مختصر كتابي

How Do We Know a Run Succeeded?



عندما تضغط «تشغيل» ويظهر ناتج، كيف تعرف أنه صحيح؟

تطلب سلمى برنامجًا صغيرًا: رسالة ترحيب باسم كل حاضر في المركز. تعطي البرنامج الاسم، فيكتب جملة الترحيب في ملف، ثم يقرؤها من الملف. وقبل أي كود قالت سلمى بالضبط ما تريده: إذا كان الاسم سلمى فالجملة «أهلا سلمى»، وإذا كان عمر فالجملة «أهلا عمر».

كل مثال من هذين يسمى **Test Case**: اسم نعطيه للبرنامج، وجملة ننتظر أن تعود منه تسمى **Expected Output**. وهذا هو الترتيب في كل مشروع: **الـTest Cases أولًا، ثم الكود**.

في الـLab يظهر ملف جاهز اسمه «كود البداية» ينقصه جزء تكتبه أنت. اضغط «تشغيل» دون أن تلمس شيئًا، واقرأ الـOutput سطرًا سطرًا. أول سطر فيه `TODO`، وهو تذكير مكتوب في الملف بالجزء المطلوب منك، وليس Error. بعده سطر لكل Test Case يبدأ بـ `FAIL` ثم اسمها بالإنجليزية ثم `AssertionError`، أي أن البرنامج قارن ما عاد منه بالـExpected Output فوجدهما مختلفين. وفي النهاية سطر `0/2 checks passed`.

هل يعني `FAIL` أن الـLab تعطل؟ لا. **الكود هو الناقص**، والـLab أخبرك بالضبط بما لم يُكتب بعد. في أعلى الملف Function فارغة اسمها `save_and_read`، وتحتها الـTest Cases، وتحتها كود ينادي الـFunction بكل اسم ويقارن، ثم يطبع `PASS` أو `FAIL` بجانب اسم كل حالة، ويعدّ كم حالة تحققت. فـ `PASS` يعني أن هذه الحالة تحققت، لا أن البرنامج كله انتهى.

```
from pathlib import Path
def save_and_read(name):
    path = Path("greeting.txt")
    path.write_text("أهلا " + name, encoding="utf-8")
    return path.read_text(encoding="utf-8")
```

الـFunction تكتب في الملف «أهلا» والاسم، ثم تقرأ الملف وتعمل return للجملة. الاسم عربي هنا لنتأكد أنه يعود سليمًا. والـ**encoding** هو الطريقة التي تُحفظ بها الحروف في الملف؛ فإذا كتبت بطريقة وقرأت بأخرى قد تعود حروف غريبة. لذلك يستخدم الـwrite والـread نفس الـencoding.

بعد الحل يظهر `PASS` أمام كل حالة. ثم اكتب «مرحبا» مكان «أهلا» في الـExpected Output لحالة سلمى فقط، فتظهر `FAIL` أمامها؛ لأن الـFunction ما زالت تكتب «أهلا سلمى». نفعل ذلك لنتأكد أن الـTest Case قادرة على الفشل فعلًا. وعند `FAIL` حقيقي لا نغيّر الـExpected Output، فهو آتٍ من طلب سلمى، بل ننظر في الكود عند أول اختلاف.

- الـTest Case اسم يُعطى للبرنامج مع الـExpected Output المنتظر منه.
- `TODO` تذكير بالجزء الناقص، وليس Error.
- `AssertionError` يعني أن الناتج الفعلي لا يساوي الـExpected Output.

- الكتابة والقراءة يجب أن تستخدما نفس الـencoding حتى يعود العربي سليمًا.
- عند `FAIL` نصلح الكود، ولا نعدّل الـExpected Output لينجح.

جرّب في Lesson Lab تحت الدرس: اكتب `save_and_read` حتى ترى `2/2 checks passed`، ثم ضع «مرحبا» مكان «أهلا» في حالة سلمى لترى `FAIL`، ثم أعدها كما كانت.

الخلاصة: التشغيل ينجح عندما يطابق الناتج الفعلي الـExpected Output الذي اتفقنا عليه قبل الكود.