بسم الله الرحمن الرحیم

A.چرا Python زبان برنامهنویسی محبوب علم داده است؟

Python به دلایل زیر زبان برنامه‌نویسی محبوبی در علم داده است:

1. سادگی و خوانایی:

سینتاکس ساده و قابل فهم، یادگیری و نگهداری کد را آسان می‌سازد.

2. کتابخانه‌های متنوع :وجود کتابخانه‌های قدرتمند مانند Matplotlib، pandas، NumPy و Scikit-learn ابزارهای مناسبی برای تحلیل داده و یادگیری ماشین ارائه می‌دهد.

3. جامعه بزرگ: وجود یک جامعه فعال و بزرگ به کاربران این امکان را می‌دهد که به راحتی از تجربیات و منابع دیگران بهره‌مند شوند.

4. قابلیت‌های تحلیل داده: Python از قابلیت‌های پیشرفته تحلیل داده و بصری‌سازی داده‌ها برخوردار است.

5. *تناسب با علوم مختلف*: از علم داده گرفته تا یادگیری ماشین، هوش مصنوعی و وب، Python در زمینه‌های متنوعی کاربرد دارد.

این ویژگی‌ها باعث شده‌اند که Python به یکی از زبان‌های برتر در علم داده تبدیل شود.

B. NumPy و Pandas چه تفاوتی دارند؟

NumPy و Pandas هر دو کتابخانه‌های مهمی در Python برای کار با داده‌ها هستند، اما اهداف و قابلیت‌های متفاوتی دارند:

1. *Numpy*:

هدف: ارائه ابزارهای کار با آرایه‌ها و انجام محاسبات عددی.

ساختار داده: تمرکز بر روی آرایه‌های چند بعدی (ndarray).

عملکرد: بهینه‌سازی برای محاسبات ریاضی و علم داده، با سرعت بالا در عملیات عددی.

2. *Pandas*:

هدف: تحلیل و دستکاری داده‌های ساختار یافته.

-ساختار داده: فراهم‌آوری دو نوع ساختار داده اصلی: DataFrame (برای داده‌های جدولی) و Series (برای داده‌های یک بعدی).

عملکرد: ابزارهای پیشرفته برای جستجو، فیلتر کردن و تحلیل داده‌ها در قالب جدولی با قابلیت‌های آسان‌تر برای کار با داده‌های زمان‌سری و داده‌های گمشده.

به‌طور کلی، NumPy برای محاسبات عددی و Pandas برای تحلیل داده‌های ساختار یافته طراحی شده است.

C .چرا Matplotlib برای تجسم دادهها استفاده میشود؟

Matplotlib به دلایل زیر برای تجسم داده‌ها استفاده می‌شود:

1. *انعطاف‌پذیری*: امکان ایجاد انواع مختلف نمودارها و گراف‌ها را با جزئیات بالا فراهم می‌کند.
2. *سفارشی‌سازی*: قابلیت تنظیم و سفارشی‌سازی عناصر مختلف نمودارها مانند رنگ، فونت، برچسب‌ها و غیره.
3. *ادغام با کتابخانه‌های دیگر*: به خوبی با کتابخانه‌های دیگری مانند NumPy و Pandas ادغام می‌شود، که امکان تجسم مستقیم داده‌های پردازش شده را فراهم می‌کند.
4. *پشتیبانی گسترده*: دارای مستندات کامل و پشتیبانی جامعه کاربری بزرگی است که حل مشکلات و استفاده بهینه از آن را آسان می‌کند.
5. *سازگاری با محیط‌های مختلف* قابل استفاده در محیط‌های مختلف مانند اسکریپت‌های Python، محیط‌های تعاملی (مانند Jupyter Notebook) و برنامه‌های وب.

به طور خلاصه، Matplotlib به دلیل انعطاف‌پذیری بالا، قابلیت سفارشی‌سازی، ادغام با دیگر کتابخانه‌ها و پشتیبانی گسترده، یک ابزار قدرتمند برای تجسم داده‌ها است.

.Seabornچرا برای تجسم دادههای پیشرفته کاربرد دارد؟

Seaborn برای تجسم داده‌های پیشرفته به دلایل زیر کاربرد دارد:

1. *زیبایی بصری*: Seaborn بر اساس Matplotlib ساخته شده و نمودارهای جذاب‌تر و زیباتری را با تنظیمات پیش‌فرض بهینه ارائه می‌دهد.
2. نمودارهای آماری پیچیده: Seaborn امکان ایجاد نمودارهای آماری پیچیده مانند نمودارهای توزیع، نمودارهای رابطه‌ای و نمودارهای دسته‌بندی را فراهم می‌کند که درک عمیق‌تری از داده‌ها ارائه می‌دهند.
3. *یکپارچگی با Seaborn :*Pandas به خوبی با DataFrame های Pandas کار می‌کند، که امکان تجسم مستقیم داده‌های ساختاریافته را آسان می‌کند.

4. سادگی استفاده: با وجود پیچیدگی نمودارها، Seaborn رابط کاربری ساده‌تری برای ایجاد این نمودارها ارائه می‌دهد، به طوری که با چند خط کد می‌توان نمودارهای پیچیده‌ای ایجاد کرد.

5. *تمرکز بر روابط بین متغیرها*: Seaborn ابزارهای خاصی برای بررسی و تجسم روابط بین متغیرهای مختلف در یک مجموعه داده ارائه می‌دهد.

به طور خلاصه، Seaborn به دلیل زیبایی بصری، قابلیت ایجاد نمودارهای آماری پیچیده، یکپارچگی با Pandas و سادگی استفاده، ابزاری قدرتمند برای تجسم داده‌های پیشرفته است.

.چگونه میتوانید یک Function در Python تعریف کنید؟

در Python، یک تابع با استفاده از کلمه کلیدی `def` تعریف می‌شود، به این صورت:

```python
Def نام_تابع(پارامترها):
    """
    توضیحات (اختیاری)
    """
    # بدنه تابع (کد)
    Return مقدار_بازگشتی (اختیاری)
```

* `def`: کلمه کلیدی برای تعریف تابع
* `نام_تابع`: نامی که برای تابع انتخاب می‌کنید
* `پارامترها`: ورودی‌های تابع (اختیاری، می‌تواند خالی باشد)
* `"""توضیحات"""`: یک رشته توضیحی که عملکرد تابع را شرح می‌دهد (اختیاری) `return`: مقدار بازگشتی تابع (اختیاری)

F .چرا Comprehension List در Python استفاده میشود؟

List comprehension در Python به دلایل زیر استفاده می‌شود:

* *کوتاه‌تر و خواناتر:* جایگزینی فشرده‌تر و خواناتر برای ایجاد لیست‌ها با استفاده از حلقه‌ها.
* سرعت بیشتر:اغلب سریع‌تر از حلقه‌های `for` سنتی عمل می‌کند.
* کدنویسی پایتونیک:یک روش استاندارد و مرسوم برای ایجاد لیست‌ها در Python.

G.چگونه میتوانید یک file CSV را در Python خواند؟

برای خواندن فایل CSV در پایتون، از ماژول `csv` استفاده می‌کنیم:

```python
Import csv

:With open('your_file.csv', 'r') as file
    Reader = csv.reader(file)
    :For row in reader
        # پردازش هر سطر (row)
        Print(row)
```

به جای `your_file.csv` نام فایل CSV خود را قرار دهید.