

دانشگاه ملی مهارت

آموزشکده میناب

نام و نام خانوادگی : آرش زارعیان

واحد درسی : مباحث ویژه

رشته : مهندسی حرفه ای کامپیوتر

مدرس : محمد احمد زاده

بخش 2 Python Programming :

- A. چرا Python زبان برنامه‌نویسی محبوب علم داده است؟
- B. NumPy و Pandas چه تفاوتی دارند؟
- C. چرا Matplotlib برای تجسم داده‌ها استفاده می‌شود؟
- D. Seaborn چرا برای تجسم داده‌های پیشرفته کاربرد دارد؟
- E. چگونه می‌توانید یک Function در Python تعریف کنید؟
- F. چرا List Comprehension در Python استفاده می‌شود؟
- G. چگونه می‌توانید یک CSV file را در Python خواند؟
- H. JSON و XML چه تفاوتی دارند؟

مدرس: محمد احمدزاده

ترم بهمن ۱۴۰۳

مباحث ویژه

Day... Month... Year...

Subject...

بخش 2 Python Programming

A چرا Python زبان برنامه نویسی محبوب علم داده است؟

سادگی و خوانایی - پایتون سبک و شفاف دارد که یادگیری و استفاده از آن آسان است.

کتابخانه‌های قدرتمند - وجود کتابخانه‌هایی مانند Matplotlib, SciKit-learn.

Numpy و Pandas پردازش داده و یادگیری ماشین را بسیار ساده‌تر می‌کنند.

انعطاف پذیری بالا - پشتیبانی گسترده - یکپارچه‌سازی با سایر ابزارها.

B. Numpy و Pandas چه تفاوتی دارند؟

از هر دو کتابخانه در پایتون برای پردازش داده استفاده می‌شود اما تفاوت‌های عمده دارند.

1. نوع داده و ساختار:

Numpy: آرایه‌های چندبعدی را مدیریت می‌کند. این آرایه‌ها همتای هستند.

Pandas: دو ساختار داده‌ای اصلی دارد: Series (یک بعدی) و DataFrame (دو بعدی مثل جدول).

این ساختارها می‌توانند داده‌های نامرتب را ذخیره کنند.

Day...	Month...	Year...	Subject...
1			۲. قابلیت‌ها و عملکرد:
2			
3			Numpy : سریع تر است، چون بهینه سازی شده و روی آرایه های عددی کار می کند.
4			
5			Pandas : روی داده های جدولی (مانند فایل های CSV و پایگاه داده ها) تمرکز دارد و
6			
7			توانمندی برای دسته بندی، فیلتر کردن و تجزیه و تحلیل داده ها ارائه می دهد.
8			
9			۳. استفاده های رایج:
10			
11			Numpy : پردازش داده های عددی، عملیات ریاضی، محاسبات ماتریسی
12			
13			Pandas : تحلیل داده، پردازش و پاک سازی داده های خام، کار با داده های ساختاریافته
14			
15			۴. نحوه دسترسی به داده ها:
16			
17			Numpy : آرایه های عددی استفاده می کنند
18			
19			Pandas : علاوه بر آرایه های عددی، آرایه های سفارشی و به حسب دار هم داریم (مثل سری زمانی)
20			
21			و انعطاف پذیری بیشتری می دهد.
22			
23			۵. واسطه ها:
24			

Day... Month... Year...

Subject...

1 **Numpy** یک پایم برای بسیاری از کتابخانه‌های دیگر (مثل SciPy) است.

2

3 **Pandas** خودش وابسته به Numpy است و از قابلیت‌های آن برای پردازش

4

5 سریع تر داده‌ها استفاده می‌کند.

6

7 در کل این با اعداد و عملیات ریاضی کار می‌کند Numpy و این با داده‌های جدولی

8

9 و تحلیلی سروکار دارید Pandas گزینه مناسب‌تری است.

10

11 **C** یا **Matplotlib** برای تجسم داده‌ها استفاده می‌شود.

12

13 **Matplotlib** یکی از محبوب‌ترین کتابخانه‌های پایتون برای تجسم داده‌ها است.

14

15 ۱- تنوع نمودارها؟

16

17 **Matplotlib** امکان رسم انواع مختلف نمودارها را فراهم می‌کند از جمله: **Line Plot**

18

19 **Box Plot** - **Histogram** - **Pie chart** - **Scatter Plot** - **Bar chart**

20

21 ۲- قابلیت سفارشی‌سازی بالا ۳- سازگاری با سایر کتابخانه‌ها (مثلاً اسکین ذخیره

22

23 نمودارها در فرمت‌های مختلف ۵- رسم نمودارهای تعاملی

24

AVANCE

Day... Month... Year...

Subject...

1 D. Seaborn چرا برای تجسم داده‌های پیچیده نام برود را بگو؟

2
3 Seaborn یک کتابخانه پیشرفته برای تجسم داده‌ها در پایتون است که بر پایه Matplotlib

4
5 ساخته شده است. این کتابخانه به دلیل امکانات بیشتر و ظاهر زیباتر، برای تحلیل و نمایش

6
7 داده‌های پیچیده بسیار مناسب است.

8
9 ۱- زیبایی و طراحی پیشرفته تر ۲- تجسم روابط بین متغیرها

10
11 ۳- تعامل مستقیم با Pandas ۴- نمایش توزیع داده‌ها به سادگی

12
13 ۵- نمایش داده‌های گرده پرتی شده

14
15 در کل این نمودارهای پیشرفته، زیبا و تحلیلی می‌خواهید Seaborn گزینه بهتری نسبت

16
17 به Matplotlib است. اما اگر نیاز به سفارشی‌سازی یا لا دارید، ترکیب

18
19 Matplotlib + Seaborn ایده‌آل‌تر است.

20
21 E. چگونه می‌توانید یک Function در Python تعریف کنید؟

22
23 در پایتون برای تعریف یک تابع (Function) از کلمه کلیدی def استفاده می‌کنیم.

24
AVANCE

Day... Month... Year...

Subject...

ساختار کلی تابع در پایتون:

def function-name(parameters):

def add(a, b):

return a + b

result = add(5, 3)

print(result) => خروجی = 8

مثال: تابع با مقدار بازگشتی

خواه در پایتون کد را خوانا تر،
قابل استفاده مجدد و ساختار یافته
می کنند

F. چرا List Comprehension در Python استفاده می شود؟

List comprehension یک روش کوتاه، سریع و خوانا برای ایجاد لیست ها در پایتون است.

به جای استفاده از یک می توان یک لیست را با یک خط کد ساخت:

مزایای list comprehension:

کد خوانا تر و کوتاه تر -> به جای چیدن خط ها، همان کار را می توان با یک خط انجام داد.

اجرای سریع تر -> کارایی بیشتر نسبت به حلقه های معمولی دارد.

سادگی در فیلتر کردن داده ها -> امکان اعمال شرط (if) در هنگام ساخت لیست

ایجاد تغییرات روی عناصر -> به راحتی می توان عملیات ریاضی و تغییرات دیگر را انجام داد.

AVANG

Day... Month... Year...

Subject...

1 G: چگونه می‌توان یک CSV file را در Python خواند؟

2
3 برای خواندن یک فایل CSV در پایتون، می‌توان از ماژول CSV یا Pandas استفاده کرد.

4
5 روش 1: استفاده از ماژول CSV (داخلی پایتون)

6
7 `import csv`

8
9 `with open('file.csv', mode='r', encoding='utf-8') as file:`
10 `reader = csv.reader(file)`

11 `for row in reader:`

12 `print(row)` # هر ردیف یک لیست است

13
14 * این روش سبک و سریع است اما نیاز به پردازش دستی داده‌ها دارد.

15
16 روش 2: استفاده از Pandas (روش پیشرفته)

17 `import pandas as pd`

18
19 `df = pd.read_csv('file.csv', encoding='utf-8')`

20
21 `print(df)` # Data Frame (داده‌ها به صورت)

22
23 * این روش راحت‌تر و قوی‌تر است، مخصوصاً برای پردازش داده‌ها.

Day... Month... Year... Subject...

1 H. JSON و XML چه تفاوتی دارند؟

2 ۱- ساختار و خوانایی

3

4 JSON: برای JavaScript Object Notation است. ساده، خوانا و کم حجم

5

6 XML: برای زبان نشانه گذاری است. پیچیده تر و پر حجم تر

7

8 ۲- خوانایی برای انسان و ماشین

9

10 JSON خوانا و کوتاه تر است.

11

12 XML دارای ساختار درختی و برای پردازش توسط ماشین مناسب تر است.

13

14 ۳- حجم و کارایی؟

15

16 JSON حجم کمتری دارد و سریع تر پردازش می شود.

17

18 XML به دلیل استفاده از تگ های اضافی، سنگین تر است.

19

20 ۴- پشتیبانی در زبان های برنامه نویسی

21

22 JSON مستقیماً در JavaScript و اکثر زبان های پشتیبانی می شود.

23

24 XML نیاز به پردازش با کتابخانه های مخصوص دارد.

AVANGE

Day... Month... Year...

Subject...

1 در مورد استفاده

2
3 **Json**: مناسب برای وب سرویس ها (API) پایه داده NoSQL ، تنظیمات نرم افزار
4 داده های شبکه

5
6 **XML**: مناسب برای سند های پیچیده ، وب سرویس های قدیمی (SOAP) ، RSS
7 و تنظیمات سیستم های بزرگ

8
9 در حال سرعت و سادگی هم است Json بهتر است و الزاماً به ساختار پیچیده

10
11 استانداردهای قدیمی داریم ، XML بهتر است .
12
13
14
15
16
17
18
19
20
21
22
23
24

AVANGE