1. چرا Python زبان برنامه‌نویسی محبوب علم داده است؟

زبان پایتون یکی از با ارز شتری ن و جالبترین زبا نها برای تجزیه و تحلیل دادهها است ومحبوبیت آن در دنیای تجزیه و تحلیل دادهها و علم داده، روز به روز در حال افزایش است. ازآنجا که پایتون یکی از انعطا فپذیرترین زبا نهای برنام هنویسی است، از همینرو مورد علاقه علم داده است. همچ نین، افرادی که میخواهند وارد دنیای علم داده شوند، پایتون را به بسیاری از زبا نهای برنام هنویسی دیگر ترجیح م یدهند. چرا که مجبور نیستند زمان ز یادی را برای یادگیری آن صرف کنند. همچنین، بستههایی در پایتون وجود دارد که بهطور خاص برای کارهای مشخصی بهطور خلاصه میتوان گفت، پایتون به .SciPy و NumPy ،pandas ، طراحی شدهاند، از جمله دلیل ویژگیهای که در زیر آ نها را فهرست کرد هایم، علم داده را به تسخیر خود در آورده است:

1. **سادگی:** پایتون ی کی از آسا نتری ن زبا نها برای شروع است. همچنین، این سادگی شما را از امکاناتی که به آ نها نیاز دارید محدود ن میکند.
2. **کتابخانهها و چارچوبها:** به دلیل محبوبیت، پایتون صدها کتابخانه و چارچوب مختلف را دارا است که به روند توسعه شما کمک بسیار زیادی میکند و در زمان بسیارصرف هجویی م یکنند. به عنوان ی ک دانش مند داده، متوجه میشوید که بسیاری از این کتابخان هها به علم داده و یادگیری ماشین متمرکز هستند.
3. **جامعهی عظیم:** یکی از دلایل مشهور بودن پایتون وجود یک جامعهی عظیم متشکل از مهندسان و دانشمندان داده است. ممکن است فکر کنید که این نباید یکی از دلایل اصلی انتخاب شما برای پایتون باشد، اما حقیقت برعکس اس ت. اگر از نظرات و **پشتیبانی** های متخصصان دیگر استفاده نکنید، مسیر یادگیری شما دشوار خواهد بود.
4. NumPy و Pandas چه تفاوتی دارند؟

NumPy و Pandas هر دو کتابخانه‌های پرکاربرد در پایتون هستند که برای تحلیل داده‌ها و محاسبات علمی به‌ویژه در کار با داده‌های عددی مورد استفاده قرار می‌گیرند. اما این دو کتابخانه تفاوت‌هایی در نحوه استفاده و کاربرد دارند. در اینجا تفاوت‌های اصلی میان آنها را بررسی می‌کنیم:

**۱. نوع داده‌ها و ساختار داده‌ها**

* **NumPy**:
  + **آرایه‌ها (Arrays)**: NumPy اصلی‌ترین ساختار داده‌ای خود را به نام ndarray (آرایه‌ی n-بعدی) دارد که برای کار با داده‌های عددی و ماتریس‌ها طراحی شده است.
  + مناسب برای انجام محاسبات عددی پیچیده بر روی داده‌های عددی.
* **Pandas**:
  + **DataFrame**: Pandas ساختار داده‌ای اصلی خود را به نام DataFrame دارد که مشابه یک جدول پایگاه داده یا یک شیت اکسل است و شامل ردیف‌ها و ستون‌ها است.
  + **Series**: همچنین Pandas دارای یک ساختار داده‌ای به نام Series است که مشابه یک ستون در DataFrame است.
  + مناسب برای کار با داده‌های جدولی، داده‌های برچسب‌دار، و داده‌های متنوع از انواع مختلف (عدد، رشته و غیره).

**۲. عملیات و محاسبات**

* **NumPy**:
  + به‌طور عمده برای انجام محاسبات عددی و عملیات ریاضی مانند جمع، ضرب، ضرب ماتریسی، جذر و توابع مثلثاتی استفاده می‌شود.
  + عملیات سریع بر روی آرایه‌ها انجام می‌شود.
* **Pandas**:
  + علاوه بر قابلیت‌های محاسباتی مشابه با NumPy، بیشتر برای تجزیه و تحلیل داده‌ها، تمیزکاری، و دستکاری داده‌های جدولی طراحی شده است.
  + قابلیت‌های بیشتری برای مدیریت داده‌های گمشده، فیلتر کردن داده‌ها، گروه‌بندی و تجزیه‌وتحلیل داده‌ها را داراست.

**۶. استفاده و کاربردها**

* **NumPy**:
  + برای انجام محاسبات علمی، تحلیل داده‌های عددی، پردازش تصویر، و مدل‌سازی ریاضی مناسب است.
* **Pandas**:
  + مناسب برای کار با داده‌های جدولی، تجزیه و تحلیل داده‌های برچسب‌دار، داده‌های مالی، تحلیل داده‌های زمان‌بندی، و همچنین پیش‌پردازش داده‌ها است.

1. چرا Matplotlib برای تجسم داده‌ها استفاده می‌شود؟

Matplotlib یکی از محبوب‌ترین کتابخانه‌های پایتون برای تجسم داده‌ها است که برای ایجاد نمودارها و گراف‌ها مورد استفاده قرار می‌گیرد. این کتابخانه به‌ویژه برای تجسم داده‌های علمی و آماری بسیار کاربردی است. دلایلی که باعث می‌شود Matplotlib برای تجسم داده‌ها بسیار مفید و پرکاربرد باشد، به شرح زیر است:

**۱. قابلیت‌های گسترده در تجسم داده‌ها**

* **Matplotlib** به شما این امکان را می‌دهد که انواع مختلفی از نمودارها را بسازید، از جمله:
  + **نمودارهای خطی (Line plots)**
  + **نمودارهای میله‌ای (Bar charts)**
  + **نمودارهای پراکندگی (Scatter plots)**
  + **نمودارهای دایره‌ای (Pie charts)**
  + **نمودارهای هیستوگرام (Histograms)**
  + **نمودارهای سه‌بعدی (3D plots)**
* این انعطاف‌پذیری به شما کمک می‌کند تا داده‌ها را به بهترین شکل ممکن تجزیه و تحلیل کنید.

**۲. سفارشی‌سازی و انعطاف‌پذیری بالا**

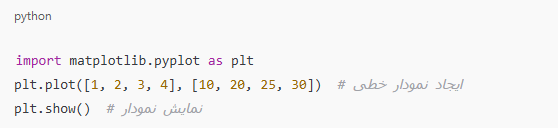
* با **Matplotlib** می‌توانید تمامی جزئیات نمودارها را سفارشی کنید:
  + تغییر رنگ‌ها
  + تنظیم عنوان، برچسب‌های محورها
  + تنظیم مقیاس‌ها
  + انتخاب فونت‌ها
  + استفاده از انواع مختلف خطوط و مارکرها
* این ویژگی‌ها به شما این امکان را می‌دهند که نمودارهایی دقیقا مطابق با نیازهای خاص خود ایجاد کنید.

**۳. یکپارچگی با دیگر کتابخانه‌ها**

* **Matplotlib** به راحتی با سایر کتابخانه‌های پایتون برای تجزیه و تحلیل داده‌ها، مانند **NumPy**، **Pandas**، و **Seaborn** یکپارچه می‌شود.
* این یکپارچگی به شما این امکان را می‌دهد که از قابلیت‌های تجزیه و تحلیل داده‌ها و سپس از **Matplotlib** برای تجسم آن‌ها استفاده کنید.

**۴. سهولت استفاده**

* حتی اگر تجربه زیادی در تجسم داده‌ها نداشته باشید، **Matplotlib** ابزارهای ساده‌ای برای ایجاد نمودارهای ابتدایی ارائه می‌دهد.
* برای مثال، تنها با چند خط کد می‌توانید یک نمودار خطی یا میله‌ای ساده ایجاد کنید:



**۵. گستردگی و پشتیبانی**

* **Matplotlib** یکی از قدیمی‌ترین کتابخانه‌های تجسم داده در پایتون است و بنابراین مستندات فراوان و جامعه کاربری بزرگی دارد.
* این کتابخانه به طور فعال توسعه می‌یابد و به روزرسانی‌های منظم دارد.

**۶. پشتیبانی از نمودارهای تعاملی**

* **Matplotlib** به کمک کتابخانه‌هایی مانند **mpl\_toolkits** و **plotly** امکان ایجاد نمودارهای تعاملی را فراهم می‌کند.
* با این قابلیت، کاربران می‌توانند نمودارها را به‌طور داینامیک و با تعامل بیشتر مشاهده کنند.

1. Seaborn چرا برای تجسم داده‌های پیشرفته کاربرد دارد؟

Seaborn یک کتابخانه‌ی سطح بالا در پایتون برای تجسم داده‌ها است که بر اساس Matplotlib ساخته شده و با pandas یکپارچه شده است. این کتابخانه برای تجسم داده‌های پیشرفته کاربرد دارد زیرا:

1. سادگی و خوانایی بالا

Seaborn به طور پیش‌فرض استایل‌های جذاب و حرفه‌ای دارد، که باعث خواناتر شدن نمودارها می‌شود.

بسیاری از توابع آن برای تحلیل داده‌های آماری طراحی شده‌اند و نیاز به کدنویسی پیچیده را کاهش می‌دهند.

2. یکپارچگی با Pandas

Seaborn مستقیماً با DataFrameهای pandas کار می‌کند و نیازی به تبدیل داده‌ها به فرمت‌های دیگر ندارد.می‌توان به راحتی ستون‌های DataFrame را برای محورهای x و y و دیگر ویژگی‌ها مشخص کرد.

3. پشتیبانی از تجسم‌های آماری پیچیده

توابع داخلی برای تجزیه‌وتحلیل توزیع داده‌ها (مانند sns.histplot()، sns.kdeplot() و sns.boxplot()).

رسم روابط بین متغیرها (مانند sns.scatterplot()، sns.pairplot() و sns.regplot()).

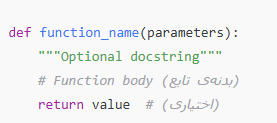
ایجاد نمودارهای دسته‌بندی‌شده و چندبعدی (مانند sns.violinplot() و sns.catplot()).

4. پشتیبانی از پالت‌های رنگی متنوع و قابل تنظیم

Seaborn دارای پالت‌های رنگی پیش‌فرضی است که برای نمایش بهتر داده‌ها مناسب‌اند.با استفاده از توابعی مانند sns.color\_palette() می‌توان پالت رنگی سفارشی‌سازی کرد.

1. چگونه می‌توانید یک Function در Python تعریف کنید؟

در پایتون، برای تعریف یک تابع (Function) از کلمه‌ی کلیدی def استفاده می‌کنیم. در ادامه ساختار کلی تعریف یک تابع و مثال‌هایی از آن آورده شده است:

ساختار کلی یک تابع در Python

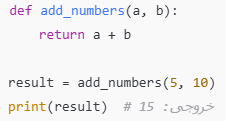
: function\_name: نام تابع (بهتر است توصیفی باشد).

 parameters: ورودی‌های تابع (در صورت نیاز).

 """docstring""": توضیحات اختیاری درباره‌ی عملکرد تابع.

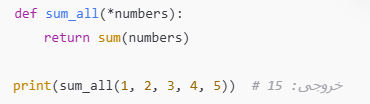
 return: مقدار بازگشتی (در صورت نیاز).

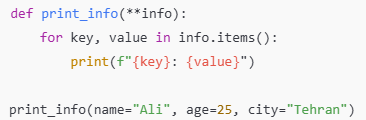
مثال ۱: تابع بدون ورودی و مقدار بازگشتی 



مثال ۲: تابع با ورودی و مقدار بازگشتی

مثال ۳: مقدار پیش‌فرض برای آرگومان‌ها 

مثال ۴: تابع با تعداد آرگومان نامشخص (\*args)

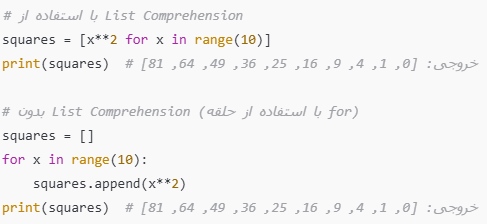
مثال ۵: تابع با آرگومان‌های کلیدی (\*\*kwargs)

1. چرا List Comprehension در Python استفاده می‌شود؟

List Comprehension در پایتون یک ویژگی قدرتمند است که به شما این امکان را می‌دهد که با استفاده از یک خط کد، لیست‌های جدید را از لیست‌های موجود ایجاد کنید. این روش به‌ویژه زمانی مفید است که بخواهید یک لیست جدید بسازید که بر اساس یک یا چند شرط از لیست‌های دیگر استخراج شده باشد.

دلایلی که List Comprehension در پایتون بسیار مورد استفاده قرار می‌گیرد، عبارتند از:

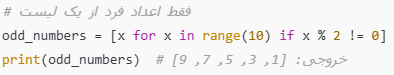
**۱. نوشتن کد کوتاه‌تر و خواناتر**

* **List Comprehension** کد را کوتاه‌تر و خواناتر می‌کند. به‌جای استفاده از حلقه‌های for پیچیده برای ایجاد لیست‌ها، می‌توان همه چیز را در یک خط نوشت.
* این روش نوشتن کد، به‌ویژه برای افرادی که با پایتون آشنا هستند، بسیار ساده و شفاف است.

۲. سرعت بیشتر

* List Comprehension به‌طور معمول سریع‌تر از استفاده از حلقه‌های for است. این موضوع به دلیل استفاده از نحوه‌ی بهینه‌تر پردازش داده‌ها در خود List Comprehension است.
* در پردازش‌های بزرگ، این تفاوت می‌تواند قابل‌توجه باشد.

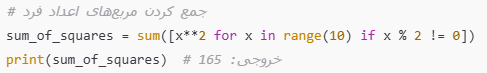
۳. استفاده از شرایط (Filters) به‌راحتی

* می‌توانید از List Comprehension برای اعمال شرط‌های مختلف و فیلتر کردن داده‌ها استفاده کنید. این ویژگی باعث می‌شود که بتوانید به‌سادگی از داده‌های غیرضروری صرف‌نظر کرده و تنها داده‌های مورد نظر خود را استخراج کنید

.

۴. خواندن و نوشتن کد در یک خط

* می‌توانید هم‌زمان عملیات روی داده‌ها و اعمال فیلترها را در یک خط انجام دهید. این کار نه‌تنها کد را مختصر می‌کند بلکه عملیات را در یک مرحله انجام می‌دهد.



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

* برای خواندن یک فایل CSV در پایتون، می‌توانید از کتابخانه‌های مختلفی مانند csv و pandas استفاده کنید. در اینجا دو روش متداول برای خواندن فایل‌های CSV را بررسی می‌کنیم:
* ۱. استفاده از کتابخانه csv
* کتابخانه‌ی csv یک کتابخانه داخلی در پایتون است که به شما امکان می‌دهد فایل‌های CSV را بخوانید و پردازش کنید.
* مثال:
* ۲. استفاده از کتابخانه pandas
* کتابخانه pandas یکی از قدرتمندترین ابزارها برای پردازش داده‌ها در پایتون است و به‌ویژه برای کار با داده‌های جدولی مثل CSV بسیار مناسب است. این روش به‌ویژه برای تحلیل داده‌ها و انجام پردازش‌های پیچیده‌تر بسیار مفید است.
* مثال:

۳. انتخاب روش مناسب

* کتابخانه csv برای کارهای ساده و زمانی که نیاز به پردازش پیچیده داده‌ها ندارید، گزینه خوبی است.
* کتابخانه pandas برای کار با داده‌های پیچیده‌تر و انجام تجزیه‌وتحلیل‌های آماری و عددی مناسب است.

1. JSON و XML چه تفاوتی دارند؟

* JSON (JavaScript Object Notation) و XML (eXtensible Markup Language) هر دو فرمت‌های محبوب برای تبادل داده‌ها بین سیستم‌ها هستند. این دو فرمت دارای تفاوت‌های قابل توجهی از نظر نحوه ساختاردهی داده‌ها، سادگی و کاربردها هستند. در اینجا تفاوت‌های اصلی میان JSON و XML آورده شده است:

۱. ساختار و قالب بندی



**۲**. اندازه و حجم

* JSON:
  + به طور معمول فایل‌های کوچک‌تر و سبک‌تر دارد.
  + فقط داده‌های مورد نیاز را شامل می‌شود و بنابراین حجم کمتری دارد.
* XML:
  + معمولاً فایل‌های بزرگ‌تر و سنگین‌تر دارد.
  + نیاز به استفاده از تگ‌ها برای هر عنصر داده دارد که باعث افزایش حجم می‌شود.

۳. خوانایی و سادگی

* JSON:
  + خواندن و نوشتن آن برای انسان‌ها بسیار ساده‌تر است.
  + ساختار آن بسیار کمپکت و بدون پیچیدگی است.
* XML:
  + خواندن و نوشتن آن ممکن است پیچیده‌تر باشد.
  + باید از تگ‌های مختلف استفاده کنید که ساختار را حجیم‌تر و پیچیده‌تر می‌کند.
* **نام استاد :مهندس احمدزاده**
* **نام درس :مباحث ویژه**
* **نام ونام خانوادگی :امیرحسین کریم خوی**