Programming Fundamentals

Week 4

Fruitful Functions

تعلمنا خلال الأسابيع السابقة تعريف وظيفة تقوم بعملية حسابية او تطبع جملة في المخرجات

هذا الأسبوع سنتعلم كتابة وظيفة ترجع قيمة . بعد كتابة الوظيفة يتم استدعاء الدالة و التي من المفترض ان ترجع قيمة

عادة تكون عبارات الارجاع موجودة في عبارات شرطية حيث ترجع قيمة معينة اذا تحقق شرط من الشروط

Incremental development

كلما كان البرنامج أطول كلما زادت الأخطاء لذلك تعتبر عملية التطوير المتزايد هي الحل و التي تعني ان يتم تقسيم الكود و تجريب كل قسم صغير منفردا . و نستخدم متغيرات لحفظ القيم . هذه العملية تسهل قراءة الكود و اكتشاف الأخطاء

يمكننا استدعاء دالة واحدة من داخل أخرى. مثلا اذا عرفنا وظيفة تحسب نصف قطر الدائرة و داخلها متغير يحفظ القيمة و داخل نفس الوظيفة متغير يحفظ قيمة مساحة الدائرة

نكتب دالة تأخذ نقطتين، مركز الدائرة ونقطة على الدائرة

المحيط، وحساب مساحة الدائرة.

الخطوة الأولى هي إيجاد نصف قطر الدائرة، وهي المسافة بينهما

النقطتان. لقد كتبنا للتو دالة

الخطوة التالية هي إيجاد مساحة الدائرة التي لها نصف القطر؛

تعتبر المتغيرات المؤقتة نصف القطر والنتيجة مفيدة للتطوير وتصحيح الأخطاء

Boolean Functions

يمكن للوظائف إرجاع القيم المنطقية، والتي غالبًا ما تكون ملائمة لإخفاء الاختبارات المعقدة داخل الوظائف.

```
def is_divisible(x, y):
if x % y == 0:
return True else: return False
```

من الأخطاء التي من الممكن ان تواجه أي مبرمج هي حلقة تكرار لا نهائية

مثلا اذا عرفنا وظيفة تقوم بالعد العكسي و اعطينا هذه الوظيفة قيمة معينة دون ان نحدد شرط التوقف سنواجه هذا الخطأ

RuntimeError: Maximum recursion depth exceeded

كما ان هذا الخطأ شائع عند التعامل مع المضروب

Factorial

تحنبا لهذا الخطأ يمكننا استخدام وظيفة جاهزة في لغة بايثون

Isinstance

و التي تتحقق من نوع القيمة قبل تنفيذ الوظيفة

مثال

```
def factorial(n):
if not isinstance(n, int):
print('Factorial is only defined for integers.')
return None
elif n < 0:
print('Factorial is not defined for negative integers.')
return None elif n == 0:
return 1 else:
return n * factorial(n-1)
```

glossary

يرجى مراجعة الصطلحات داخلها كل أسبوع

يؤدي تقسيم برنامج كبير إلى وظائف أصغر إلى إنشاء نقاط فحص طبيعية لتصحيح الأخطاء.

إذا كانت الوظيفة لا تعمل، فهناك ثلاثة احتمالات يجب أخذها في الاعتبار:

- هناك خطأ ما في الوسيطات التي تحصل عليها الدالة؛ شرط مسبق تم انتهاكه.
- هناك خطأ ما في الوظيفة؛ تم انتهاك الشرط اللاحق.
- هناك خطأ ما في القيمة المرتجعة أو في طريقة استخدامها.

لاستبعاد الاحتمال الأول، يمكنك إضافة عبارة طباعة في بداية الأمر وظيفة وعرض قيم المعلمات (وربما أنواعها). أو يمكنك ذلك كتابة التعليمات البرمجية التي تتحقق من الشروط المسبقة بشكل صريح.

إذا كانت المعلمات تبدو جيدة، أضف عبارة طباعة قبل كل عبارة إرجاع و عرض قيمة الإرجاع.

Discussion

يسرد القسم 6.9 تصحيح الأخطاء في كتابك المدرسي ثلاثة احتمالات يجب مراعاتها في حالة عدم عمل إحدى الوظائف.

•

صف كل احتمال بكلماتك الخاصة.

• حدد "الشرط المسبق" و"الشرط اللاحق" كجزء من الوصف الخاص بك.

• أنشئ مثالك الخاص لكل احتمالية في كود بايثون. قم بإدراج الكود الخاص بكل مثال، بالإضافة إلى نموذج الإخراج من المحاولة

لتشغيله.

Assignment

المطلوب تطوير دالة تحسب طول الوتر في مثلث قائم الزاوية مع الأخذ بعين الاعتبار طول الضلعين الآخرين كوسيطات

استخدام التطوير المتزايد لإنشاء الوظيفة الضرورية وتوثيق كل مرحلة من مراحل عملية التطوير

اختبار الوظيفة باستخدام وسائط مختلفة وتسجيل المخرجات

مع شرح لكل مرحلة من مراحل التطوير، بما في ذلك الكود وأي مدخلات ومخرجات اختبارية.

ناتج الوتر (3،4).

• إخراج نداءين إضافيين للوتر مع وسيطات مختلفة.

Part two

لنفترض اننا ك مبرمجين نخطط لانشاء بورتفوليو خاص بكل شخص منا حيث يعرض مهاراتنا و قدرتنا على كتابة برامج صحيحة و فعالة . ك جزء من ملفنا علينا ان ننشىء وظيفة داخل كود تقوم بعمليات حسابية باستخدام نهج التطوير المتزايد

ستقوم بتوثيق كل مرحلة من مراحل عملية التطوير، بما في ذلك الكود وأي مدخلات ومخرجات اختبارية

يجب شرح الكود ومخرجاته و شرح لكل مرحلة من مراحل التطوير بما في ذلك الكود وأي اختبار المدخلات والمخرجات. إخراج ثلاث مكالمات إلى وظيفتك باستخدام وسائط مختلفة.

الجزء الوصفي من إجابتك يجب أن لا يقل عن 200 كلمة.

Good Luck!