جزوه: مفاهیم برنامه‌نویسی تابعی در پایتون

1. تعریف توابع و تمرین‌ها

درس با تمرینی شروع می‌شود که خواسته می‌شود دو تابع ایجاد کنند:

- تابع اول یک عدد را به عنوان ورودی گرفته و مقدار مربع آن را برمی‌گرداند.
- تابع دوم یک تابع (به‌طور خاص، تابع اول) و مجموعه‌ای از اعداد را به عنوان ورودی دریافت کرده، این تابع را روی هر عدد اعمال کرده و لیستی از نتایج را بازمی‌گرداند.

نکات مهم:

- مدرس دو روش برای حل این مسئله را نشان داد: یکی با استفاده از حلقه و دیگری با استفاده از "List Comprehension".

2. کپی کردن اشیاء

در این بخش، تفاوت بین **کپی سطحی** و **کپی عمیق** مورد بحث قرار گرفت:

- **کپی سطحی:** یک شیء جدید ایجاد می‌کند اما اشیای تو در تو را ارجاع می‌دهد. بنابراین، تغییرات در اشیای تو در تو در نسخه کپی شده، روی شیء اصلی نیز تاثیر می‌گذارد.
- **کپی عمیق:** یک نسخه مستقل کامل از شیء اصلی ایجاد می‌کند، از جمله تمام اشیای تو در تو. بنابراین، تغییرات در نسخه کپی شده هیچ تاثیری روی شیء اصلی ندارند.

نکات کلیدی:

- کپی عمیق برای لیست‌ها و دیکشنری‌ها به صورت پیش‌فرض وجود ندارد و باید از ماژول copy وارد شود.
- کپی عمیق برای مواردی استفاده می‌شود که نیاز به تغییر نسخه‌ای از داده‌ها بدون تاثیر بر داده‌های اصلی داریم.

3. برنامه‌نویسی تابعی

مدرس مفاهیم برنامه‌نویسی تابعی را معرفی کرد که در آن همه‌چیز از طریق توابع انجام می‌شود. در این رویکرد:

- توابع **اولین کلاس شهروندی** هستند: می‌توان آن‌ها را به عنوان آرگومان به دیگر توابع داد و به عنوان مقدار بازگشتی برگرداند.
- از خروجی یک تابع به عنوان ورودی برای توابع دیگر استفاده می‌شود.

اصول کلیدی برنامه‌نویسی تابعی:

- **تغییرناپذیری:** اشیاء پس از ایجاد نباید تغییر کنند. به جای آن، یک شیء جدید با تغییرات مورد نیاز ایجاد می‌شود.
- **توابع خالص:** این توابع همیشه برای ورودی‌های مشخص، خروجی یکسانی را تولید می‌کنند و اثر جانبی ندارند (یعنی متغیرهای خارجی را تغییر نمی‌دهند).
- **اجتناب از اثرات جانبی:** توابع نباید چیزی خارج از حوزه محلی خود را تغییر دهند، مانند متغیرهای سراسری یا انجام عملیات ورودی/خروجی.

مزایا:

- کاهش پیچیدگی کد
- افزایش پیش‌بینی‌پذیری

4. توابع لامبدا

- **توابع لامبدا:** روشی برای ایجاد توابع کوچک و ناشناس هستند که به صورت درون‌خطی نوشته می‌شوند.
- توابع لامبدا می‌توانند چندین ورودی داشته باشند اما فقط یک عبارت می‌توانند داشته باشند.
- معمولاً برای عملیات ساده استفاده می‌شوند و در ابزارهای برنامه‌نویسی تابعی به کار می‌روند.

5. ابزارهای برنامه‌نویسی تابعی در پایتون

- **map()**:
  - یک تابع مشخص را روی هر عضو از یک iterable اعمال کرده و یک شیء map (که یک iterator است) بازمی‌گرداند.
  - اغلب با توابع لامبدا برای عملیات سریع استفاده می‌شود.
- **filter()**:
  - یک iterator ایجاد می‌کند که فقط شامل عناصری است که تابع مشخص شده مقدار true را برای آن‌ها برمی‌گرداند.
  - می‌تواند برای فیلتر کردن داده‌ها بر اساس یک شرط استفاده شود.
- **reduce()**:
  - به صورت تجمعی یک تابع را روی اعضای یک iterable اعمال کرده و آن را به یک مقدار کاهش می‌دهد.
  - در ماژول functools قرار دارد و به صورت پیش‌فرض موجود نیست.
  - تابع مشخص شده باید دو آرگومان بپذیرد و روی اولین دو عضو iterable اعمال شود و سپس روی نتیجه و عضو بعدی اعمال گردد.

نکته: map() و filter() اشیاء iterator بازمی‌گردانند که می‌توانند به لیست تبدیل شوند یا با حلقه for پیمایش شوند.

6. **ژنراتورها و Iteratorها**

- **ژنراتورها:** روشی برای ایجاد iteratorها به صورت تنبل (lazy) هستند:
  - از کلیدواژه yield برای تولید مقادیر به صورت تک‌به‌تک استفاده می‌کنند.
  - به‌جای محاسبه همه مقادیر به‌طور هم‌زمان، حافظه و زمان پردازش را ذخیره می‌کنند.
  - برای کار با مجموعه‌داده‌های بزرگ که بارگذاری همه آن‌ها در حافظه عملی نیست، مناسب هستند.
- تفاوت کلیدی بین yield و return:
  - yield اجرا را متوقف کرده و حالت را برای تکرار بعدی ذخیره می‌کند.
  - return عملکرد را خاتمه می‌دهد.
- **comprehension ژنراتور:** از پرانتزها به جای کروشه‌ها یا آکولادها استفاده شده و اشیاء iterator تنبل به صورت مختصر ایجاد می‌کند.

نکته: ژنراتورها فقط یک بار قابل استفاده هستند. اگر مقادیر آن‌ها لازم باشد، باید ذخیره شوند.

7. **توضیحات اضافی**

- **متد sort()**:
  - لیست را به‌طور inplace تغییر می‌دهد.
  - استفاده از کلید key برای سفارشی‌سازی عملیات مرتب‌سازی امکان‌پذیر است.

نتیجه‌گیری: این درس مفاهیم برنامه‌نویسی تابعی در پایتون را توضیح داده و ابزارهایی مانند توابع لامبدا، map()، reduce()، filter() و ژنراتورها را برای نوشتن کدی مختصرتر و کارآمدتر معرفی می‌کند. بر اهمیت تغییرناپذیری، توابع خالص و اجتناب از اثرات جانبی تاکید شده است.