مختصر كتابي

Set Up Claude: A Project, Instructions, Good Questions



كيف تطلب من Claude تلميحًا دون أن يسلّمك الحل؟

قريبًا ستقف أمام Test Case تقول `FAIL` ولا تفهم السبب. عندها تسأل **Claude**. التلميح هو **أول خطوة فقط في الحل، دون كود جاهز**. تعمل ذلك في موقع Claude أو في تطبيق Claude Desktop، والاثنان سواء. والكورس كله يكفيه الحساب المجاني، ولا تحتاج الاشتراك المدفوع Pro. وإذا أخبرك Claude أنك بلغت حد الـusage limit فانتظر حتى يتجدد ثم أكمل.

أول خطوة: أنشئ Project جديدًا باسم «مشاريع Python». الـProject يتذكر تعليماتك، فلا تكررها في كل محادثة. انسخ الـInstructions الجاهزة من «نظرة عامة» في صفحة الدرس كلها، وضعها في خانة الـInstructions. أهم ثلاثة أمور فيها:

1. يبدأ **بتلميح واحد لا بالحل**، ويسألك عن الـExpected Output إذا كان طلبك غير واضح، وإذا طلبت الحل أعطاك جزءًا منه مع شرحه لتكتب الباقي بنفسك.
2. لا يغيّر Test Case صحيحة ليخفي خطأ في الكود.
3. لا يقول إنه شغّل الكود إلا إذا شغّله فعلًا؛ وإلا قال إن كلامه تخمين واقترح Test Case تشغّلها أنت.

ثم سألنا داخل الـProject سؤالين، والردود حقيقية. السؤال الأول عام جدًا: «ساعدني في Python». بدأ الرد بكلام لطيف، ثم **سأل بدل أن يخمّن**: ماذا تريد أن تبني؟ وما النتيجة التي تنتظرها؟ وإن كان عندك كود فأرسله، وإن لم يكن فابدأ بمحاولة منك وهو يعطيك تلميحًا واحدًا.

السؤال الثاني محدد: عندي Function تحدد نتيجة جولة، وفي `rock` أمام `scissors` يجب أن يفوز اللاعب، فاشرح لي كيف أفكر دون كود. هنا تغيّر الرد إلى **تلميح بلا سطر كود**: اكتب قواعد اللعبة على ورقة، من يغلب من. ثم اسأل نفسك: هل الحركتان متماثلتان؟ وإن لم تكونا، فمن الفائز بينهما؟ ثم الحركة الفائزة لمن، اللاعب أم الكمبيوتر؟ لأن الـFunction تأخذ الحركتين بترتيب، ومكان كل منهما يفرق. وختم بسؤال: ماذا ترجع الـFunction في كل حالة، `player` أو `computer` أو `tie` ؟ وطلب أن تكتب محاولتك الأولى وترسلها ليراجعها معك. قد يختلف الكلام عندك، لكن الفكرة واحدة.

الفرق أن السؤال الثاني فيه الهدف والمثال بالضبط، وفيه أنك تريد طريقة التفكير لا الكود. ومن أول مشروع اكتب في سؤالك عن الكود **أربعة أشياء**: الهدف، ومحاولتك، والـExpected Output، والـOutput الذي ظهر فعلًا. أرسل كودك أنت، واطلب منه أن يشرح أول اختلاف بتلميح واحد.

وإذا رد Claude بالحل كله فاكتب له: ابدأ بتلميح واحد فقط، واتركني أحاول بنفسي. ثم اكتب الكود بنفسك وشغّله؛ لأن **الـTest Cases هي التي تقرر** أن الكود صحيح، لا كلام Claude. هو لم يشغّل الكود، فكلامه تخمين حتى نشغّله ونرى.

وهناك مكان آخر للسؤال: مساعد داخل الـLab يجيب عن أسئلة الدرس. أما مراجعة مشروعك فمكانها Claude. والاثنان غير مرتبطين؛ Claude لا يرى الـLab، وأنت من يرسل إليه كودك.

- اعمل Project واحدًا باسم «مشاريع Python» وضع فيه الـInstructions مرة واحدة.

- السؤال العام يجلب أسئلة، والسؤال المحدد بالهدف والمثال يجلب تلميحًا.
- اكتب في سؤالك: الهدف، ومحاولتك، والـExpected Output، والـOutput الفعلي.
- كلام Claude تخمين حتى تشغّل الكود، والـTest Cases هي الحكم.

جرّب في Lesson Lab تحت الدرس: أنشئ Project «مشاريع Python» وضع فيه الـInstructions، واسأل السؤال العام ثم سؤال `rock` و`scissors`، وقارن الردين في سطر، ثم اضغط «تشغيل» مرة في الـLab لترى `PASS` أمام الحالة الوحيدة.

الخلاصة: سؤال فيه الهدف والمثال والمحاولة والناتج يجلب تلميحًا مفيدًا، والـTest Cases هي الحكم الأخير.