

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

(تمارین درس مباحث ویژه)

بخش دوم

مدرس: مهندس احمدزاده

دانشجویان:

عباس کارگر جهرمی / مهدی فرازنده شهرکی

دانشکده ملی مهارت میناب

بهمن ۱۴۰۳

۱. چرا Python زبان برنامه‌نویسی محبوب علم داده است؟

Python به دلیل سادگی سینتکس، انعطاف‌پذیری و داشتن کتابخانه‌های قدرتمند مانند NumPy، Pandas، Matplotlib، Scikit-learn و TensorFlow و غیره، یکی از محبوب‌ترین زبان‌های برنامه‌نویسی در علم داده است. همچنین، جامعه کاربری بزرگ، پشتیبانی از برنامه‌نویسی شی‌گرا و اسکرپتی، و توانایی اجرای آسان کدها در محیط‌هایی مانند Jupyter Notebook، از دیگر دلایل محبوبیت آن است.

۲. NumPy و Pandas چه تفاوتی دارند؟

- NumPy (Numerical Python) برای کار با آرایه‌های چندبعدی و عملیات عددی استفاده می‌شود. این کتابخانه کار با داده‌های عددی را سریع‌تر می‌کند و شامل توابع ریاضی و آماری پیشرفته است.
 - Pandas بر پایه NumPy ساخته شده و بیشتر برای تحلیل داده‌های ساختاریافته (جدولی) مانند DataFrame و Series به کار می‌رود. این کتابخانه ابزارهای قدرتمندی برای فیلتر کردن، گروه‌بندی، تغییر شکل و پردازش داده‌ها دارد.
- جمع‌بندی NumPy: برای پردازش آرایه‌های عددی و Pandas برای تحلیل داده‌های جدولی مناسب‌تر است.
-

۳. چرا Matplotlib برای تجسم داده‌ها استفاده می‌شود؟

Matplotlib یکی از محبوب‌ترین کتابخانه‌های تجسم داده در پایتون است که قابلیت ایجاد نمودارهای خطی، میله‌ای، دایره‌ای، هیستوگرام و ... را دارد. ویژگی‌های مهم آن:

- ✓ توانایی سفارشی‌سازی بالا
- ✓ امکان ذخیره نمودارها در فرمت‌های مختلف (PNG, PDF, SVG, ...)
- ✓ سازگاری با NumPy و Pandas
- ✓ مناسب برای تحلیل‌های ابتدایی داده

اگر نیاز به رسم نمودارهای پایه‌ای دارید، Matplotlib گزینه مناسبی است.

۴. چرا Seaborn برای تجسم داده‌های پیشرفته کاربرد دارد؟

Seaborn یک کتابخانه سطح بالاتر از Matplotlib است که برای تجسم داده‌های آماری و پیشرفته‌تر استفاده می‌شود. ویژگی‌های آن:

رنگ‌بندی زیباتر و طراحی حرفه‌ای‌تر

پشتیبانی از نمودارهای پیچیده مثل Heatmap، Boxplot، Violin plot، KDE plot و...

سازگاری عالی با Pandas DataFrame

نمایش بهتر روندها و توزیع‌های داده‌ها

اگر می‌خواهید داده‌ها را به صورت جذاب‌تر و حرفه‌ای‌تر نمایش دهید، Seaborn گزینه مناسبی است.

۵. چگونه می‌توانید یک Function در Python تعریف کنید؟

در پایتون، تابع‌ها با استفاده از کلمه کلیدی def تعریف می‌شوند. مثال:

python

CopyEdit

```
def greet(name):
```

```
    return f"Hello, {name}!"
```

```
print(greet("Abbas")) # : Hello, Abbas!
```

تابع‌ها به ما کمک می‌کنند تا کدهای تکراری را حذف کنیم و برنامه‌های بهینه‌تر بنویسیم.

۶. چرا List Comprehension در Python استفاده می‌شود؟

List Comprehension یک روش مختصر و کارآمد برای ایجاد لیست‌ها در یک خط کد است. این روش:

✓ خوانایی کد را افزایش می‌دهد

✓ اجرای سریع‌تری نسبت به حلقه for دارد

✓ به خصوص برای فیلتر کردن و تبدیل داده‌ها مفید است

مثال:

python

CopyEdit

```
numbers = [x**2 for x in range(10) if x % 2 == 0]
```

```
print(numbers) # خروجی [0, 4, 16, 36, 64]
```

اگر بخواهید لیست‌ها را به سرعت تولید کنید، List Comprehension انتخاب خوبی است.

۷. چگونه می‌توانید یک CSV file را در Python بخوانید؟

برای خواندن فایل CSV می‌توان از کتابخانه Pandas استفاده کرد:

```
python
```

```
CopyEdit
```

```
import pandas as pd
```

```
df = pd.read_csv("data.csv")
```

```
print(df.head()) # نمایش ۵ سطر اول داده‌ها
```

این روش بسیار سریع و مناسب برای تحلیل داده‌های جدول محور است.

۸. JSON و XML چه تفاوتی دارند؟

هر دو JSON و XML فرمت‌هایی برای ذخیره و انتقال داده‌ها هستند، اما تفاوت‌های کلیدی دارند:

ویژگی	JSON	XML
خوانایی	ساده‌تر و خواناتر	پیچیده‌تر
حجم داده	کمتر	بیشتر
ندارد (همه چیز به صورت متن ذخیره می‌شود) دارد (عدد، رشته، بولین، لیست) پشتیبانی از انواع داده		
سرعت پردازش	سریع‌تر	کندتر
ساختار	شبه دیکشنری پایتون	تگ‌های تو در تو مثل HTML

مثال: JSON

json

CopyEdit

```
{  
  "name": "Abbas",  
  "age": 20,  
  "city": "Tehran"  
}
```

مثال: XML

xml

CopyEdit

```
<person>  
  <name>Abbas</name>  
  <age>20</age>  
  <city>Tehran</city>  
</person>
```

JSON سریع‌تر و سبک‌تر است، بنابراین معمولاً برای وب‌سرویس‌ها و API ها استفاده می‌شود XML. قدیمی‌تر است و در سیستم‌های قدیمی‌تر کاربرد دارد.