کنند. در حوزه وب، ارائه می‌شوند که به یک برنامه اجازه می‌دهد به داده‌ها یا منطق برنامه‌های دیگر دسترسی پیدا کند HTTP.

چیست؟ SOAP API و RESTful API تفاوت بین .2 ###

:پاسخ

- **RESTful API:** ها URI شامل سبک بودن، استفاده از REST برای ارسال درخواست‌ها و دریافت پاسخ‌ها استفاده می‌کند. اصول طراحی HTTP از پروتکل است (DELETE، PUT، POST، GET مانند) HTTP برای شناسایی منابع، و استفاده از روش‌های.

- **SOAP API:** برای فرمت پیام‌ها استفاده می‌کند و قابلیت‌هایی مانند امنیت و انتقال پیام‌های پیچیده را فراهم می‌کند XML پروتکل رسمی‌تری است که از.

چیست و چه کاربردی دارد؟ (JSON Web Token) JWT .3 ###

:پاسخ

شامل سه JWT .یک استاندارد است برای ایجاد توکن‌های ایمن که به کاربران اجازه می‌دهد بدون نیاز به ارسال مکرر اطلاعات ورود، احراز هویت شوند JWT بخش است:

1. **Header:** مشخصات الگوریتم رمزنگاری

2. **Payload:** اطلاعات کاربر و سایر داده‌ها

3. **Signature:** بخش رمزنگاری شده برای تایید اعتبار

های مبتنی بر وب استفاده می‌شودAPI به طور معمول برای احراز هویت در JWT.

استفاده می‌شود؟ Session به جای JWT چرا از .4 ###

:پاسخ

:مزایایی دارد Session نسبت به JWT

.، تمام اطلاعات احراز هویت در توکن قرار دارد و نیاز به ذخیره‌سازی سرور نداردJWT عدم نیاز به ذخیره‌سازی سرور:** در** -

.برای اپلیکیشن‌هایی با مقیاس بالا که نیاز به توزیع بار دارند مناسب است JWTقابلیت مقیاس‌پذیری بیشتر:** به دلیل مستقل بودن از سرور، ** -

.در سمت سرور نیست Session می‌فرستد، بنابراین نیازی به مدیریت JWT کلاینت اطلاعات کاربر را به همراه **:session بی‌نیازی از مدیریت** -

---

چیست؟ Django REST Framework روش‌های مختلف احراز هویت در .5 ###

:پاسخ

:از چندین روش احراز هویت پشتیبانی می‌کند Django REST Framework

.کار می‌کند session ID برای اپلیکیشن‌های مبتنی بر مرورگر استفاده می‌شود و بر پایه‌ی **:SessionAuthentication** -

.به آنها اختصاص می‌دهد API از توکن‌های ساده‌ای استفاده می‌کند که سرور برای احراز هویت کاربران در **:TokenAuthentication** -

.برای ایجاد توکن‌های امن جهت احراز هویت استفاده می‌کند JWT از **:JWT Authentication** -

.رمزگذاری شده ارسال می‌کند base64 نام کاربری و رمز عبور را در هر درخواست به صورت **:BasicAuthentication** -

---

وجود دارد؟ JWT Authentication و Token Authentication چه تفاوتی بین .6 ###

:پاسخ

در این روش، یک توکن تصادفی به کاربر اختصاص داده می‌شود و در سرور ذخیره می‌گردد. در هر درخواست، این توکن **:Token Authentication** -
.بررسی شده و کاربر احراز هویت می‌شود

را در هر JWT ، اطلاعات کاربر در توکن رمزنگاری می‌شود و نیازی به ذخیره‌سازی در سرور نیست. کلاینتJWT در **:JWT Authentication** -
.درخواست ارسال می‌کند و سرور فقط توکن را اعتبارسنجی می‌کند

پیاده‌سازی می‌کنید؟ Django REST Framework را در JWT چگونه .7 ###

:پاسخ

:استفاده کرد **djangorestframework-simplejwt** ، می‌توان از کتابخانه‌های خارجی مانندDjango REST Framework در JWT برای پیاده‌سازی

:نصب بسته .1

```bash
pip install djangorestframework-simplejwt
```

2. **افزودن تنظیمات JWT در settings.py:**

```python
REST_FRAMEWORK = {
'DEFAULT_AUTHENTICATION_CLASSES': (
'rest_framework_simplejwt.authentication.JWTAuthentication',
),
}
```

3. **ایجاد توکن:**

برای ایجاد یک توکن JWT، می‌توان از ویوهای پیش‌ساخته استفاده کرد:

```python
from rest_framework_simplejwt.views import TokenObtainPairView, TokenRefreshView

urlpatterns = [
path('api/token/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
path('api/token/refresh/', TokenRefreshView.as_view(), name='token_refresh'),
]
```

---

8. **چگونه JWT را اعتبارسنجی می‌کنید؟**

پاسخ:

برای اعتبارسنجی JWT، سرور بخش **Signature** را با استفاده از کلید مخفی بررسی می‌کند. اگر توکن تغییر داده شده باشد یا زمان اعتبار آن گذشته باشد، اعتبار توکن رد می‌شود.

---

9. **چرا از Refresh Token در JWT استفاده می‌شود؟**

:پاسخ

به کاربران اجازه می‌دهد که توکن دسترسی خود را تمدید کنند بدون اینکه مجبور باشند دوباره لاگین کنند. این ویژگی امنیت بیشتری ایجاد می‌کند Refresh Token جایگزین شود Refresh Token زیرا توکن دسترسی عمر کوتاهی دارد و در صورت منقضی شدن می‌تواند به راحتی با.

دارد؟ APIView چیست و چه تفاوتی با ViewSet، Django REST Framework در .10 ###

:پاسخ

به عنوان HTTP را مدیریت کنید. هر متد DELETE ، وPUT، POST، GET مانند HTTP ویویی است که به شما اجازه می‌دهد تا درخواست‌های **:APIView** - یک متد جداگانه در کلاس تعریف می‌شود.

، **()list** می‌تواند با استفاده از روش‌هایی مانند ViewSet .را مدیریت می‌کند CRUD یک کلاس انتزاعی‌تر است که عملیات‌های استاندارد **:ViewSet** - به طور خودکار ویوها را ایجاد کند **()update** و **()create**، **()retrieve**.

استفاده می‌شود؟ Django REST Framework در Serializer چرا از .11 ###

:پاسخ

داده‌های مدل Serializerو همچنین اعتبارسنجی داده‌ها استفاده می‌شود. در واقع، (XML یا JSON مانند) برای تبدیل داده‌ها از و به فرمت‌های قابل نمایش Serializer تبدیل می‌کند API را به فرمت‌های مورد نیاز برای.

چیست؟ Serializer و ModelSerializer تفاوت بین .12 ###

:پاسخ

یک کلاس عمومی برای تبدیل داده‌های ورودی به فرمت مورد نظر و اعتبارسنجی آنها است. شما باید به صورت دستی تمام فیلدها و متدها را **:Serializer** - تعریف کنید.

تبدیل می‌کند. این کلاس Serializer را به Django است که به صورت خودکار فیلدهای یک مدل Serializer یک کلاس مشتق از **:ModelSerializer** - ساده‌سازی زیادی برای کار با مدل‌ها انجام می‌دهد.

ایجاد می‌کنید؟ Django REST Framework در pagination های مبتنی برAPI چگونه .13 ###

:پاسخ

`settings.py` ، می‌توان از ویژگی‌های داخلی فریمورک استفاده کرد. برای مثال، می‌توان در Django REST Framework در pagination برای پیاده‌سازی pagination تنظیمات را به شکل زیر افزود:

```python
REST_FRAMEWORK = {
'DEFAULT_PAGINATION_CLASS': 'rest_framework.pagination.PageNumberPagination',
'PAGE_SIZE': 10,
}
```

.این کد هر درخواست را به 10 مورد در هر صفحه محدود می‌کند

---

14. **چگونه از Throttling در Django REST Framework استفاده می‌کنید؟**

پاسخ:

می‌توانید throttling در یک بازه زمانی خاص را محدود کنید. برای پیاده‌سازی IP به شما اجازه می‌دهد که تعداد درخواست‌های مجاز از یک کاربر یا Throttling ، تنظیمات زیر را در `settings.py` اضافه کنید:

```python
REST_FRAMEWORK = {
'DEFAULT_THROTTLE_CLASSES': [
'rest_framework.throttling.UserRateThrottle',
'rest_framework.throttling.AnonRateThrottle',
],
'DEFAULT_THROTTLE_RATES': {
'user': '100/day',
'anon': '10/hour',
}
}
```

.این تنظیمات اجازه می‌دهد که کاربران ثبت‌نام شده 100 درخواست در روز و کاربران ناشناس 10 درخواست در هر ساعت ارسال کنند

---

15. چگونه از Versioning در Django REST Framework استفاده می‌کنید؟

پاسخ:

برای پیاده‌سازی versioning در Django REST Framework ابتدا نوع versioning را در `settings.py` مشخص کنید:

```python
REST_FRAMEWORK = {
'DEFAULT_VERSIONING_CLASS': 'rest_framework.versioning.URLPathVersioning',
}
```

سپس می‌توانید در مسیرهای URL نسخه API را تعریف کنید:

```python
urlpatterns = [
path('v1/products/', include('products.urls')),
path('v2/products/', include('new_products.urls')),
]
```

این تنظیمات به شما امکان مدیریت نسخه‌های مختلف API را می‌دهد.

در ادامه، مجموعه‌ای از سوالات مرتبط با کلاس‌بیس (Class-Based Views) در جنگو (Django) آورده شده است، که می‌تواند برای مصاحبه‌های جنگو مفید باشد:

---

1. Class-Based View (CBV) چیست و چرا از آن استفاده می‌کنیم؟

پاسخ:

Class-Based View (CBV) یک رویکرد شیگرا برای نوشتن ویوها در جنگو است. به جای تعریف ویوها به صورت تابعی (Function-Based Views یا FBV)، می‌توان از کلاس‌ها برای تعریف رفتار ویوها استفاده کرد. این روش اجازه می‌دهد تا از قابلیت‌های شیگرایی مانند وراثت، ماژولار بودن و قابلیت استفاده مجدد استفاده کنیم.

---

2. تفاوت اصلی بین Class-Based Views (CBV) و Function-Based Views (FBV) چیست؟

پاسخ:

.ویوها به صورت تابعی تعریف می‌شوند و به طور مستقیم به درخواست‌ها پاسخ می‌دهند. ساده‌تر و مناسب برای ویوهای کوچک و ساده است **FBV:** -

است. این روش برای (DELETE، PUT، POST، GET) ویوها به عنوان کلاس تعریف می‌شوند و هر متد کلاس مسئول مدیریت یک نوع درخواست **CBV:** - .پروژه‌های بزرگ و پیچیده به دلیل قابلیت ماژولار بودن و استفاده مجدد مناسب‌تر است

در جنگو چیست؟ CBV مهم‌ترین کلاس‌های پایه‌ای برای .3 ###

:پاسخ

:در جنگو عبارتند از CBV برخی از مهم‌ترین کلاس‌های پایه‌ای برای

.کلاس پایه برای تمامی ویوها **View:** -

.ویویی که صرفاً یک قالب را رندر می‌کند **TemplateView:** -

.ویویی که یک لیست از اشیا را نمایش می‌دهد **ListView:** -

.ویویی که جزئیات یک شیء خاص را نمایش می‌دهد **DetailView:** -

.ویویی که برای ایجاد یک شیء جدید استفاده می‌شود **CreateView:** -

.ویویی که برای به‌روزرسانی یک شیء استفاده می‌شود **UpdateView:** -

.ویویی که برای حذف یک شیء استفاده می‌شود **DeleteView:** -

استفاده کنیم؟ TemplateView چه زمانی باید از .4 ###

:پاسخ

زمانی استفاده می‌شود که نیازی به پردازش خاصی روی داده‌ها نداریم و فقط می‌خواهیم یک قالب را رندر کنیم. برای مثال، می‌توان از آن `TemplateView` از .برای نمایش صفحات ساده مانند صفحه درباره ما، تماس با ما، یا صفحات استاتیک دیگر استفاده کرد

استفاده می‌شود؟ ListView در چه مواردی از .5 ###

:پاسخ

زمانی استفاده می‌شود که نیاز داریم یک لیست از اشیا (مانند لیست محصولات، لیست مقالات) را به کاربر نمایش دهیم. این کلاس به صورت خودکار با `ListView` .استفاده از مدل مرتبط، تمام اشیا را از پایگاه داده بازیابی کرده و در قالب نمایش می‌دهد

ایجاد کنیم؟ DetailView چگونه می‌توانیم یک ویو شخصی‌سازی شده با استفاده از کلاس .6 ###

:پاسخ

:برای نمایش جزئیات یک شیء خاص استفاده می‌شود. شما می‌توانید از آن به شکل زیر برای یک مدل خاص استفاده کنید `DetailView`

```python
from django.views.generic import DetailView
from .models import Article

class ArticleDetailView(DetailView):
model = Article
template_name = 'article_detail.html'
context_object_name = 'article'
```

.نمایش داده می‌شود `Article` در این مثال، جزئیات یک شیء

را شخصی‌سازی کنیم؟ CBV چگونه می‌توانیم رفتار یک کلاس .7 ###

:پاسخ

را تغییر دهیم، GET (بازنویسی) متدهای پیش‌فرض کلاس انجام شود. برای مثال، اگر بخواهیم رفتار متد override می‌تواند از طریق CBV شخصی‌سازی رفتار یک :می‌توانیم آن را در کلاس بازنویسی کنیم

```python
class CustomView(View):
def get(self, request, *args, **kwargs):
# رفتار جدید برای GET
return HttpResponse("Hello, World!")
```

.کاربرد دارد `form_valid()` و `post()`، `get()` این روش برای تغییر رفتار متدهای پیش‌فرض مانند

ها استفاده کنیم؟mixin ها چگونه می‌توانیم ازCBV در .8 ###

:پاسخ

برای احراز هویت کاربر قبل از دسترسی به `LoginRequiredMixin`فراهم می‌کنند. برای مثال، CBV ها کلاس‌هایی هستند که قابلیت‌های خاصی را برایMixin اضافه کنیم CBV ها را به شکل زیر بهmixin ویو استفاده می‌شود. می‌توانیم:

```python
from django.contrib.auth.mixins import LoginRequiredMixin
from django.views.generic import TemplateView

class DashboardView(LoginRequiredMixin, TemplateView):
    template_name = 'dashboard.html'
    login_url = '/login/'
```

.در این مثال، اگر کاربر لاگین نکرده باشد، به صفحه لاگین هدایت می‌شود

---

9. چگونه می‌توانیم یک فرم را با استفاده از CreateView ایجاد کنیم؟

پاسخ:

برای ایجاد فرم‌ها و ذخیره داده‌های جدید در پایگاه داده استفاده می‌شود. به عنوان مثال، برای ایجاد یک فرم برای مدل `CreateView`، کد زیر استفاده می‌شود:

```python
from django.views.generic.edit import CreateView
from .models import Post

class PostCreateView(CreateView):
    model = Post
    template_name = 'post_form.html'
    fields = ['title', 'content']
    success_url = '/posts/'
```

.هدایت می‌شود `/posts/` در این مثال، فرم برای ایجاد یک پست جدید نمایش داده می‌شود و پس از موفقیت به

چگونه می‌توانیم یک فرم را اعتبارسنجی کنیم؟ CBV در .10 ###

:پاسخ

:استفاده کنیم و رفتار پیش‌فرض آن را تغییر دهیم. برای مثال `form_valid()` ، می‌توانیم از متدCBV برای اعتبارسنجی یک فرم در

```python
from django.views.generic.edit import FormView
from .forms import ContactForm

class ContactFormView(FormView):
    form_class = ContactForm
    template_name = 'contact.html'
    success_url = '/thank-you/'

    def form_valid(self, form):
        # کد اعتبارسنجی اضافی
        return super().form_valid(form)
```

.رفتار پیش‌فرض را فراخوانی کنیم `super()` در اینجا، می‌توانیم اعتبارسنجی خاصی را روی فرم اعمال کنیم و سپس با استفاده از

استفاده کنیم؟ UpdateView چه زمانی باید از .11 ###

:پاسخ

زمانی استفاده می‌شود که نیاز داریم یک شیء موجود را به‌روزرسانی کنیم. این کلاس به صورت خودکار فرم‌های لازم برای ویرایش شیء را `UpdateView` از .فراهم می‌کند و تغییرات را پس از اعتبارسنجی به پایگاه داده ارسال می‌کند

برای حذف یک شیء استفاده می‌شود؟ DeleteView چگونه از .12 ###

:پاسخ

:برای حذف یک شیء خاص از پایگاه داده استفاده می‌شود. به عنوان مثال `DeleteView`

```python
from django.views.generic.edit import DeleteView
```

```
from .models import Post

class PostDeleteView(DeleteView):
model = Post
template_name = 'post_confirm_delete.html'
success_url = '/posts/'
```

.حذف می‌شود و کاربر به صفحه‌ی لیست پست‌ها هدایت می‌شود `Post` در اینجا، وقتی کاربر تأیید کند، شیء

سفارشی استفاده کنیم؟ context data ها از CBV چگونه در .13 ###

:پاسخ

:استفاده کرد `get_context_data()` ، می‌توان از متد CBV یک `context` برای اضافه کردن داده‌های سفارشی به

```python
class CustomView(TemplateView):
template_name = 'custom_template.html'

def get_context_data(self, **kwargs):
context = super().get_context_data(**kwargs)
context['custom_data'] = 'This is custom data'
return context
```

.این کد به شما اجازه می‌دهد تا داده‌های دلخواه را به قالب ارسال کنید

استفاده کرد؟ middleware ها از CBV آیا می‌توان در .14 ###

:پاسخ

ها Middleware به صورت مستقیم نیستند. اما می‌توان از CBV اعمال می‌شوند و مربوط به کلاس‌های Django ها به صورت کلی برای کل برنامهMiddleware برای افزودن `LoginRequiredMixin` های خاصی مانندmixin برای مدیریت درخواست‌ها و پاسخ‌ها در سطوح مختلف برنامه استفاده کرد. همچنین می‌توان از .رفتارهای خاص مانند احراز هویت استفاده کرد

توصیه نمی‌شود؟ CBV چه زمانی استفاده از .15 ###

:پاسخ

منطقی‌تر FBV بیش از حد پیچیده شود، ممکن است استفاده از CBV اگر یک ویو بسیار ساده باشد یا نیاز به عملکردی بسیار خاص داشته باشد که پیاده‌سازی آن در .در پروژه‌های بزرگ و پیچیده که نیاز به استفاده مجدد و ماژولار بودن دارند، بیشتر مفید است CBV .باشد

در پایتون به ترتیب از آسان به سخت همراه با پاسخ‌ها آورده شده **(Object-Oriented Programming) در ادامه، سوالات مربوط به **برنامه‌نویسی شیگرا :است

چیست؟ (Object-Oriented Programming) شیگرایی .1 ###

:پاسخ

شیگرایی یک پارادایم برنامه‌نویسی است که به وسیله آن، برنامه‌ها به صورت مجموعه‌ای از اشیاء تعریف می‌شوند. این اشیاء می‌توانند داده‌ها و توابعی که روی این .داده‌ها عمل می‌کنند را در خود نگهداری کنند. هدف شیگرایی، فراهم کردن ساختاری منظم و ماژولار برای کدنویسی است

در پایتون چیست؟ (Object) و شیء (Class) کلاس .2 ###

:پاسخ

.برای ایجاد اشیاء است blueprint نقشه یا قالبی است که ویژگی‌ها (متغیرها) و رفتارها (توابع) را تعریف می‌کند. به عبارتی، کلاس :**(Class) کلاس** -

.نمونه‌ای از یک کلاس است. شیء دارای ویژگی‌ها و رفتارهایی است که در کلاس تعریف شده‌اند :**(Object) شیء** -

:مثال

```python

class Dog:

def __init__(self, name):

self.name = name

def bark(self):
```

print(f"{self.name} says woof!")

my_dog = Dog("Buddy")

my_dog.bark() # خروجی: Buddy says woof!

```

---

### 3. **در پایتون چیست؟ (Constructor) سازنده**

**پاسخ:**

سازنده یک متد ویژه در کلاس‌ها است که هنگام ایجاد یک شیء جدید از کلاس فراخوانی می‌شود. در پایتون، سازنده با نام `__init__` تعریف می‌شود.

مثال:

```python

class Person:

def __init__(self, name, age):

self.name = name

self.age = age

person = Person("Alice", 30)

```

---

4. **در پایتون چیست؟ (Inheritance) ارث‌بری**

پاسخ:

ارث‌بری مکانیزمی است که به کلاس‌ها این امکان را می‌دهد که ویژگی‌ها و متدهای کلاس‌های دیگر را به ارث ببرند. با استفاده از ارث‌بری می‌توان کد را مجدداً استفاده کرده و ساختار سلسله‌مراتبی از کلاس‌ها ایجاد کرد.

مثال:

```python

class Animal:

def speak(self):

print("Animal speaks")

class Dog(Animal):

def bark(self):

print("Dog barks")

dog = Dog()

dog.speak()  # خروجی: Animal speaks

dog.bark()  # خروجی: Dog barks

```

---

5. کپسوله‌سازی (Encapsulation) چیست؟

پاسخ:

کپسوله‌سازی مکانیزمی است که در آن داده‌ها و توابع مرتبط درون یک کلاس بسته‌بندی می‌شوند و دسترسی به این داده‌ها از خارج کلاس محدود می‌شود. این کار معمولاً با استفاده از متغیرهای خصوصی و متدهای عمومی انجام می‌شود.

مثال:

```python

class Car:

def __init__(self, make, model):

self.make = make

self.__model = model  # متغیر خصوصی

def get_model(self):

return self.__model

car = Car("Toyota", "Corolla")

print(car.get_model())  # خروجی: Corolla

# print(car.__model)  # این خط باعث خطا می‌شود

```

چیست؟ (Polymorphism) چندریختی .6 ###

:پاسخ

چندریختی به این معنی است که یک متد می‌تواند به اشکال مختلف در کلاس‌های مختلف پیادسازی شود. این امکان را می‌دهد که متدهای مشابه در کلاس‌های مختلف داشته باشیم و بر اساس نوع شیء، متد مناسب فراخوانی شود.

مثال:

```python

class Bird:

def fly(self):

print("Bird flies")

class Airplane:

def fly(self):

print("Airplane flies")

def make_it_fly(flyable):

flyable.fly()

bird = Bird()

plane = Airplane()

make_it_fly(bird)  # خروجی: Bird flies

make_it_fly(plane)  # خروجی: Airplane flies

```

در پایتون چیست و چه کاربردی دارند؟ (Special Methods) متدهای ویژه .7 ###
```

پاسخ:

متدهای ویژه در پایتون متدهایی هستند که با دو زیرخط (`__`) در ابتدا و انتهای نامشان تعریف می‌شوند. این متدها برای انجام عملیات‌های خاصی مانند تعریف عملگرهای ریاضی، مقایسه و تبدیل داده‌ها استفاده می‌شوند.

مثال:

```python
class Point:
def __init__(self, x, y):
self.x = x
self.y = y

def __add__(self, other):
return Point(self.x + other.x, self.y + other.y)

def __str__(self):
return f"Point({self.x}, {self.y})"

p1 = Point(1, 2)
p2 = Point(3, 4)
p3 = p1 + p2
print(p3)  # خروجی: Point(4, 6)
```

---

8. **چه تفاوت‌هایی بین وراثت تک‌سویه و چندسویه وجود دارد؟**

پاسخ:

- **وراثت تک‌سویه (Single Inheritance)**: در این نوع وراثت، یک کلاس تنها از یک کلاس دیگر ارث می‌برد.
- **وراثت چندسویه (Multiple Inheritance)**: در این نوع وراثت، یک کلاس می‌تواند از چندین کلاس دیگر ارث ببرد.

مثال از وراثت چندسویه:

```python
```

```
class A:
def method_a(self):
print("Method A")

class B:
def method_b(self):
print("Method B")

class C(A, B):
pass

c = C()
c.method_a()  # خروجی: Method A
c.method_b()  # خروجی: Method B
```

---

9. **چه تفاوت‌هایی بین متدهای کلاس و متدهای شیء وجود دارد؟**

پاسخ:

- **متدهای شیء (Instance Methods)**: این متدها برای دسترسی به ویژگی‌های شیء و تغییر آنها استفاده می‌شوند و به شیء خاصی که متد روی آن فراخوانی شده است وابسته‌اند.
- **متدهای کلاس (Class Methods)**: این متدها با استفاده از دکوراتور `@classmethod` تعریف می‌شوند و به کلاس به طور کلی دسترسی دارند. آنها به داده‌های کلاس (متغیرهای کلاسی) دسترسی دارند و معمولاً برای ساخت نمونه‌های جدید از کلاس استفاده می‌شوند.

مثال:

```python
class MyClass:
class_variable = "class var"

def __init__(self, value):
self.instance_variable = value
```

def instance_method(self):

print(self.instance_variable)

@classmethod

def class_method(cls):

print(cls.class_variable)

obj = MyClass("instance var")

obj.instance_method() # خروجی: instance var

MyClass.class_method() # خروجی: class var

```

---

### **10. چگونه می‌توان از متدهای استاتیک (Static Methods) استفاده کرد و چه کاربردی دارند؟**

**پاسخ:**

متدهای استاتیک با استفاده از دکوراتور `@staticmethod` تعریف می‌شوند و مانند توابع عادی هستند که به کلاس و شیء خاصی وابسته نیستند. این متدها برای انجام عملیات‌هایی که به داده‌های کلاس یا شیء نیازی ندارند، مناسب هستند.

مثال:

```python

class Math:

@staticmethod

def add(x, y):

return x + y

result = Math.add(5, 3)

print(result)  # خروجی: 8

```

```

چیست؟ `__repr__` و `__str__`، `__init__` مفهوم و کاربرد متدهای .11 ###

:پاسخ

.سازنده‌ای است که هنگام ایجاد یک شیء جدید از کلاس فراخوانی می‌شود :**`__init__`** -

.متدی است که برای بازگرداندن نمای رشته‌ای خوانا از شیء استفاده می‌شود و معمولاً در عملیات چاپ یا نمایش داده می‌شود :**`__str__`** -

.متدی است که برای بازگرداندن نمای رسمی و دقیق از شیء استفاده می‌شود و معمولاً برای دیباگ و توسعه استفاده می‌شود :**`__repr__`** -

:مثال

```python

class Person:

def __init__(self, name):

self.name = name

def __str__(self):

return f"Person(name={self.name})"

def __repr__(self):

return f"Person('{self.name}')"

p = Person("Alice")

print(str(p))  # خروجی: Person(name=Alice)

print(repr(p))  # خروجی: Person('Alice')

```

در پایتون چیست؟ `super()` مفهوم و کاربرد .12 ###

:پاسخ

برای دسترسی به متدهای کلاس والد از درون یک کلاس فرزند استفاده می‌شود. این تابع به ویژه در وراثت چندگانه و در هنگام `super()` تابع

.استفاده از متدهای والد مهم است

مثال:

```python
class Parent:
def __init__(self):
print("Parent init")

class Child(Parent):
def __init__(self):
super().__init__()
print("Child init")

c = Child()
# خروجی:
# Parent init
# Child init
```

در پایتون چگونه می‌توان از کلاس‌های خصوصی و محافظت‌شده استفاده کرد؟ .13 ###

پاسخ:

کلاس‌های خصوصی و محافظت‌شده با استفاده از دو زیرخط (`__`) و یک زیرخط (`_`) تعریف می‌شوند. کلاس‌هایی که با دو زیرخط شروع می‌شوند خصوصی هستند و به طور مستقیم از خارج از کلاس قابل دسترسی نیستند. کلاس‌هایی که با یک زیرخط شروع می‌شوند محافظت‌شده هستند و معمولاً از آن‌ها به عنوان قسمت داخلی و خصوصی کلاس استفاده می‌شود.

مثال:

```python
class MyClass:
def __init__(self):
self._protected_var = "protected"
self.__private_var = "private"

def get_private_var(self):
```

return self.__private_var

obj = MyClass()

print(obj._protected_var) # خروجی: protected

print(obj.__private_var) # این خط باعث خطا می‌شود

print(obj.get_private_var()) # خروجی: private
```

---

### 14. **مفهوم Composition و Aggregation؟ چه تفاوت‌هایی دارند؟ در شی‌گرایی چیست و چه**

**پاسخ:**

- **Composition**: زمانی که یک شیء یکی دیگر از اشیاء را به عنوان بخشی از خود در نظر می‌گیرد و شیء وابسته (جزئی) نمی‌تواند به صورت مستقل وجود داشته باشد. به عبارتی، شیء جزئی در طول عمر شیء اصلی ایجاد و حذف می‌شود.

- **Aggregation**: مشابه Composition است اما شیء وابسته می‌تواند به صورت مستقل وجود داشته باشد و در طول عمر شیء اصلی به‌وجود آمده و حذف نشود.

مثال از Composition:

```python
class Engine:
def __init__(self):
print("Engine created")

class Car:
def __init__(self):
self.engine = Engine()  # Composition
print("Car created")

car = Car()
# خروجی:
# Engine created
```
```

Car created

```

مثال از Aggregation:

```python

class Driver:

def __init__(self, name):

self.name = name


class Car:

def __init__(self, driver):

self.driver = driver  # Aggregation

print("Car created with driver")


driver = Driver("Alice")

car = Car(driver)

خروجی: #

# Car created with driver

```

---

15. برای تغییر رفتار متدها استفاده کرد؟ decorators در پایتون، چگونه می‌توان از

پاسخ:

می‌توان decorators در پایتون توابعی هستند که به شما اجازه می‌دهند رفتار توابع یا متدها را به صورت داینامیک تغییر دهید. با استفاده از Decorators ویژگی‌های اضافی را به توابع یا متدها اضافه کرد.

مثال:

```python

def decorator_function(func):

def wrapper():

print("Something is happening before the function is called.")

```
    func()
    print("Something is happening after the function is called.")
    return wrapper

@decorator_function
def say_hello():
    print("Hello!")

say_hello()
# خروجی:
# Something is happening before the function is called.
# Hello!
# Something is happening after the function is called.
```

در پایتون استفاده کرد و چه کاربردی دارد؟ `__call__` چگونه می‌توان از متد .16 ###

پاسخ:

در یک کلاس، می‌توانید رفتار شیء را به عنوان یک `__call__` به شما اجازه می‌دهد تا اشیاء کلاس‌ها را مانند توابع فراخوانی کنید. با تعریف متد `__call__` متد تابع تعریف کنید.

مثال:

```python
class CallableClass:
    def __call__(self, *args, **kwargs):
        print("Object called with arguments:", args, kwargs)

obj = CallableClass()
obj(1, 2, 3, key="value")  # خروجی: Object called with arguments: (1, 2, 3) {'key': 'value'}
```

چیست؟ `@classmethod` و `@staticmethod` تفاوت بین .17 ###

پاسخ:

- **`@staticmethod`**: متد استاتیک به کلاس وابسته نیست و هیچ دسترسی به متغیرهای کلاس یا شیء ندارد. این متدها می‌توانند مستقل از کلاس و شیء مورد استفاده قرار گیرند.

- **`@classmethod`**: متد کلاس به کلاس وابسته است و می‌تواند به متغیرهای کلاس دسترسی پیدا کند. اولین پارامتر متد کلاس معمولاً `cls` است که به کلاس مربوطه اشاره دارد.

مثال:

```python
class MyClass:
    class_variable = "class var"

    @staticmethod
    def static_method():
        print("Static method called")

    @classmethod
    def class_method(cls):
        print(f"Class method called with class variable: {cls.class_variable}")

MyClass.static_method()  # خروجی: Static method called
MyClass.class_method()  # خروجی: Class method called with class variable: class var
```

در پایتون برای مدیریت ویژگی‌های یک شیء استفاده کرد؟ `property` چگونه می‌توان از .18 ###

پاسخ:

`property` یک دکوراتور در پایتون است که به شما اجازه می‌دهد تا به صورت مستقیم ویژگی‌های خصوصی یک شیء را مدیریت کنید و از آنها مانند ویژگی‌های عمومی استفاده کنید. با استفاده از `property` می‌توانید getter، setter و deleter را تعریف کنید.

مثال:

```python
class Person:
def __init__(self, name):
self._name = name

@property
def name(self):
return self._name

@name.setter
def name(self, value):
if len(value) > 0:
self._name = value

@name.deleter
def name(self):
del self._name

p = Person("Alice")
print(p.name)  # خروجی: Alice
p.name = "Bob"
print(p.name)  # خروجی: Bob
del p.name
```

برای بررسی ترتیب رزولوشن متد در وراثت چندگانه استفاده کرد؟ `mro()` چگونه می‌توان از .19 ###

پاسخ:

برای بررسی ترتیب رزولوشن متدها در وراثت چندگانه استفاده می‌شود. این متد ترتیب کلاس‌ها را در زنجیره وراثت برمی‌گرداند که در آن متدها `mro()` متد جستجو می‌شوند.

:مثال

```python

class A:

pass


class B(A):

pass


class C(A):

pass


class D(B, C):

pass


print(D.mro())

# خروجی: [<class '__main__.D'>, <class '__main__.B'>, <class '__main__.C'>, <class '__main__.A'>, <class 'object'>]

```

**20. **برای فراخوانی متدهای کلاس‌های والد در وراثت چندگانه استفاده کرد؟ `super()` چگونه می‌توان از

:پاسخ

می‌توانید به متدهای `super()` به شما اجازه می‌دهد که متدهای کلاس‌های والد را به ترتیب مناسب فراخوانی کنید. با استفاده از `super()`در وراثت چندگانه، کلاس‌های والد در زنجیره وراثت دسترسی پیدا کنید.

:مثال

```python

class A:

def __init__(self):

print("A init")
```

```
class B(A):
def __init__(self):
super().__init__()
print("B init")

class C(A):
def __init__(self):
super().__init__()
print("C init")

class D(B, C):
def __init__(self):
super().__init__()
print("D init")

d = D()
خروجی: #
# A init
# C init
# B init
# D init
```