لزوم آشنایی با برنامه نویسی

شاگرد: اینکه تأکید داری طلبه باید برنامه‌نویسی بداند، منظورتان از آن چیست؟

استاد: ببینید یک چیزهایی کلیات زبان‌های برنامه‌نویسی است. وقتی شما بدانید اینها چه کار می‌کنند؛ مه‌ندس نرم‌افزار چه کار می‌کند و بعد برنامه‌نویس چه کار می‌کند، ذهن شما یک نظم خاصی پیدا می‌کند و هم درک می‌کند؛ این برای کارهای علمی شما نافع است و هم بعداً برای کارهایی که نیاز دارید.

اما اینکه کدام زبان باشد، حال وقتی آشنا شدید. زبان‌ها جوراجور است. زبان‌های ساده هست. شاید زبانی که الآن باز است - سی شارپ مایکروسافت «Open source» نیست- پایتون است. «Open source» است و زبان خیلی قوی سطح بالا است. قوی‌ترین زبان همین است. شبکه‌های عصبی را در هوش مصنوعی نوعا با همین پایتون و کتابخانه‌های آماده‌ای که دارند می‌نویسند. نوع ابزارهایی که در گوگل استفاده می‌کنند را با پایتون می‌نویسند. من اینها را بلد نیستم. فقط عرض می‌کنم؛ تجربه‌ای است؛ ما به سن پیری رسیدیم و برای ما ممکن نیست، اما شما در سن جوانی برای آن وقت زیادی نگذارید که به تمرکزتان در علوم طلبگی صدمه بزند. اما مدیریت زمان کنید. مثلاً در هفته نیم ساعت. یا در روز ده دقیقه. این ده دقیقه در هر روز را در یک سال و پنج سال آینده می‌بینید چه می‌شود. منظور من این است که یک چیزی که پر فایده است از دست نرود.

جلوتر هم عرض کردم؛ من یک خاطره خیلی خوبی از همین کد نویسی‌ها دارم. المنطق را که در صحن مسجد فاطمیه مباحثه می‌کردیم، در بین مباحثه حالات خاصی برای من پیش می‌آمد. دقیقاً یادم هست که چطور می‌شد. دو نفر از آقایان بودند که خوش ذهن بودند، بین مباحثه رفت‌وبرگشت می‌شد، یک دفعه من بی اختیار می‌گفتم این حیثیاتی که الآن برای ما ظهور کرد، اگر هزار سال دیگر هم همه بحث کنند و به سر و کله هم بزنند، جدا کردنشان سخت است. نمی‌شود. آن می‌گوید تو حرف من را نفهمیدی و دیگری می‌گوید تو حرف من را نفهمیدی. آدم یک دفعه می‌بیند که در اینجا ده حیثیت است؛ الفاظ هم کم است، هر کدام یکی از آن‌ها را می‌گیرند؛ حال بیا تا این ده تا روشن شود. این حال من در آن مباحثه بود. تا اینکه این زبان‌ها را بعداً دیدم؛ یک حالت وجدی به من دست دارد که نمی‌توانم توضیح بدهم. چرا؟ به‌خاطر اینکه کسانی که این کدها را می‌نوشتند سر و کارشان با یک هم مباحثه نبود که بگویند حرف من را نمی فهمی. بلکه یک کد را می‌نوشت و غلط عمل می‌کرد. می‌گفت من که درست نوشته ام، تو چرا اینطور غلط می‌روی؟! آن هم با زبان بی زبانی می‌گفت من یک سیم مسی و مدار هستم، تو خودت می‌دانی! باید برگردی و خودت آن را اصلاح کنی! دقیقاً به این صورت بود. باگ گیری که می‌کنند همین است. مدام بر می گرد تا ببیند کجا را اشتباه کرده است. می‌فهمد که در اینجا اشتباه کرده است.

> 10:31

لذا اولین باری که قبل از پایتون، سی شارپ آمد، ظاهراً اولین زبانی که به تمام معنا شیء گرا بود همین سی شارپ بود. بعدش زبان‌های دیگر هم آمد. قبل از آن «C++» بود اما اصل زبان C، شیءگرا نبود. بعد آن را با پلاس، شیء گرا کردند. وقتی هم شیء گرا می‌شد برای هوش مصنوعی کاربرد خیلی خوبی داشت. زبان C از اساس شیءگرا بود، پایتون هم همینطور بود. یعنی تمام ساختارشان روی چیزی است که به درد هوش مصنوعی می‌خورد.

منظورم این بود: اولین بار که در آن کتاب شیء گرا دیدم، شما نمی‌دانید بعضی از واژه‌ها هست که نزدیک هم است، مدتی کار می‌برد تا خود شما بفهمید فرق اینها چیست. یعنی حیثیات به قدری نزدیک هم است که فرق آن‌ها برای ما که هنوز کار نکرده‌ایم سخت است. اگر ما آستین بالا بزنیم و پنج-شش برنامه بنویسیم و از اینها استفاده کنیم، آن وقت می‌فهمیم که اینها فرق دارد. خدای متعال همه اینها را در ذهن ما قرار داده، اما تا فرق اینها را با لدقة متوجه شویم و پیاده شود، نیاز به این کارهایی داشته که انجام داده‌اند.

بازبینی #1
ایجاد شده 2 مه 2026 11:56:38 توسط ... .
به روزرسانی شده 2 مه 2026 11:57:13 توسط ... .