جزوه: اصول برنامه‌نویسی پایتون

- **استفاده از هوش مصنوعی:**

استفاده از هوش مصنوعی برای حل تمرین‌ها غیرمجاز است. هدف تمرین‌ها، یادگیری و درک عمیق مطالب است. در صورتی که از هوش مصنوعی برای تحقیق یا جمع‌آوری اطلاعات استفاده می‌کنید، باید توانایی توضیح و درک پاسخ‌های تولید شده را داشته باشید. ارسال پاسخ‌های تولید شده توسط هوش مصنوعی بدون این توانایی قابل قبول نیست.

- **فرمت تکالیف:**

تکالیف باید به صورت فایل‌های پایتون (.py) ارسال شوند. فرمت‌هایی مانند فایل‌های آی‌پایتون (.ipynb)، PDF، یا تصاویر کد پذیرفته نمی‌شوند.

  - فایل‌های پایتون باید:
    1. قابل اجرا باشند.
    2. خروجی مورد انتظار را از طریق دستور python یا python3 در خط فرمان تولید کنند.
    3. یا به صورت importable باشند و خروجی لازم را هنگام فراخوانی فراهم کنند.

- **دریافت کمک:**
  - از منابع موجود استفاده کنید.
  - سوالات خود را در جلسات تمرینی مطرح کنید. این جلسات برای شفاف‌سازی موضوعات طراحی شده‌اند.

- **آزمون‌ها و ارزیابی‌ها:**

آزمون‌هایی برای ارزیابی مهارت‌های فنی برگزار خواهد شد که ممکن است شامل سوالات کدنویسی و توصیفی باشند. آزمون‌ها ممکن است به جلسات تمرینی اضافه شوند.

تمرین کلاسی: نوشتن توابع برای مدیریت فایل‌ها

تمرین: تابعی بنویسید که نام یک فایل را به عنوان ورودی بگیرد:

1. اگر فایل وجود داشته باشد، محتوای آن بازگردانده شود.
2. اگر فایل وجود نداشته باشد:
   - فایلی جدید با محتوای "Hello World" ایجاد شود.
   - عبارت "New file created" بازگردانده شود.
3. تابع باید خروجی را بازگرداند، نه چاپ کند.

مفاهیم توابع در پایتون

- **بازگرداندن مقادیر:**

توابع باید خروجی‌های مورد نیاز را بازگردانند. بازگرداندن داده به کمک دستور return امکان‌پذیر است.

- **توقف اجرا با return:**

با رسیدن به دستور return، اجرای تابع متوقف می‌شود. این ویژگی می‌تواند برای خروج از یک شرط یا حلقه درون تابع استفاده شود.

- **مدیریت خطا با try/except:**

استفاده از بلوک‌های try/except برای مدیریت خطاها (مانند FileNotFoundError) روشی استاندارد برای جلوگیری از خرابی برنامه است.

انواع داده در پایتون

1. دیکشنری‌ها (Dictionaries):

- دیکشنری‌ها داده‌ها را به صورت جفت‌های **کلید-مقدار** ذخیره می‌کنند.
- **ویژگی‌ها:**
  - کلیدها باید تغییرناپذیر باشند.
  - مقادیر با استفاده از کلیدها قابل دسترسی هستند.
  - استفاده از متد .get() برای جلوگیری از خطای KeyError.
- **کاربردها:**
  - مناسب برای ذخیره داده‌های ساختاریافته مثل تنظیمات و پیکربندی‌ها.
- **متدهای مهم:**
  - .pop()، .items()، .values()، .keys().
  - امکان مرتب‌سازی کلیدها با استفاده از تابع sorted() وجود دارد.

2. تاپل‌ها (Tuples):

- تاپل‌ها مشابه لیست‌ها هستند ولی **تغییرناپذیر** هستند.
- **ویژگی‌ها:**
  - استفاده بهینه‌تر از حافظه نسبت به لیست‌ها.
  - مناسب برای داده‌هایی که نیازی به تغییر ندارند.
  - امکان استفاده به عنوان کلید در دیکشنری‌ها.
- **کاربردها:**
  - بازگرداندن چند مقدار از یک تابع.

3. تمرین کلاسی با تاپل‌ها:

تابعی بنویسید که یک تاپل از اعداد دریافت کند و لیستی از اعدادی که مضرب 3 هستند را بازگرداند.

4. (List Comprehension):

- **تعریف:**

  روشی مختصر برای ایجاد لیست‌ها.

  - فرمت: [عبارت for متغیر in قابل‌تکرار if شرط].
- **کاربردها:**
  - ایجاد دیکشنری‌ها یا مجموعه‌ها با فرمت مشابه.
  - فیلتر کردن یا انجام عملیات روی عناصر.

5. مجموعه‌ها (Sets):

- مجموعه‌ها داده‌ها را بدون ترتیب و بدون تکرار ذخیره می‌کنند.
- **ویژگی‌ها:**
  - قابل تغییر هستند.
  - امکان افزودن یا حذف عناصر با متدهایی مثل .add() و .remove() وجود دارد.
- **کاربردها:**
  - حذف مقادیر تکراری از یک لیست.