

کلاس خصوصی (مخصوص یادگیری مفهومی و عمیق)

موضوع: برنامه نویسی پایتون (۳)

این جزوه برای یادگیری مفهومی و عمیق (چرا و چگونه؟) مطالب برنامه نویسی هست

هدف: درک منطق پشت کدها بدون حفظ کردن

۱. داستان "فضای نام (Scope)"؛ شهر در مقابل خانه

تصور کنید شما در خانه خودتان (تابع) یک متغیر به نام «دما» دارید که روی ۲۵ درجه تنظیم است. در سطح شهر (بدنه اصلی برنامه) هم یک متغیر «دما» داریم که ۳۰ درجه است.

- **متغیر محلی (Local):** دمای خانه شماست. فقط کسانی که در خانه (تابع) هستند آن را می بینند.
- **متغیر سراسری (Global):** دمای شهر است. همه به آن دسترسی دارند.
- **مشکل:** اگر داخل خانه بگویید «دما = ۴۰»، فقط دمای خانه عوض می شود.
- **راه حل (global):** اگر بخواهید از داخل خانه، دمای کل شهر را عوض کنید، باید از کلمه کلیدی **global** استفاده کنید تا پایتون بفهمد منظور شما همان متغیر بیرونی است.

۲. جعبه ابزار منعطف (kwargs* و args*)

- **args* (لیست متغیر):** فرض کنید یک دستگاه آبمیوه گیری (تابع) دارید. گاهی ۱ سیب می اندازید، گاهی ۱۰ سیب. args* مثل یک **قیف جادویی** است که هر تعداد میوه (آرگومان) بریزید، همه را می گیرد و در یک بسته (**Tuple**) به شما تحویل می دهد.
- **kwargs** (لیست با پرچسب):** این مثل سفارش غذا با جزئیات است. فقط نمی گوئید غذا؛ می گوئید: نوع = پیتزا، سس = سیر. پایتون این ها را به صورت دیکشنری (**کلید: مقدار**) ذخیره می کند.

۳. دنیای شی گرایی (OOP)؛ نقشه در برابر ساختمان

این مهم ترین تغییر تفکر در برنامه نویسی است.



- **کلاس (Class):** مثل نقشه معماری ساختمان است. نقشه، خانه نیست؛ فقط می‌گوید خانه چه شکلی است.
- **شیء (Object):** خانه‌ای است که با آجر و سیمان از روی آن نقشه ساخته شده. شما می‌توانید از روی یک نقشه (کلاس Car)، هزاران خانه (شیء car1, car2) بسازید.
- **متد (Method):** کارهایی که آن شیء می‌تواند انجام دهد (مثل ترمز کردن خودرو). در واقع همان تابع است که داخل کلاس حبس شده.
- **__init__ و self :**
 - وقتی ساختمان ساخته می‌شود (شیء ایجاد می‌شود)، همان لحظه اول باید رنگ دیوار و تعداد اتاقش مشخص شود. __init__ تابع سازنده‌ای است که در لحظه تولد شیء اجرا می‌شود.
 - self یعنی "مال خودم". وقتی می‌گوییم self.color، یعنی رنگِ همین ساختمانی که الان تویش هستیم، نه ساختمان کناری.