مختصر كتابي

Arithmetic: Operators, Parentheses and Order



بأي ترتيب تحسب Python العمليات، وكيف تتحكم فيه بالأقواس؟

اشتريت دفترين سعر الواحد 30 جنيهًا، وكل دفتر يحتاج إلى غلاف بخمسة جنيهات. هل تجمع 30 و5 ثم تضرب الناتج في 2؟ أم تحسب ثمن دفتر واحد ثم تضيف ثمن غلافين؟ الجملتان مختلفتان، وكتابة الحساب يجب أن تعبّر عما تقصده بالضبط.

في Python، كما في الحساب العادي، للضرب والقسمة **أولوية** (Operator Precedence) على الجمع والطرح. و**الأقواس** (Parentheses) تجعلك تحدد جزءًا يُحسب قبل غيره.

```
print(30 + 5 * 2)
print((30 + 5) * 2)
print(9 / 2)
print(9 // 2)
print(9 % 2)
print(2 ** 3)
```

النتيجة الأولى 40: خمسة في اثنين تساوي عشرة، ثم نضيف 30. والنتيجة الثانية 70: الأقواس جعلت 30 و5 يُجمعان أولًا فصارا 35، ثم ضُربا في 2. وهذا هو الحساب الصحيح لقصتنا، لأننا اشترينا دفترين وغلافين. أما 40 فهي ثمن دفتر واحد مع غلافين.

علامة القسمة الواحدة `/` تعطي هنا 4.5. والعلامتان `//` تنفذان **القسمة الصحيحة** (Floor Division): يُقرَّب الناتج إلى الأسفل لأقرب عدد صحيح، فيظهر 4. ولا نسميها حذفًا للكسور بشكل عام، لأن التقريب إلى الأسفل يُحدث فرقًا مع الأرقام السالبة.

علامة النسبة `%` تعطي **باقي القسمة** (Remainder): في 9 على 2 أربع مجموعات كاملة ويبقى 1.
والنجمتان `**` تعنيان **الأس** (Exponent): 2 مرفوعة للقوة 3 تساوي 2 في 2 في 2، أي 8.

هذه أدوات مختلفة، فاستخدم منها ما يحل مشكلتك، ولا تحاول وضعها كلها في كل برنامج. وإذا لم يكن ترتيب عملية ما واضحًا للقارئ، فالأقواس توضحه حتى لو كانت Python ستحسبه صحيحًا بدونها. وتذكّر أن الكود قد يعمل ويعرض رقمًا، ومع ذلك يعبّر عن حساب غير المطلوب.

- الضرب والقسمة يُحسبان قبل الجمع والطرح.
- ما بين الأقواس يُحسب أولًا.
- `/` قسمة عادية، و `//` قسمة صحيحة تقرّب إلى الأسفل.
- `%` تعطي باقي القسمة، و `**` تعني الأس.
- اربط كل جزء من التعبير بمعناه في المشكلة.

جرّب في المحرر تحت الدرس: لديك 17 قطعة حلوى وتضع كل 5 قطع في علبة، فاعرض عدد العلب الكاملة وعدد القطع الباقية، ثم احسب تكلفة 3 وجبات سعر الواحدة 40 جنيهًا ومع كل واحدة إضافة بعشرة جنيهات.

الخلاصة: اكتب الحساب كما تقصده، واستخدم الأقواس لتحدد ما يُحسب أولًا.