المحاضرة 4

الدرس الرابع: البرمجة في بايثون - المنطق البرمجي (الحلقات والشروط)

في هذا الدرس، سنضع القواعد التي تحكم كيفية تدفق البيانات. سنتعلم كيف نجعل البرنامج يفكر (عبر الشروط) وكيف نجعل البرنامج يعمل بجد (عبر الحلقات التكرارية) لمعالجة مئات التقارير في لمح البصر.[1]

1. الجمل الشرطية (Conditions): كيف يتخذ الحاسوب قراراً؟

تخيل الجملة الشرطية كأنها مفرق طرق. في علم النفس، نستخدم المنطق الشرطي لتصنيف الحالات.[3] في بايثون، نستخدم ثلاث كلمات رئيسية: if (إذا)، elif (اختصار لـ وإلا إذا)، و else (وإلا/فيما عدا ذلك).[4]

الهيكل المنطقي لاتخاذ القرار:

if: نضع بعدها الشرط الأول. إذا تحقق، ينفذ الحاسوب الأمر الملحق بها.

elif: نستخدمها إذا لم يتحقق الشرط الأول وأردنا فحص شرط آخر.

else: هي الملاذ الأخير، وتنفذ إذا لم يتحقق أي من الشروط السابقة.

تطبيق تربوي (تصنيف درجات القلق):

لنفترض أن لدينا مقياساً للقلق من 10 درجات:

Python

```
score = 8

if score >= 8:
    print("قلق مرتفع: يحتاج لتدخل فوري")
elif score >= 5:
    print("قلق متوسط: يحتاج لمتابعة")
else:
    print("حالة مستقرة")
```

أدوات المقارنة المنطقية:

ليتمكن الحاسوب من المقارنة، نستخدم الرموز التالية:

== (يساوي تماماً) - *لاحظ استخدام علامتي يساوي للتمييز عن تعريف المتغيرات.*

!= (لا يساوي).

> و < (أكبر من وأصغر من).

and (و): لربط شرطين معاً (يجب أن يتحققا معاً).

or (أو): يكفي تحقق واحد منهما فقط.

2. الحلقات التكرارية (Loops): المساعد الذي لا يتعب

في البحث التربوي، قد يكون لديك قائمة تضم 1000 طالب، وتريد فحص درجاتهم جميعاً. هل ستقوم بكتابة كود لكل طالب؟ بالطبع لا. هنا تأتي **حلقة for**.[4]

الحلقة التكرارية هي أمر يخبر الحاسوب: كرر هذه الخطوات على كل عنصر في هذه القائمة حتى تنتهي القائمة.[4]

مثال: فحص قائمة من الكلمات المفتاحية

تخيل أنك تبحث عن كلمات تدل على التنمر في مذكرات يومية لطالب:

Python

```
diary_words = [لعبت, ضربني, أكلت, سخروا, درست]
keywords = [ضربني, سخروا, تنمر]

for word in diary_words:
    if word in keywords:
        print(f"تحذير: تم رصد كلمة مقلقة: {word}")
```

المفهوم	الوظيفة	مثال من الواقع التربوي
حلقة for	المرور على عناصر قائمة واحداً تلو الآخر	المرور على أسماء الطلاب في فصل دراسي لتوزيع المهام
حلقة while	التكرار طالما أن هناك شرطاً محققاً	استمرار عرض الأسئلة للطالب طالما لم يضغط على إنهاء الاختبار
break	إيقاف التكرار فوراً والخروج	التوقف عن البحث بمجرد العثور على أول مؤشر لخطر الانتحار في النص

3. الدمج بين الشروط والحلقات (القوة الحقيقية)

القوة الحقيقية تظهر عندما نضع الشرط داخل الحلقة. هذا هو الأساس الذي تقوم عليه برامج تحليل المحتوى (Content Analysis).[4]

سيناريو: لديك قائمة بدرجات 5 طلاب، وتريد معرفة من منهم متفوق (أعلى من 90) ومن متعثر (أقل من 50).

Python

```
grades =

for g in grades:
    if g >= 90:
        print(f"الدرجة {g}: طالب متفوق")
    elif g < 50:
        print(f"الدرجة {g}: طالب يحتاج دعم")
```

4. تطبيق بايثون في معالجة النصوص النفسية

بناءً على ما تعلمناه في الدرسين السابقين (التقطيع وإزالة كلمات الوقف)، يمكننا الآن استخدام الحلقات والشروط للقيام بتحليل أعمق:[4]

عد الكلمات يدوياً: يمكننا إنشاء حلقة تمر على كل الكلمات في نص المقابلة، وإذا كانت الكلمة هي خوف، نقوم بزيادة عداد خاص.[5]

تصنيف المشاعر المبسط: يمكننا وضع قائمة بكلمات إيجابية وقائمة بكلمات سلبية، ثم جعل الحاسوب يمر على النص ويصنف صاحبه (إيجابي/سلبية) بناءً على الكلمات الأكثر تكراراً.[9]

5. لماذا هذا مهم لطلبة علم النفس التربوي؟

كأخصائيين مستقبليين، ستواجهون مواقف تتطلب دقة ومنطقاً:

تصحيح الاختبارات: بدلاً من التصحيح اليدوي، كود بسيط من 5 أسطر يمكنه تصحيح آلاف الاختبارات وإعطاء تقرير بالنتائج.[11]

الملاحظة المنظمة: عند رصد سلوك معين في الفصل، يمكن للحاسوب تسجيل التكرارات وتحليل الأنماط الزمنية (مثلاً: هل يزداد السلوك العدواني إذا كان الوقت بعد الفسحة؟).[12]

تخصيص التعلم: بناء أنظمة تخبر الطالب: إذا أخطأت في هذا السؤال، فإليك هذا الفيديو الشارح، وإلا فانتقل للسؤال التالي.[14]

الخلاصة

لقد تعلمنا اليوم أن البرمجة ليست مجرد كتابة نصوص، بل هي بناء منطق. الشروط هي التي تعطي البرمجيات ذكاءها في الاختيار، والحلقات هي التي تعطيها كفاءتها في العمل.[1]

في الدرس القادم، سنتعلم كيف نطبق هذه المنطق ليس فقط على متغيرات بسيطة، بل على **ملفات كاملة؛** حيث سنقوم بفتح ملفات نصية (Text Files) من جهاز الكمبيوتر ومعالجتها آلياً.[4]

■ **مهمة للتفكير:**

لو أردت تصميم برنامج يكتشف التنمر الإلكتروني في محادثات الطلاب، ما هي الشروط (if) التي ستضعها؟ وما هي القائمة التي ستحتاج للمرور عليها باستخدام (for)؟