OCTOBER 20 – 26, 2019

المبادرة السعودية للمطورين

مسار الـPython

مشرفي المسار:

عبدالله عوده – انتصار النصار – رؤى كردي – لينا المصعبي

ملاحظات قبل بدء الدروس:

- على المتدربين نشر كل يوم الجزئية التي تم كتابتها من النص البرمجي في الـ Github تحت Topic بعنوان saudidevorg

كما تم توضيحه في دروس الـ Github سابقاً

- على المتدربين نشر كل يوم مقدار التقدم وصورة لما تم تعلمه وتطبيقه على Twitter تحت الهاشتاقات:

#المبادرة_السعودية_للمطورين

#100_يوم_برمجة

#100DaysOfCode

تمنياتنا لك بالتوفيق

المبادرة السعودية للمطورين

اليوم الثاني والستون

مدير الحِزم في لغة البايثون

Python PIP

ما هو مفهوم الـ PIP

➢ What is PIP?

PIP is a package manager for **Python** packages, or modules if you like.

هي أداة تساعدك في تحميل/تضمين أي حزمة، PIP هو مدير الحزم الخاص بلغة بايثون لتحميل الحزم أو المكتبات من بايثون

Note: If you have **Python** version 3.4 or later, **PIP** is included by default.

في حال كنت تستخدم الإصدار ٣.٤ أو إصدارات أحدث ، فإن أداة PIP تكون ملحقة بالبايثون بشكل تلقائي أثناء تثبيته

فيمكنك استخدامها مباشرة دون الحاجة لتثبيتها

ما هو مفهوم الـحزمة

➢ What is a Package?

A package contains all the files you need for a module.

Modules are **Python** code libraries you can include in your project.

الحزمة تحتوي على الملفات التي يضعها المطوّر في مجلد واحد لجعلها قابلة للتحميل/التضمين

إذا هي عبارة عن مجلد تحتوي على وحدة أو أكثر

تعلمت سابقا كيف تقوم ببناء وحدات/موديول وذلك بإنشاء ملف بايثون وبداخله تعريف للأصناف والدوال ... إلخ

معرفة ما إذا كانت PIP موجودة على جهازك أم لا

➢ Check if PIP is Installed

Navigate your **command line** to the location of **Python's** script directory, and type the following: `pip --version`

قم بكتابة الأمر pip --version في موجه الأوامر الخاص بجهازك

وستظهر لك إصدار النسخة المنصبة منها، فإذا كانت PIP مثبتة على جهازك فإنه يمكنك الآن استخدامها لتحميل أي حزمة

Check **PIP** version

```
C:\Users\Your Name\AppData\Local\Programs\Python\Python36-32\Scripts>pip --version
```

١/ أفتح موجه الأوامر command line

٢/ اكتب الأمر python --version

للتأكد من إصدار بايثون الذي في جهازك

٣/ ثم اكتب الأمر pip --version

للتأكد من وجود الأداة PIP

إذا كان اصدارك هو بايثون ٣

١/ أفتح موجه الأوامر command line

٢/ قم بكتابة الأمر python3 --version

للتأكد من إصدار بايثون

٣/ ثم اكتب الأمر pip3 --version

للتأكد من وجود الأداة PIP

➢ Install PIP

تثبيت PIP

If you do not have **PIP** installed, you can download and install it from this page: https://pypi.org/project/pip/

في حال لم تكن PIP مثبتة على جهازك، يمكنك تثبيتها من الموقع الرسمي

وستجد أيضا في الموقع الخطوات الصحيحة الواجب عليك اتباعها لتثبيت PIP بشكل صحيح

➢ Download a Package

تحميل/تنزيل الحزم

لتحميل أي حزمة قم بفتح موجه الأوامر في جهازك، ثم استدعاء الأمر PIP ثم كلمة install ثم اكتب اسم الحزمة التي تريد

Downloading a package is very easy.

Open the **command line** interface and tell **PIP** to download the package you want.

Navigate your **command line** to the location of **Python's** script directory, and type the following: **pip install [name of package]**

Example

انظر المثال التالي

Download a package named "**camelcase**"

قمنا بتحميل وتثبيت الحزمة camelcase

```
C:\Users\Your Name\AppData\Local\Programs\Python\Python36-32\Scripts>pip install camelcase
```

Now you have downloaded and installed your first package!

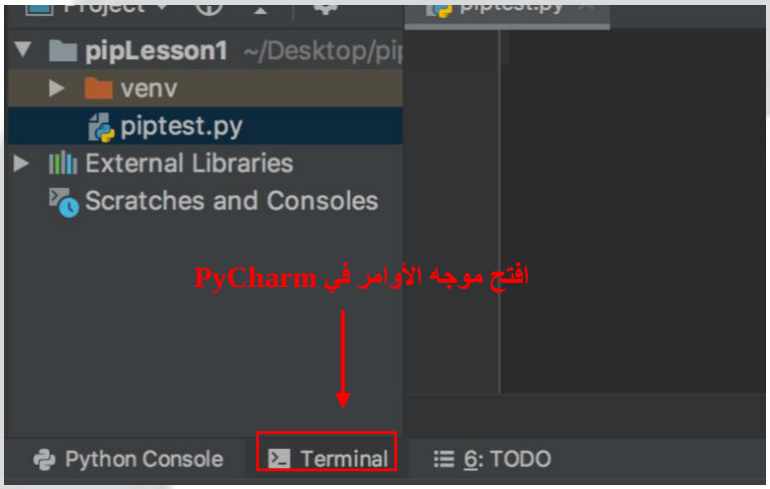
الآن لقد قمت بتحميل/تثبيت أول حزمة

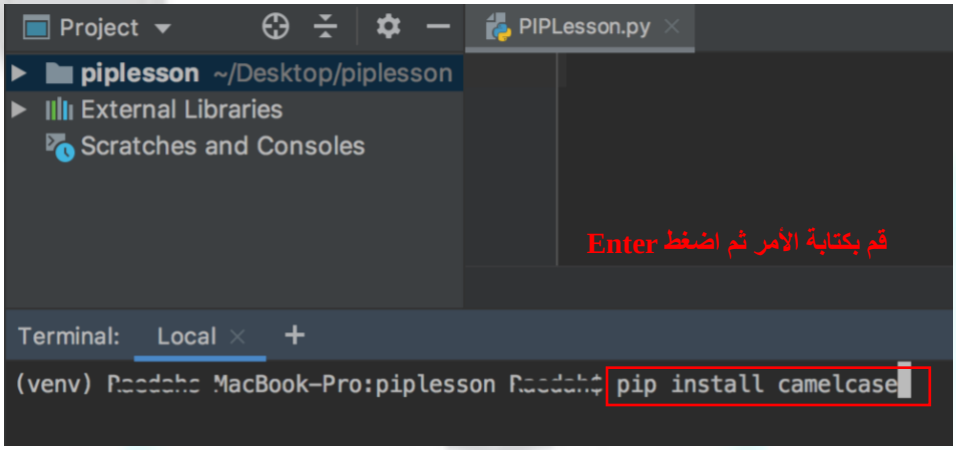
ملاحظة تأكد من أن جهازك متصل بالإنترنت، لأن أداة PIP تقوم بتحميل الحزمة من الإنترنت

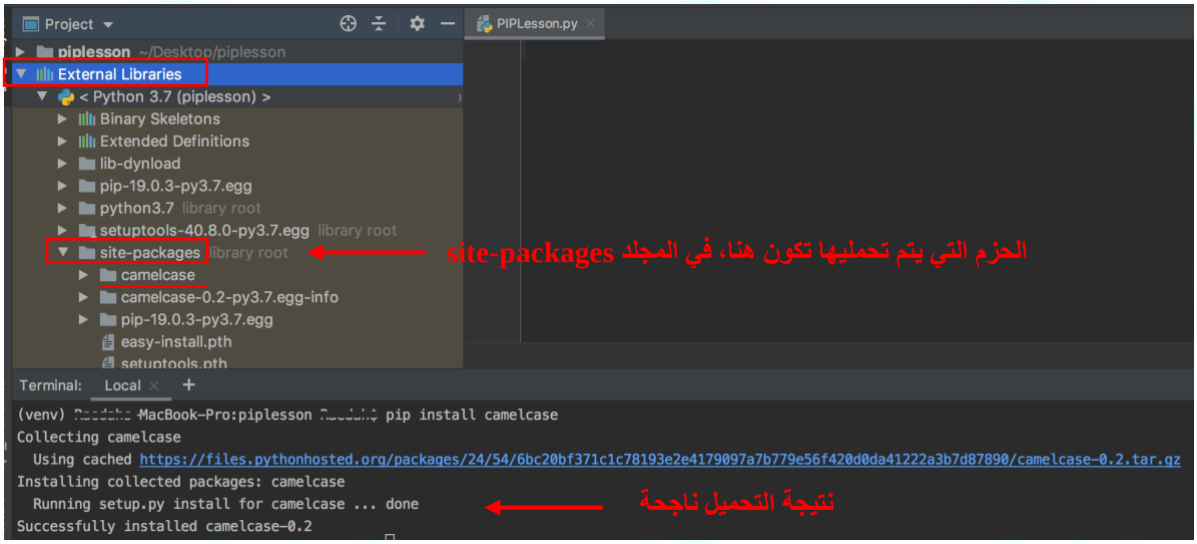
يمكنك أيضا استخدام موجه الأوامر الموجود في البرنامج PyCharm لتحميل الحزمة camelcase

انظر المثال التالي:

Example







➢ Using a Package

استخدام/تضمين الحزم

Once the package is installed, it is ready to use. Import the "camelcase" package into your project.

بمجرد تحميلك وتنزيلك للحزمة، تكون جاهزة للاستخدام يمكنك استخدامها وتضمينها في برنامجك

Example

في المثال هنا

Import and use "camelcase".

قمنا باستدعاء واستخدام الحزمة camelcase التي قمنا بتحميلها وتثبيتها

الحزمة camelcase تحتوي على صنف اسمه CamelCase

وهذا الصنف يحتوي على دالة hump()

```
import camelcase

c = camelcase.CamelCase()

txt = "lorem ipsum dolor sit amet"

print(c.hump(txt))

#This method capitalizes the first letter of each word.
```

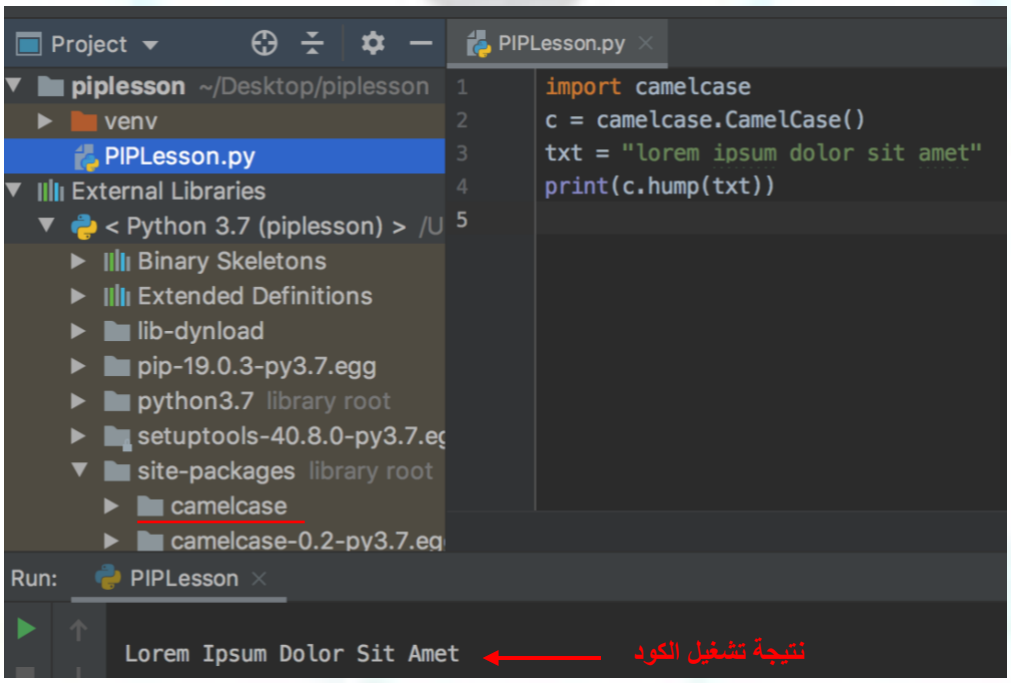
الدالة تقوم بتحويل أول حرف في كل كلمة

إلى حرف كبير Capital Letter

```
C:\Users\My Name>python demo_camelcase.py
Lorem Ipsum Dolor Sit Amet
```

نتيجة تشغيل الكود

تطبيق نفس المثال في برنامج PyCharm



➢ Find Packages

إيجاد والبحث عن الحزم

Find more packages at https://pypi.org/.

للاطلاع على المزيد من حزم بايثون قم بزيارة هذا الموقع

و كما رأيت سابقا طرق تحميل وتضمين واستخدام الحزمة camelcase فإنه ينطبق على كل الحزم الأخرى

➢ Remove a Package

إلغاء وحذف الحزم

Use the **uninstall** command to remove a package

استخدم الأمر uninstall لإلغاء وإزالة تثبيت أي حزمة قمت بتحميلها

Example

Uninstall the package named "camelcase"

هنا قمنا بإلغاء تثبيت الحزمة camelcase نكتب في موجه الأوامر الأمر **pip uninstall** ثم اسم الحزمة

```
C:\Users\Your Name\AppData\Local\Programs\Python\Python36-32\Scripts>pip uninstall camelcase
```

The **PIP** Package Manager will ask you to confirm that you want to remove the camelcase package

Press y and the package will be removed.

سيتم عرض مسار جميع ملفات الحزمة التي تريد إلغاءها، ثم يسألك ما إذا كنت متأكدا من عملية حذف الحزمة أم لا ؟

اكتب الحرف y للموافقة على إلغاء الحزمة

```
Uninstalling camelcase-02.1:
  Would remove:
    c:\users\Your Name\appdata\local\programs\python\python36-32\lib\site-packages\camecase-0.2-py3.6.egg-info
    c:\users\Your Name\appdata\local\programs\python\python36-32\lib\site-packages\camecase\*
Proceed (y/n)?
```

```
Terminal:  Local  +
(venv) [illegible]-MacBook-Pro:pip [illegible]$ pip uninstall camelcase
Uninstalling camelcase-0.2:
  Would remove:
    /Users/[illegible]/Desktop/pip/venv/lib/python3.7/site-packages/camelcase-0.2-py3.7.egg-info
    /Users/[illegible]/Desktop/pip/venv/lib/python3.7/site-packages/camelcase/*
Proceed (y/n)? y
  Successfully uninstalled camelcase-0.2
(venv) [illegible]s-MacBook-Pro:pip [illegible]$
6: TODO  Terminal  Python Console
```

تطبيق المثال في PyCharm

عرض الحزم

➢ List Packages

Use the **list** command to list all the packages installed on your system.

لعرض جميع الحزم التي قمت بتحميلها وتثبيتها والموجودة لديك ، اكتب الأمر **pip list** في موجه الأوامر

Example

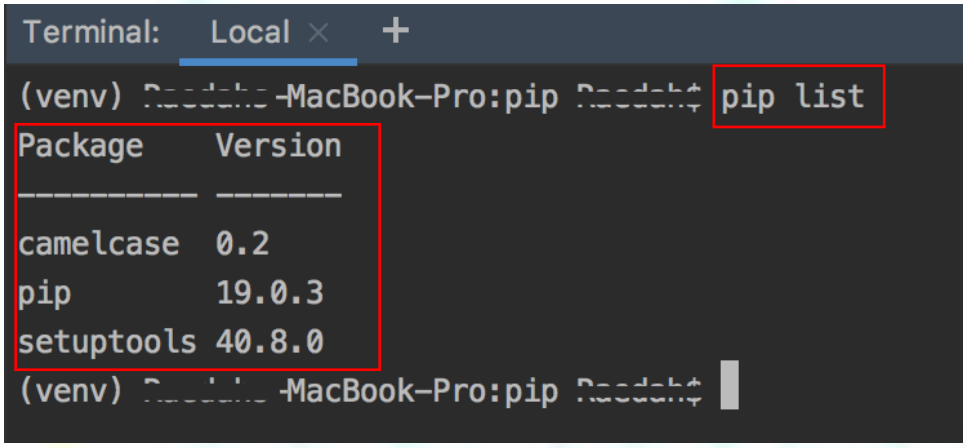
List installed packages.

```
C:\Users\Your Name\AppData\Local\Programs\Python\Python36-32\Scripts>pip list
```

```
Package          Version
-----------------------
camelcase        0.2
mysql-connector  2.1.6
pip              18.1
pymongo          3.6.1
setuptools       39.0.1
```

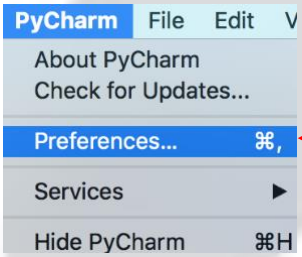
ظهرت جميع الحزم

تطبيق المثال في **PyCharm**



بدلا من استخدامك موجه الأوامر لتحميل الحزم، سواءا موجه الأوامر الخاص بنظام التشغيل لجهازك أو الخاص بـ PyCharm

يمكنك استخدام مدير الحزم PIP الموجود في PyCharm لتحميل/تثبيت الحزم، اتبع الخطوات التالية:

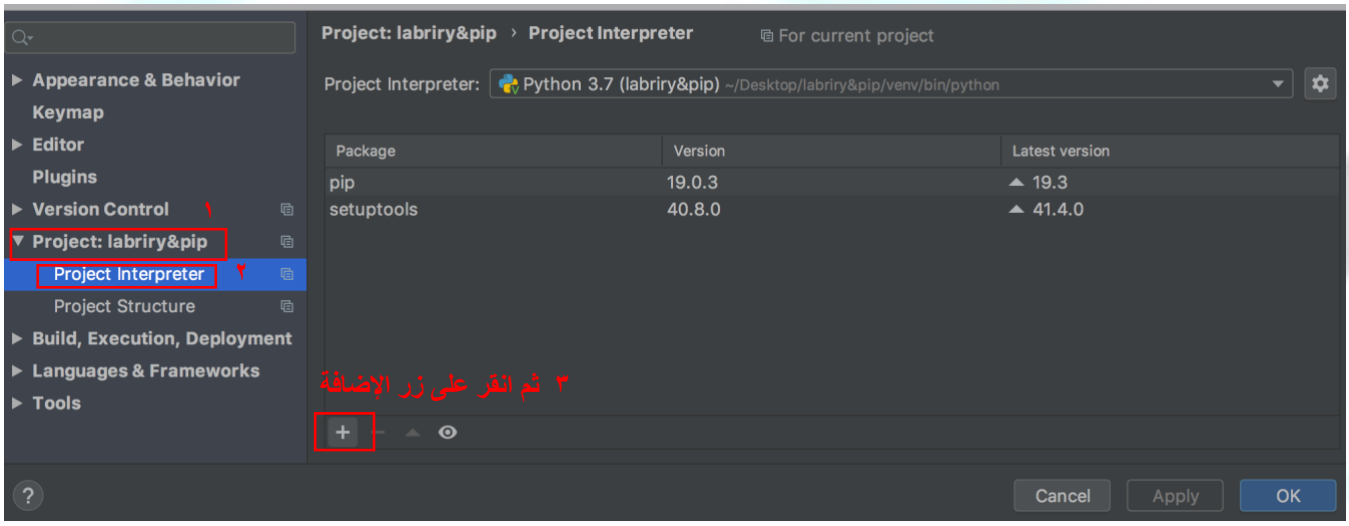


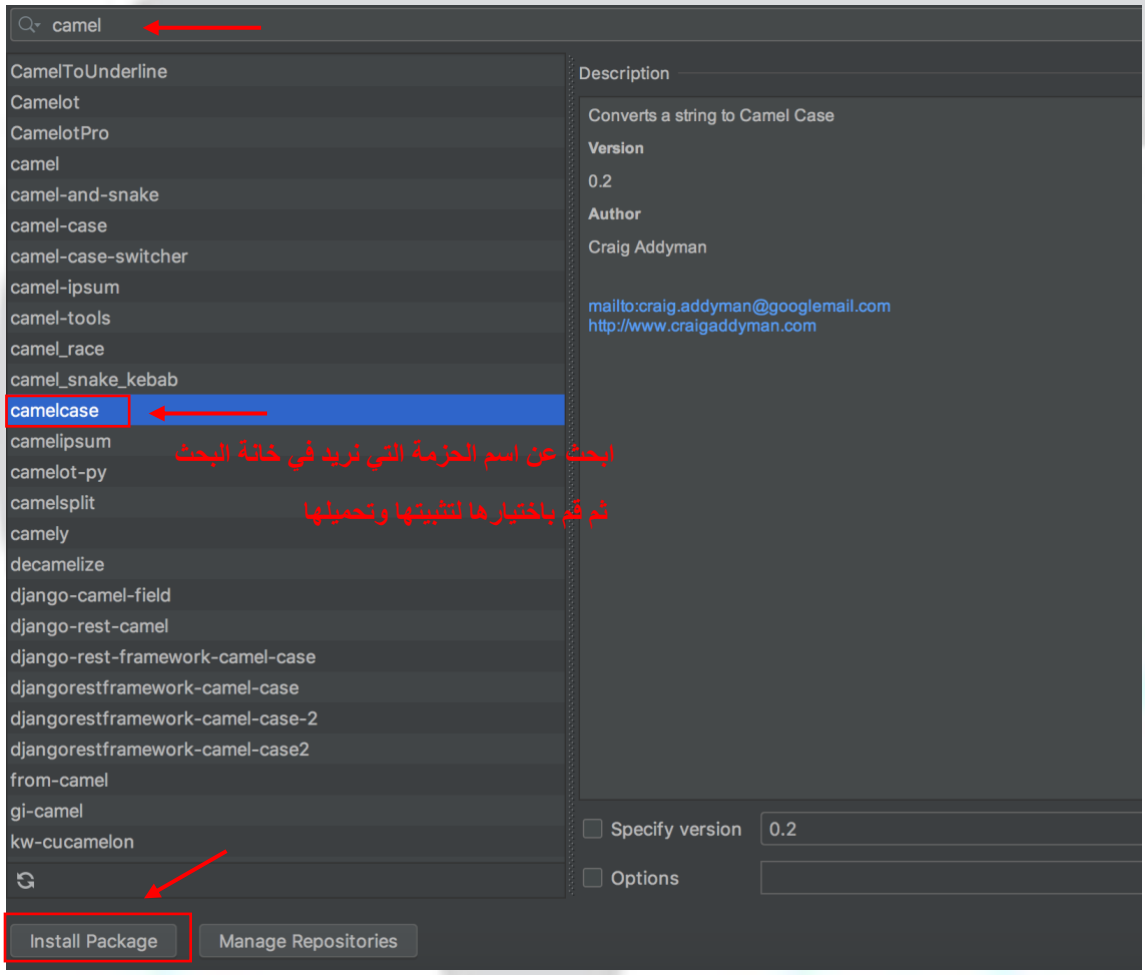
أفتح برنامج PyCharm ثم اختر

١ Project: labriry&pip اسم المشروع الذي قمت بإنشائه

٢ Project Interpreter ثم اختر

٣ add ثم إضافة





راجع الروابط التالية، وأيضا يمكنك البحث عن موضوع install package in pycharm

- How to install library in Pycharm?
- How to Install Python PIP Packages in PyCharm

أتممت درسك بنجاح!

روابط قد تهمك

Useful links

- What is pip?
- How to install Python on Mac OS
- How to install pip on Mac OSX [SCREENCAST]
- How to Install PIP for Python
- How to Install Python PIP on Windows, Mac, and Linux
- Install python PIP, Requests and Beautiful soup for WINDOWS (in 5 minutes)
- How to Install Python PIP on Windows 8 / Windows 10
- How to install pip on Windows (The easy way)
- package installer
- pip - The Python Package Installer
- Installation
- How do I install pip on macOS or OS X?
- Python Beginner Tutorial: Install Python on Windows and Install Packages with pip
- مقدمة إلى ادارة الحزم البايثون [PIP]
- أدوات إدارة الحزم والبرمجيات البايثونية (دليل الحيران)
- Python PIP How To - Install and Uninstall Packages
- Modules & pip in Python | Python Tutorial #20

طبّق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم الثالث و الستون

إدخال البيانات من المستخدم في لغة البايثون

Python Command Line Input

➢ Command Line Input

إدخال البيانات

Python allows for command line input.
That means we are able to ask the user for input.

سابقا كنا نكتب الأكواد ونقوم بتجربتها وتنفيذها وتحديد المتغيرات قبل تشغيل البرنامج

في بايثون يمكنك التفاعل مع المستخدم، بحيث يتم إدخال قيم للمتغيرات من المستخدم لكتابة برنامج أكثر تفاعلا

The method is a bit different in **Python 3.6** than **Python 2.7**.
Python 3.6 uses the `input()` method.
Python 2.7 uses the `raw_input()` method.

يتم استخدام دالة جاهزة وهي `input()` في الإصدارات الحديثة

أما الإصدارات السابقة تُستخدم الدالة `raw_input()`

Example

في المثال التالي أنشأنا برنامج يقوم بطلب الاسم من المستخدم وتخزينه في متغير، ثم يتم عرضه له

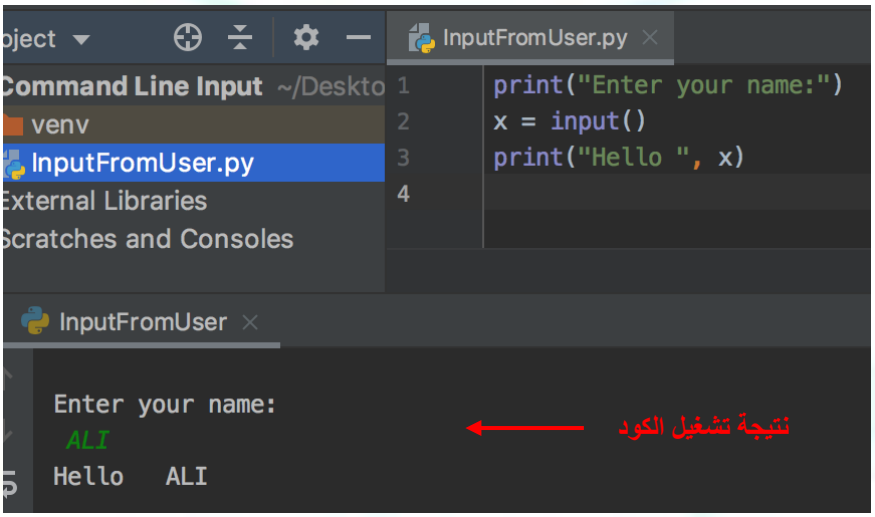
The following example asks for the user's name, and when you entered the name, the name gets printed to the screen.

Python 3.6

```
print("Enter your name:")
x = input()
print("Hello ", x)
```

Python 2.7

```
print("Enter your name:")
x = raw_input()
print("Hello ", x)
```



تطبيق المثال في PyCharm

Save this file as demo_string_input.py, and load it through the **command line**

قم بحفظ الكود السابق بملف باسم demo_string_input.py وباستخدام موجه الأوامر قم باستدعاء وتحميل الملف

```
C:\Users\Your Name>python demo_string_input.py
```

Our program will **prompt** the user for a string

سيطلب البرنامج من المستخدم ادخال الاسم

حيث أن **Prompt** هي الرسالة التي تظهر للمستخدم

```
Enter your name:
```

The user now enters a name

سيقوم المستخدم بإدخال الاسم

```
Linus
```

Then, the program prints it to screen with a little message

ثم تتم طباعة رسالة الترحيب مع الاسم الذي ادخله المستخدم

```
Hello Linus
```

مبرمج الغد!

أتممت درسك

روابط قد تهمك

Useful links

- Input function in python | دالة الإدخال في بايثون
- #18 Python Tutorial for Beginners | User input in Python | Command Line Input
- Getting Input From Users | Python | Tutorial 8
- input الادخال في البايثون
- Learn Python in Arabic #10 - شرح بايثون بالعربي استلام بيانات من المستخدم INPUT OR READ DATA

طبّق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم الرابع والستون

التعامل مع الاستثناءات في لغة البايثون

Python Try-Except

هنا في هذا الدرس سنتكلم عن نوع من أنواع الأخطاء وهي الأخطاء الاستثنائية، وليس الأخطاء اللغوية والمنطقية

The **try** block lets you test a block of code for errors. الجملة **try** توضع بداخل الكود الذي قد يتسبب بحدوث خطأ

The **except** block lets you handle the error. الجملة **except** لمعالجة والتعامل مع الخطأ الذي حدث في الجملة **try**

The **finally** block lets you execute code, regardless of the result of the try- and except blocks.

الجملة **finally** تجعلك تنفذ الكود بغض النظر حدث خطأ أم لا في الجملة **try**

بمجرد وضع الجملة **try** داخل الكود، يكون إجباريا وضع الجملة **except** أو **finally** أو كلاهما

أما الجملة **except** يمكنك بعدها وضع الجملة **finally** أو الجملة **else** ولا يمكنك وضعهما كلامها بنفس الوقت

➢ Exception Handling معالجة الأخطاء

When an error occurs, or exception as we call it, **Python** will normally stop and generate an error message. These exceptions can be handled using the **try** statement.

عند حدوث خطأ أو استثناء سيتوقف البرنامج ويظهر لك بأنه هناك خطأ، هنا الاستثناءات تقوم بالتعامل مع هذه الأخطاء باستخدام الجملة **try** بطريقة أنها تضمن إذا حدث الخطأ المتوقع أو أي خطأ فإن البرنامج لن يتم إغلاقه بشكل مفاجئ

الاستثناءات هي الأخطاء التي تحدث أثناء تشغيل البرنامج وتؤدي إلى تعليق البرنامج أو إيقافه فجأة

Example

في المثال التالي

The try block will generate an exception, because x is not defined

استخدمنا الجملتين try و except

ووضعنا خطأ متعمدا، وهو أنه لم نقم بتعريف المتغير x أو إعطائه قيمة

إذا سيحدث خطأ داخل الجملة try وسيكون هناك استثناء وانتقال إلى الجملة except عند حدوث الخطأ

ويتم تنفيذ الأوامر الموجودة في جملة except وسيعمل البرنامج

```
#The try block will generate an error, because x is not defined:

try:
  print(x)
except:
  print("An exception occurred")
```

```
C:\Users\My Name>python demo_try_except.py
An exception occurred
```

نتيجة تشغيل الكود

Since the try block raises an error, the except block will be executed.

Without the try block, the program will crash and raise an error

Example

This statement will raise an error, because x is not defined

```
#This will raise an exception, because x is not defined:

print(x)
```

نفس المثال السابق وهنا خطأ استثنائي

قمنا بطباعة قيمة المتغير ولم يتم تعريفة أو اعطائه قيمة

إذا سيحدث خطأ عند التشغيل، لأن المتغير لا يحتوي على قيمة

```
C:\Users\My Name>python demo_try_except_error.py
Traceback (most recent call last):
  File "demo_try_except_error.py", line 3, in <module>
    print(x)
NameError: name 'x' is not defined
```

نتيجة تشغيل الكود
Error

عدة استثناءات

➢ Many Exceptions

You can define as many exception blocks as you want, e.g. if you want to execute a special block of code for a special kind of error.

يمكنك وضع أكثر من جملة except للتعامل ومعالجة كل أنواع المشاكل التي قد تحدث

Example

Print one message if the **try** block raises a **NameError** and another for other errors.

هنا في المثال استخدمنا الجملتين try و except

ووضعنا الخطأ المتعمد أنه لم يتم تعريف المتغير x أو إعطائه قيمة بالتالي سيحدث خطأ داخل الجملة try

سيكون هناك استثناء وانتقال إلى الجملة except الأولى عند حدوث الخطأ من نوع NameError

وهو محاولة عرض قيمة متغير لم يتم تعريفه مسبقا، وسيتم تنفيذ جملة الطباعة

أما لو كان الخطأ هو نوع آخرغير NameError سيتم تنفيذ كود جملة except الأخرى

```
#The try block will generate a NameError, because x is not defined:

try:
  print(x)
except NameError:
  print("Variable x is not defined")
except:
  print("Something else went wrong")
```

```
C:\Users\My Name>python demo_try_except2.py
Variable x is not defined
```

نتيجة تشغيل الكود

➢ Else

الجملة else

You can use the else keyword to define a block of code to be executed if no errors were raised

الجملة else تُستخدم داخل الكود في حال لم يحدث خطأ أثناء تنفيذ الجملة try

Example

In this example, the try block does not generate any error.

```
#The try block does not raise any errors, so the else block is executed:

try:
  print("Hello")
except:
  print("Something went wrong")
else:
  print("Nothing went wrong")
```

هنا في المثال استخدمنا الجمل الثلاث try و except و else

ولا يوجد خطأ داخل الكود

الجملة else يتم تنفيذ الأوامر التي فيها في حال لم يكن هناك خطأ في الجملة try

```
C:\Users\My Name>python demo_try_except3.py
Hello
Nothing went wrong
```

نتيجة تشغيل الكود

➢ Finally

الجملة finally

The **finally** block, if specified, will be executed regardless if the **try** block raises an error or not.

الجملة **finally** تجعلك تنفذ الكود بغض النظر حدث خطأ أم لا في الجملة **try**

Example

```
#The finally block gets executed no matter if the try block raises any errors or not:

try:
  print(x)
except:
  print("Something went wrong")
finally:
  print("The 'try except' is finished")
```

هنا في المثال استخدمنا الجمل الثلاث **try** و **except** و **finally**

سيحدث خطأ داخل الجملة **try** لم يتم تعريف المتغير **x** أو إعطائه قيمة

سيكون هناك استثناء وانتقال إلى الجملة **except** عند حدوث الخطأ

ويتم تنفيذ الأوامر الموجودة في جملة **except**

و الجملة **finally** يتم تنفيذ الأوامر التي بداخلها سواءا كان هناك خطأ أم لا في الجملة **try**

```
C:\Users\My Name>python demo_try_except4.py
Something went wrong
The 'try except' is finished
```

نتيجة تشغيل الكود

أتممتَ درسك بنجاح!

تابع التقدّم

روابط قد تساعدك

Check the links below

- التعامل مع الاستثناءات باستخدام عبارة try-except
- Exceptions
- #63 Python Tutorial for Beginners | Exception Handling
- شرح - 24 try, except - تعلم البرمجة بلغة بايثون
- Learn Python in Arabic #70 - معالجة الأخطاء و الاستثناءات Errors Exceptions Python
- #16 تعلم بايثون 3 - التعامل مع الاخطاء Error Handling
- Python Programming #11 - Try and Except + Commenting
- Python tutorial Exception مفهوم الـ
- Exceptions in Python || Python Tutorial || Learn Python Programming
- 28- Python|| Exceptions اقتناص الاخطاء
- Try / Except | Python | Tutorial 27

طبّق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم الخامس والستون

آلية تنسيق السلاسل النصية في لغة البايثون

Python String Formatting

للتأكد من أن السلسلة سوف تظهر كما هو متوقع نستطيع تنسيق المخرج أو النتيجة بإستخدام الدالة format()

To make sure a string will display as expected, we can format the result with the format() method.

➢ String format()

دالة تنسيق النص/السلسة النصية

The format() method allows you to format selected parts of a string.

Sometimes there are parts of a text that you do not control, maybe they come from a database, or user input? .To control such values, add placeholders (curly brackets {}) in the text, and run the values through the format() method.

الدالة format() تسمح لك بتنسيق أجزاء مختارة من قبلك في السلسلة النصية.

في بعض الأحيان يكون هناك أجزاء نصية لا تتحكم بها, قد تكون جاءت من قاعدة البيانات, او من المستخدم.
للتحكم في هذه القيم أضف الأقواس المعكوفة {} في النص, ثم شغل هذه القيم باستخدام الدالة format()

Example

في المثال هنا أستخدمنا الدالة format() والأقواس المعكوفة لإظهار القيمة الرقمية للسعر

Add a placeholder where you want to display the price.

نقوم بوضع حقول قابلة للاستبدال وذلك عن طريق المُنسقّات

بوضع الأقواس المعكوفة في السلسة النصية

ثم تمرير القيمة إلى الدالة format()

```
price = 49
txt = "The price is {} dollars"
print(txt.format(price))
```

```
C:\Users\My Name>python demo_string_formatting1.py
The price is 49 dollars
```

نتيجة تشغيل الكود

تستطيع اضافة الفواصل العشرية بين الأقواس المعكوفة لتحويل القيمة الرقمية الى قيمة رقمية عشرية كما نرى في المثال التالي

You can add parameters inside the curly brackets to specify how to convert the value.

Example

Format the price to be displayed as a number with two decimals.

في المثال قمنا بتنسيق السعر ليتم عرضه كقيمة عشرية، بعددين عشريين وذلك باستخدام الدالة format() والأقواس المعكوفة

```
price = 49
txt = "The price is {:.2f} dollars"
print(txt.format(price))
```

```
C:\Users\My Name>python demo_string_formatting2.py
The price is 49.00 dollars
```

نتيجة تشغيل الكود

القيم المتعددة

➢ Multiple Values

If you want to use more values, just add more values to the **format()** method.

إذا أردت استخدام قيم أكثر، فقط قم بإضافة هذه القيم إلى الدالة **format()** ويتم الفصل بينهم بواسطة الفاصلة ,

```
print(txt.format(price, itemno, count))
```

And add more placeholders.

هنا في المثال، قمنا بإضافة الأقواس المعكوفة في المكان الذي نريد أن نضع فيه القيم الرقمية

Example

```
quantity = 3
itemno = 567
price = 49
myorder = "I want {} pieces of item number {} for {:.2f} dollars."
print(myorder.format(quantity, itemno, price))
```

```
C:\Users\My Name>python demo_string_formatting3.py
I want 3 pieces of item number 567 for 49.00 dollars.
```

نتيجة تشغيل الكود

أتممت درسك بنجاح!

واصل التعلّم

طبّق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم السادس والستون

آلية تنسيق السلاسل النصية في لغة البايثون ٢

Python String Formatting 2

➢ Index Numbers

أرقام الفهرس

You can use **index** numbers (a number inside the curly brackets {0}) to be sure the values are placed in the correct placeholders.

تستطيع استخدام أرقام الفهرسة (وهي الأرقام التي بداخل الأقواس المعكوفة { 0})
للتأكد من أن القيم المناسبة وضعت داخل الأقواس الصحيحة.

Example

في المثال التالي نرى كيف أنه تمت الإشارة الى الكمية بالعدد 0 والنوع بالعدد 1 والسعر بالعدد 2 في السطر الرابع

```
quantity = 3
itemno = 567
price = 49
myorder = "I want {0} pieces of item number {1} for {2:.2f} dollars."
print(myorder.format(quantity, itemno, price))
```

يمكنك تمرير فهرس رقمي داخل القوسين المعكوفين والذي يبدأ من الفهرس 0

```
C:\Users\My Name>python demo_string_formatting4.py
I want 3 pieces of item number 567 for 49.00 dollars.
```

نتيجة تشغيل الكود

Also, if you want to refer to the same value more than once, use the index number.

Example

أيضا إذا أردنا الإشارة الى نفس القيم مرة أخرى نستطيع استخدام أرقام الفهرس

```
age = 36
name = "John"
txt = "His name is {1}. {1} is {0} years old."
print(txt.format(age, name))
```

كما نرى في المثال فإنه قد تمت الإشارة إلى نفس القيم مرة أخرى باستخدام أرقام الفهرس

```
C:\Users\My Name>python demo_string_formatting5.py
His name is John. John is 36 years old.
```

نتيجة تشغيل الكود

➢ Named Indexes

الفهرسة بالاسم

You can also use named indexes by entering a name inside the curly brackets **{carname}**, but then you must use names when you pass the parameter values

txt.format(carname = "Ford").

تستطيع أيضا استخدام الأسماء في الفهرسة عن طريق ادخال الاسم بين الأقواس المعكوفة **{carname}**

يجب استخدام الأسماء أثناء تمرير القيم للمُعاملات في الدالة **format()**

Example

```
myorder = "I have a {carname}, it is a {model}."
print(myorder.format(carname = "Ford", model = "Mustang"))
```

كما نرى في المثال، استخدام الأسماء في الفهرسة وبعد ذلك اعطاء قيم لهذه الأسماء.

```
C:\Users\My Name>python demo_string_formatting6.py
I have a Ford, it is a Mustang.
```

نتيجة تشغيل الكود

رائع!

أتممت درسك الأخير لهذا الأسبوع

روابط قد تهمك للاستفادة فقط

Useful links

- دوال السلاسل النصية | String Functions (1) | (format) تنسيق السلاس النصية(1)
- String Formating in Python - .format() Example
- Python 3.7: Format String Method
- A Python String Formatting Tutorial

طبّق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم السابع والستون

&

اليوم الثامن والستون

تحدي الأسبوع (يتم حله ورفعه على Github)

التحدي الأول

اكتب برنامج يطلب من المستخدم إدخال الحرف الأول والحرف الأخير من اسمه، ثم يقوم البرنامج بطباعة جملة تخبر المستخدم أن اسمه يبدأ بـ(الحرف الأول الذي قام المستخدم بإدخاله) وينتهي بـ (الحرف الأخير الذي قام المستخدم بإدخاله)

التحدي الثاني

اكتب كود لنص منسّق باستخدام دالة format() بنفس ترتيب الجملة التالية :

Dear Ahmad Ali, Your current balance is 53.44 $

تحدّي إضافي - لستَ مُلزمًا بحله -

اكتب برنامجا يطلب من المستخدم إدخال عدد عناصر المصفوفة، ثم يطلب البرنامج من المستخدم إدخال قيمة لكل عنصر في المصفوفة ثم يقوم البرنامج بعرض كل القيم التي قام بإدخالها المستخدم على سطر واحد.

موفق دومًا

انتظرنا في دروس الأسبوع القادم