

**دانشگاه ملی مهارت**

**آموزشکده میناب**

**تمرینات بخش دوم**

نام و نام خانوادگی: هدا مهرانی پور\_مرضیه نجفی

واحد درسی : مباحث ویژه

رشته : مهندسی کامپیوتر

مدرس : محمد احمد زاده

بهمن ۱۴۰۳

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

### A. چرا Python زبان برنامه‌نویسی محبوب علم داده است؟

پایتون به دلیل ساختار شیء‌گرا، سینتکس بهینه و کتابخانه‌های پیشرفته مانند NumPy برای محاسبات عددی، Pandas برای تحلیل داده‌های ساختاریافته، و Scikit-learn برای مدل‌سازی آماری، در علم داده پیشرو است. انعطاف‌پذیری آن در پیاده‌سازی الگوریتم‌های پیچیده نظیر رگرسیون، خوشه‌بندی و شبکه‌های عصبی، همراه با پشتیبانی از بهینه‌سازی سطح پایین (مثل Cython)، آن را برای تحقیقات آکادمیک و صنعتی ایده‌آل می‌کند. جامعه‌ی علمی و مستندات جامع پایتون، امکان اعتبارسنجی و تکرارپذیری نتایج را تضمین کرده و استاندارد برای پروژه‌های داده‌محور فراهم می‌آورد.

### B. NumPy و Pandas چه تفاوتی دارند؟

کتابخانه‌ی NumPy یک ابزار پایه برای محاسبات عددی است که با ارائه‌ی آرایه‌های چندبعدی و توابع ریاضیاتی سریع (مانند عملیات ماتریسی)، عملکرد بالایی در پردازش داده‌ها دارد. در مقابل، Pandas برای تحلیل داده‌های ساختاریافته (مانند جداول و سری‌های زمانی) طراحی شده و با ساختارهای DataFrame و Series، امکان مدیریت و پردازش پیشرفته‌ی داده‌ها را فراهم می‌کند. به بیان دقیق، NumPy زیربنای محاسباتی را تشکیل می‌دهد، در حالی که Pandas لایه‌ای تخصصی برای داده‌کاوی و پیش‌پردازش روی آن اضافه می‌کند.

### C. چرا Matplotlib برای تجسم داده‌ها استفاده می‌شود؟

کتابخانه‌ی Matplotlib به دلیل توانایی بالای خود در تولید نمودارهای متنوع و باکیفیت (مانند خطی، پراکندگی، میله‌ای) و انعطاف‌پذیری در سفارشی‌سازی جزئیات گرافیکی، برای تجسم داده‌ها استفاده می‌شود. این ابزار با ادغام کامل با NumPy و Pandas، امکان نمایش بصری دقیق داده‌های عددی و ساختاریافته را فراهم می‌کند و در تحلیل‌های علمی و ارائه‌ی نتایج کاربرد دارد. قابلیت تولید خروجی‌های استاندارد و سازگار با نشریات آکادمیک، آن را به انتخابی ایده‌آل برای محققان و داده‌کاوان تبدیل کرده است.

## D. Seaborn چرا برای تجسم داده‌های پیشرفته کاربرد دارد؟

کتابخانه‌ی Seaborn به دلیل ارائه‌ی رابط سطح بالا بر پایه‌ی Matplotlib و تولید نمودارهای آماری پیشرفته (مانند نقشه‌های حرارتی، جعبه‌ای، و توزیع) با حداقل کد، برای تجسم داده‌های پیچیده کاربرد دارد. این ابزار با پشتیبانی داخلی از تحلیل‌های آماری (مثل رگرسیون و ماتریس همبستگی) و تم‌های بصری جذاب، داده‌های چندمتغیره را به صورت شهودی و قابل فهم نمایش می‌دهد. ادغام یکپارچه با Pandas و تمرکز بر زیبایی‌شناسی و دقت، آن را برای تحقیقات علمی و ارائه‌های حرفه‌ای مناسب می‌سازد.

## E. چگونه می‌توانید یک Function در Python تعریف کنید؟

در پایتون، یک تابع با استفاده از کلمه‌کلیدی `def` تعریف می‌شود، به دنبال آن نام تابع و پارامترها در پرانتز آمده و با دو نقطه (`:`) پایان می‌یابد، سپس بدنه‌ی تابع با تورفتگی مشخص می‌شود. این ساختار امکان کپسوله‌سازی منطق قابل‌استفاده‌ی مجدد را فراهم می‌کند، مانند

```
def add(a, b): return a+b
```

که دو عدد را جمع می‌کند.

## F. چرا List Comprehension در Python استفاده می‌شود؟

List Comprehension در پایتون ابزاری برای ساخت سریع لیست‌ها با یک خط کد، مانند `[x*2 for x in range(5)]`، به جای حلقه‌های چندخطی است. این روش با افزایش خوانایی و امکان افزودن شرط‌ها (مانند `[x for x in range(10) if x % 2 == 0]`)، کارایی را بالا می‌برد. برای استادان سخت‌گیر، این تکنیک به دلیل بهینه‌سازی و رعایت استانداردهای حرفه‌ای برنامه‌نویسی ارزشمند است.

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

برای خواندن یک فایل CSV در پایتون، می‌توانید از کتابخانه `pandas` استفاده کنید. ابتدا کتابخانه را وارد کرده، سپس با تابع `read_csv` فایل را خوانده و در یک `DataFrame` ذخیره کنید. به این صورت:

```
python
import pandas as pd
data =
pd.read_csv('file.csv')

print(data)
```

این کد فایل CSV را خوانده و محتوای آن را نمایش می‌دهد.

## H. JSON و XML چه تفاوتی دارند؟

JSON و XML هر دو فرمت‌هایی برای ذخیره و انتقال داده‌ها هستند، اما JSON سبک‌تر و خواناتر برای انسان است و از ساختار کلید-مقدار استفاده می‌کند، در حالی که XML از تگ‌ها برای تعریف داده‌ها بهره می‌برد و پیچیده‌تر است. JSON معمولاً سریع‌تر پردازش می‌شود و حجم کمتری دارد، در حالی که XML انعطاف‌پذیرتر و مناسب‌تر برای داده‌های ساختاری پیچیده است.