Programming Fundamentals

Week 8

Reading

معظم البرامج التي شاهدناها حتى الآن هي برامج عابرة، بمعنى أنها تدوم لفترة قصيرة الوقت وإنتاج بعض المخرجات، ولكن عندما تنتهي، تختفي بياناتها. إذا قمت بتشغيل البرنامج مرة أخرى، فإنه يبدأ بسجل نظيف.

هناك برامج أخرى ثابتة: فهي تعمل لفترة طويلة (أو طوال الوقت)؛ يحتفظون بها على الأقل بعض بياناتهم في مخزن دائم (القرص الصلب، على سبيل المثال)؛ وإذا أغلقوا وإعادة التشغيل، يستأنفون من حيث توقفوا.

من أمثلة البرامج المستمرة أنظمة التشغيل، والتي تعمل تقريبًا في أي وقت جهاز كمبيوتر قيد التشغيل، وخوادم الويب، التي تعمل طوال الوقت، في انتظار وصول الطلبات على الشبكة.

إحدى أبسط الطرق التي تحافظ بها البرامج على بياناتها هي القراءة والكتابة ملفات نصية. لقد رأينا بالفعل برامج تقرأ الملفات النصية؛ في هذا الفصل سوف نرى البرامج التي تكتب لهم.

الملف النصي عبارة عن سلسلة من الأحرف المخزنة على وسيط دائم

مثل

Hard drive

Flash memory

Cd rom

لكتابة ملف، عليك فتحه باستخدام الوضع

W

مثل

```
>>> fout = open('output.txt', 'w')
```

إذا كان الملف موجودًا بالفعل، فإن فتحه في وضع الكتابة يؤدي إلى مسح البيانات القديمة والبدء من جديد، لذا كن حذرا! إذا كان الملف غير موجود، فسيتم إنشاء ملف جديد.

لارجاع كائن ملف و التعامل معه نستخدم وضع

Open

احد الطرق المستخدمة للتعامل مع الملفات و التعديل عليها (كتابة)

Write

بعد الانتهاء من التعديل يجب اغلاق الملف باستخدام

Close

اثناء الكتابة في ملف يجب ان يكون نوع القيم المضافة

String

لذلك يمكننا إضافة القيم ثم تحويلها الى نص

```
>>> x = 52
>>> fout.write(str(x))
```

البديل هو استخدام عامل التنسيق %. عند تطبيقها على الأعداد الصحيحة، % هو المعامل المشغل أو العامل. ولكن عندما يكون المعامل الأول عبارة عن سلسلة، يكون % هو عامل التنسيق. المعامل الأول هو سلسلة التنسيق، التي تحتوي على تسلسل تنسيق واحد أو أكثر، والتي تحدد كيفية تنسيق المعامل الثاني. والنتيجة هي سلسلة.

مثال

```
>>> camels = 42
>>> '%d' % camels
'42'
```

يتم تنظيم الملفات في دليل (وتسمى أيضًا "المجلدات"). كل برنامج قيد التشغيل لديه "الدليل الحالي"، وهو الدليل الافتراضي لمعظم العمليات.

يمكننا استخدام الوحدة التي تضم عدد من الوظائف للتعامل مع الملفات و المسارات و هي

Os

مثل

```
>>> import os
>>> cwd = os.getcwd()
```

>>> cwd

'/home/dinsdale'

هذا الدليل النصي يسمى مسار حيث يتواجد الملف

اسم الملف البسيط مثل

One.txt

يعتبر مسار نسبي لأنه

يتعلق الأمر بالدليل الحالي

من الوظائف المستخدمة أيضا وظيفة تحدد اذا كان الملف موجود داخل مسار معين ام لا مثل

>>> os.path.exists('memo.txt')

و يرجع قيمة صح او خطأ

لمعرفة ما اذا كان الملف دليل نستخجم الوظيفة

Isdir

او للتحقق ما اذا كان ملف عادي نستخدم

Isfile

(هذه بعض الوظائف التي تستخدم و ليست كل الوظائف)

اثناء التعامل مع الملفات سنواجه الكثير من الأخطاء فاذا حاولنا فتح ملف غير موجود سنجد

FileNotFoundError: [Errno 2] No such file or directory: 'bad_file'

هذا الخطأ يعني ان الملف غير موجود

او اذا حاولنا التعديل على ملف و نحن لا نملك الصلاحيات لذلك سنجد الخطأ

PermissionError: [Errno 13] Permission denied: '/etc/passwd'

و اذا حاولنا فتح مسار بدل ملف سنجد الخطا

IsADirectoryError: [Errno 21] Is a directory: '/home'

للتعامل مع هذه الأخطاء او لتجنبها يمكننا استخدام

Try , except

مثل

try:

fin = open('bad_file')

except:

print('Something went wrong.')

Discussion

المطلوب وصف كيف يمكن أن يساعد التقاط الاستثناءات في حل أخطاء الملفات. اكتب مثال بايثون الذي ينفذ معالجة الاستثناء لأي خطأ من أخطاء الملف.

يجب استخدام

Try, except

يجب ان نذكر بوضوح الاستثناء الذي نستخدمه و إضافة الكود الخاص بنا و مخرجاته

صف كيف يمكنك التعامل مع خطأ الملف إذا كنت تكتب برنامج إنتاج كبير. يجب أن تكون هذه الأوصاف أفكارًا عامة باللغة الإنجليزية

Programming assignment

اكتب برنامجًا لقراءة عناصر القاموس من ملف ثم اكتب القاموس المقلوب إلى ملف استنادا الى التالي:

1. ملف الإدخال الخاص بالقاموس الأصلي (يحتوي على ستة عناصر على الأقل).

2. برنامج بايثون الذي استخدمته للقراءة من ملف وعكس القاموس والكتابة إلى ملف مختلف. (تحتاج إلى إنشاء ملف قاموس وعكسه إلى ملف آخر).

3. ملف الإخراج للقاموس المقلوب الخاص بك.

(الشرح يجب ان يكون 200 كلمة على الأقل)

(Don’t forget to take review quiz)

Good Luck!