

## چالش برنامه‌نویسی

**مسئله:** تابعی بنویسید که مجموع مربعات اعداد زوج در یک لیست را بازگرداند.

- **بررسی راه‌حل‌ها:**

- مشکلات رایج: تعارض نام‌گذاری (مانند استفاده از کلمات کلیدی رزرو شده مثل list).
- راه‌حل‌های بهینه با استفاده از فهرست‌بندی، filter, map, reduce و توابع lambda نمایش داده شدند.

## مقدمه‌ای بر دکوریتورها

- **تعریف:** توابعی که توابع دیگر را با افزودن قابلیت‌هایی تغییر می‌دهند.
- **استفاده:**
  - نحو: decorator\_name@
- **نکات کلیدی:**
  - دکوریتورها توابع تو در تو هستند که wrapper منطق اضافی را مدیریت می‌کند.
  - استفاده از args\* و kwargs\*\* برای انعطاف‌پذیری.
- **کاربردهای عملی:** ثبت لاگ‌ها، اندازه‌گیری عملکرد و مدیریت خطا.

## انباشتن دکوریتورها

- **لایه‌بندی:** چندین دکوریتور می‌توانند روی یک تابع اعمال شوند.
- **ترتیب اهمیت دارد:** دکوریتورهای بیرونی هنگام فراخوانی تابع اول اجرا می‌شوند.

## دکوریتورها برای اعتبارسنجی و مدیریت خطا

- **چالش:** دکوریتوری بنویسید که استثناهای خاص را بگیرد و به صورت مناسب مدیریت کند.
- **کاربرد در دنیای واقعی:** مدیریت متمرکز خطا برای سرویس‌های خارجی.

## مقدمه‌ای بر بازگشت (Recursion)

- **تعریف:** توابعی که خودشان را برای حل مسائل کوچکتر فراخوانی می‌کنند.
- **مثال‌ها:**
  - **فاکتوریل:** با موارد پایه و توضیح گام‌به‌گام نمایش داده شد.
  - **فیبوناچی:** پیاده‌سازی بازگشتی توضیح داده شد، با توجه به محاسبات تکراری.
- **مفاهیم کلیدی:**
  - موارد پایه برای جلوگیری از حلقه‌های بی‌پایان ضروری هستند.
  - عمق بازگشتی می‌تواند در صورت بیش از حد بودن منجر به خطا شود.

## ساختار داده‌ای مجموعه (Set)

- **ویژگی‌ها:**
  - بدون ترتیب و عناصر یکتا.
  - قابل تغییر اما بدون ایندکس‌گذاری.
- **عملیات‌ها:**
  - **اجتماع (|)** یا a.union(b)، **اشتراک (&)** یا a.intersection(b) و **تفاضل (-)** یا a.difference(b).
  - بررسی عضویت با in.
- **کاربردها:** ذخیره و مدیریت کارآمد داده‌ها.

- **توابع کلیدی:**
  - **count():** تولید اعداد بی‌نهایت با شروع و گام دلخواه.
  - **cycle():** تکرار مداوم در یک دنباله ورودی.
  - **chain():** ترکیب چند Iterable به یک.
  - **combinations():** تولید ترکیب‌های یکتا با طول مشخص.
  - **product():** محاسبه حاصل ضرب دکارتی Iterable‌ها.
  - **permutations():** بازگرداندن تمام ترتیبات ممکن عناصر.
  - **groupby():** گروه‌بندی عناصر بر اساس یک تابع انتخاب‌کننده.
- **کاربرد:** مدیریت مجموعه داده‌های بزرگ، محاسبات ریاضی و تولید ترکیب‌های یکتا.