

- "شكراً يا فرنك، يا سلوجو!"

أخذ سلوجو يكتب عدة دقائق على المحمول ثم خرج بهذا

```
$ python -c \
> "import sys; print sys.stdin.read().replace('at',
'op')," <INPUT
A cop sop on a mop with a hop.
```



ملاحظة: لقد طبعها سلوجو على سطر واحد لكننا فصلناها لتسهيل مشاهدتها
- "سحقاً!" قال وميرت ثم تابع، "ظننت أن فصل عبارة import جعلت من عمل الأسطر
الوحيدة (غير البديهيات) أمراً مستحيلاً. لقد نسيت موضوع الفاصلة المنقوطة. اشرح
برنامجك يا سلوجو."

- "الوحدة sys تحتوي خدمات النظام المختلفة. وفي هذه الحالة نحن نستعمل sys.stdin التي هي filehandle للمدخلات القياسية.
ونحن لن نحتاج sys.stdout لأن print تطبع هناك تلقائياً. أما read فهي طريقة تعيد كل محتويات الملف على شكل سلسلة نصية.
ولما كانت سلسلة نصية يمكننا استخدام الطريقة replace. عليها، التي تعمل على تعويض كل at ب op أما الفاصلة في النهاية فهي
لمنع print من إضافة سطر جديد لتكون النتيجة مطابقة للأصل. بايثون ليس بها بديل لـ W في بيرل لكننا لا نحتاجه. صيغة بايثون
واضحة وصارمة فالبنيات التي تجعل W يحذر منها تعتبر في بايثون أخطاء." ولم يذكر سلوجو أنه يوجد W- في بايثون لأن المبرمجين
بها لا يستعملونه. التحذيرات المبيته قليلة العدد لكنها تتعلق بحالات أقل صرامة. أما العلامة W فهي عندما يصدر برنامج تحذيرات
مخصصة من تعريفك، أو إذا كنت تريد أن يضايقك بالحديث عن المزايا المدمومة.
نظر الحكم إلى البرنامج وقال بعد لحظة، "حشر عبارة import على السطر نفسه مقبول، وإلا لما كانت المنافسة ممكنة حيث أن بايثون
تضع أغلب الخدمات في مكتبات كما أن read مقبولة مع أنها تبتلع كل المدخلات دفعة واحدة وليس سطرًا بعد سطر. ففي حالتنا
البديهية لا فرق بين الطريقتين. لكنك لا ترغب باستعمال ذلك مع ملف من عدة ميغابايتات. لكن علي أن أرفض replace. فهي تعويض
نص حرفي ولا تعطي مرونة التعبيرات النمطية، يمكنك المحاولة مرة أخرى إن رغبت."
عبس سلوجو وفتح محرر نصوص. وبعد دقيقتين أظهر برنامجاً آخرًا:

```
$ cat program.py
import re, sys; print re.sub(sys.argv[1], sys.argv[2], sys.stdin.read()),
```

```
$ python program.py at op <INPUT
A cop sop on a mop with a hop.
```

- "حسنٌ، فلتشرحه" قال وميرت ذلك.

شرح سلوجو "كما الذي قبله، لكن مع التعبيرات النمطية. re.sub هي وظيفة تكافئ S///q في بيرل. وهي تلقائياً تعمل على استبدال كل
المطابقات فلا حاجة للعلامة g أما sys.argv فهي منظومة (قائمة) بالمعاملات من سطر الأوامر ولأن sys.argv[0] هي اسم البرنامج
لذا قفزت عنها. أما sys.stdin.read() فهي تقرأ المدخلات على شكل سلسلة نصية كما في السابق"
- "لكن الصيغة من سطر الأوامر مختلفة"، قال فرنك معترضاً "حتى لو سمحنا بوضع البرنامج في ملف منفصل بدلاً من أن يكون
مضمناً في سطر الأوامر"

- "بايثون لا تحتاج صيغة S///g. إن صيغتها أسهل للمستخدم دون أن يكونوا بحاجة للخوض في السلاشات، خصوصاً إذا كان هناك سلاش
في أحد الأجزاء. أعلم أنكم في بيرل تستطيعون عمل S!!!g لكن طريقتي مريحة أكثر."
قال الحكم "أحب بساطة صيغة سطر الأوامر في بايثون. لكن برنامج البايثون مسهب مقارنة مع ذلك في بيرل. أنا أسميها عقدة (تعادل)"

- "نحترم رأي الحكم" قال وميرت ذلك مخرجاً مطبوعة أخرى

```
$ perl -i wpe's/at/op/g' *
```

- "سأطلب من فرنك شرحها."