بسم الله الرحمن الرحيم

إحترف بايثون الآن

تم تحميل هذا الكتاب من موقع كتب
www.kutub.info
للمزيد من الكتب في جميع مجالات التقنية ، تفضلوا بزيارتنا

خاضع لرخصة FDL
Copyright (c) 2003 under terms of FDL license

## إهداء

اهدى هذا الكتاب لكل مبرمج على طريق الإحتراف

## مقدمه

هذا الكتاب مقسم لحلقات ف الشرح وذلك لظنى أن هذا سيكون أفضل ف إستيعاب القارئ
هذا الكتاب خاضع لرخصةGNU FDL
أى نسخ أو تصوير أو إقتباس لايعد مخالفا للقانون ولكن إذا إستخدم ف عمل منتج مشتق
فإنه يخضع لرخصة FDI
كما لايمكنك الإدعاء بأنك من قمت بالقيام بهذا العمل ويجب عليك الإشارة لمؤلفه الأصلى
لاحظ أنا غير مسؤول عن اى خطأ مادى يقع عليك أو على جهازك وطبعا ولا عن أى خطأ معنوى
لاحظ واجهتنى بعض المشقه في الكتابه باللغه العربيه فأعذرنى إن كانت الألفاظ غير سليمه لغويا فلو أنه بالإنجليزيه لكان الأمر سهلا ولكن فليعيننا الله

## كلمة عـــــامة

بايثون من اللغات القلائل التى حققت المعادله الصعبه حيث أنها جمعت بين القوة والسهوله والمرونه مما يجعلها مناسبه لكل من المبتدأين والمحترفين
برمجه ممتعه مع بايثون

## كلمة عن الكـــــــــتاب

هذا الكتاب لايشترط أى خبره برمجيه سابقه وإن كانت موجوده فلاضرر

بعد قراءتك لهذا الكتاب  بحمد الله ستكون قد علمت عن بايثون مايسمح لك بوضع لقب مبرمج بايثون
هذا الكتاب جزء من سلسلة إحتراف للغات البرمجه وهى سلسله مجانيه تهدف لتثقيف الشباب العربى
ويتعامل هذا الكتاب تدريجيا مع مستوى القارئ بفرض أنه لايعلم شئ عن البرمجه
ملاحظة تمت الإستعانه ببعض الكتب والمقالات ف كتابه هذه الحلقات التى سوف نتناول فيها
إن شاء الله مقدمه صغيره عن لغة بايثون ولكن اولا

## ماهى لغة بايثون؟؟؟

هى لغه تفسيريه (اى تحتاج برنامجا مفسر للأكواد للتنفيذ)وذات واجهه تفاعليه وتدعم البرمجه الموجهه
واقوى مايميزها هو سهولتها وشعبيتها فهى أسهل من بيرل (لغه من أسهل اللغات) وتكاد تكون تعمل على
هذه اللغة ستتغير حياتك.وتستطيع  جميع أنظمة التشغيل ويكمن ان تفعل بها ماتريده بمجرد ان تتذوق طعم
دمجها C/C++في الأكواد في لغة
برنامج على قناة CNNسميت بايثون بهذا الإسم على إسم
“Monty python's Flying Circus”

سهولة بايثون أى أنك عندما تقرأ كودا بهذه اللغه تكون كاءك تقرأ إنجليزى وقواعدها سهله للغايه وحره
ممايتيح لك التعديل فيها كما تريد ولغة عالية المستوى تبعدك عن تعقيدات إدارة الذاكره وغيره

## من الأنظمة التى تعمل عليها هذه اللغه :

Linux/FreeBsd/Windows/macintosh/Solaris/VMS/(OS/2)/Amiga/BeOS/Palm
OS/QNIX/Psion/Playstation/VxWorks/Sharp Zaurus/
Acorn Risc Os/Windows CE,/Pocket PC
وأوضح انها تعد اسهل اللغات وهى تعد الخيار الأول للمبتدئين ولكنها لغه قويه ليست لعبه

كاتب هذه اللغه: Guido van Rossum هولندى الجنسيه

**ومكتبتها القياسيه** :اكثر من رائعه فتفيدك ف الحسابات المعقده والتطبيقات **GUI**
وكل شئ والتعامل مع الملفات وغيرها لاحظ انها تعد مثل بيرل ف القوه من ناحية الويب لاحظ انك يمكنك
التعديل ف كودها المصدرى (سورس اللغه (المكتوب بلغة سى ) ) كما تريد ويناسب ذوقك
من اهم المواقع التى تفيدك هى

www.python .org
www.techbooksforfree.com

هناك عشرات الالاف من المستخدمين حول العالم يمكنهم مساعدتك غير طبعا مؤسس اللغه
**استمع لرأى ذلك الكاتب** Eric S.Raymond
وهو يعد من أعظم الكتاب والمبرمجين
" ان بايثون اصبحت هى لغته المفضله "
Bruce Eckel يقول
وهو كاتبThinking inC++ و Thinking in Java
" يقول ان بايثون ربما هى اللغه الوحيده التى تجعل العمل أسهل من اجل المبرمج والعديد "

## كتـــب مهمة لمبرمجى بايثون

انصح بقراءه
1-Non –programmer Tutorial for python for josh cogliati
2-byte of python for swaroop
3-the official documentation of python
4-Dive in python
5-Thinking in python

دعم بايثون على جهازك

## بايثون على ويندوز

هناك نوعان من بايثون على ويندوز
ActivePython>>>free one and recommended
تستطيع تنزله من هنا
www.activestate.com/Products/ActivePython/
لو بتستخدم98/me
هتحتاج تنزلwindows Installer 2.0
official python>>>free but for developers u may use it

## بايثون على مـــــاك

أمامك خيارين صارت بايثون تأتى مع النظام Mac OS X
وهى تنصيب بايثون أولا أرجح أنك ستنصبها وهذا جيد
على Mac OS X10.2
ستجد أن هناك نسخه منصبه على الجهاز ولكن تتعامل مع سطر الأوامر فإن كنت مرتاحا معه فلا مشكله

لك سوى فـXML parser بالنسبه
حيث أنه ليس من ضمنها فبالتالى ستحتاج لتنصيب الإصدار كاملا
لتشغيله ستفعل الخطوات الآتيه
/Applications
/Utilities
D.click on Terminal
ثم إكتب python
للخروجCtrl ^D
للحصول على إصدار2.3
press

أفضل أن تستخدم نظاما مفتوح المصدر كأنظمة جنو أنا افضل لينكس
لاحظ إذا لم تكن تستخدمIDLE
إستخدم احد محررات النصوص مثل Kwrite,Gedit
هتلونلك النص وكده وذلك ف حال إن لم تستخدمvi or emacs
لاحظ انا اكرهnotepad ,word pad
لأنهم محررات ضعيفه
لاحظ Cool edit رائع
ولكن يمكنك إستخدامهم القاعده العامه لتشغيل الأكواد ان تحفظها بإمتدادpy
وتعطى التصاريحchmod +xw filename.py
وبعد ذلك تستدعى المترجم وتكتب إسم الملف
python filename.py
لاتعطى تصاريح ف ويندوز لاحظ إنك هتنزل المترجم طبعا فى ويندوز اولا هيا بنا

```
#!usr/bin/python
print "Hello world \n"
print 'I am talking from the wonderful python \n'
```

**الأول حددنا مسار مفسر الكود وهذا المسار خاص بأنظمة يونكس وتغيره حسب مسار المفسر ولو ف ويندوز ولا أى مشكله تغير المسار للمكان الموجوج فيه المفسر**
**تانى سطر هو تعليمة**print=printf=system.out.println
**بالسى والجافا وبراحتك بعديها " او ' بس لازم تقفلها**
**وهى دالة إخراج ممكن تستخدم مكانها**
**دالة الإخراج الأساسيه**
**بس لاحظ هتضطر تضيف مكتبيه ع العموم سنتعرض لذلك بالتفصيل قريبا إن شاء الله**

```
sys.stdout.write()
```

```
#!usr/bin/python
import sys
sys.stdout.write(“Hello World! \n”)  #\n=newline
```

**\n بتديك سطر جديد طبعا عارفها لو جربت سى او بيرل وغيرها**
**إيه رأيك بجد سهله صح؟**

**التعليقات بتستخدم # وتكتب التعليق تختلف عن سى /* */**
**لاحظ أول سطر ف برنامجك تعليق خاص بتحدد فيه مسار المفسر اللى ستقوم بإستخدامه ال shell**

```
#!usr/bin/python
#This a comment
#This is another comment and go on
print "This program is just comments nothing else \n"
```

**المتغيرات والثوابت يمكن لك تعريفها عادى وليس مثل لغة سى**

```
#!usr/bin/python
a=5
b=2
print "a+b = ",a+b              #adding
print "a x b = ",a*b            #multiply
print "a / b= ",a / b           #dividing
print "a^b= ",a**b
```

**ملحوظة لمبرمجى C/C++**

**لايوجد مايسمى ب char في بايثون وأظن أنك لن تحتاجه**

**الحلقات والشرط**

```
 #!usr/bin/python
x=[1,2,3,4,5]
for I in x :
    print I
```

**بإستخدام for ده حلقه تكراريه صغيره هتطبع 1 2 3 4 5 رأسيا**

```
#!usr/bin/python
x=1
while x<100 :
      x=x+1
     print x
```

**الناتج هيطبعلك الأرقام من 1 ل 100 على صورة رئسيه**

```
#!usr/bin/python
x=1
while x<100 :
      x=x+1
     print x,
```

**الناتج هيطبعلك الأرقام من 1 ل 100 على صورة افقيه**

## مثال صغير برنامج باسوورد وقاعدهIf

```
#!usr/bin/python
x=raw_input("what's ur name : ")
if x=="ahmed":
  print 'hello ahmed'
elif x=="l1nUx3r":
   print 'welcome l1nUx3r'
else: print 'u r not allowed to access'
```

**لاحظ يمكنك إستخدامinput=raw_input**
**إذا كان المدخل عددى إستخدمint(input("number:"))**

**بس إيه رأيك بجد؟لغه سهله لاحظ حاول ان تبتعد عن سى او سى ++ ف البرمجه طالما لاتحتاج للتحكم ف الجهاز بتلك الدرجه فاستخدم لغه تفسيريه تهدر موارد النظام ك بايثون وبيرل**

## شروط للمتغيرات

**لاتكون كلمه محجوزه ف اللغه مثل class**
**يجب ان يكون الحرف الأول حرف ابجدى سواء كبير أو صغير أو يبدأب underscores(_)**
**يمكن أن يحتوى على أرقام**
**تختلف المتغيرات بإختلاف الإسم حيثvar1#!Var1**
**لايحتوى على مسافات**

## نسق الكتابة

**هناك عدة أساليب ف الكتابه مثل إنك تكتب أول حرف كبير**
**مثال**
myname
Myname
myName

## الكلمـــــــــات المحجوزه Reserved words

هى كلمات طبعا لايجوز لك إستخدامها بتسمية المتغيرات

```
#     and       elif      global      or          #
#     assert    else      if          pass        #
#     break     except    import      print       #
#     class     exec      in          raise       #
#     continue  finally   is          return      #
#     def   for     lambda      try               #

 del        from        not         while     #
```

## ملخص للحلقه

تعرفنا ف هذه الحلقه على لغة بايثون وكاتبها وآراء العديدين حولها
تعرفنا على اللغات التفسيريه وعلى التعليقات وكيفية تعريف المتغيرات والثوابت وشرطيةif

الكلمات المحجوزه
ووظيفة الإخراجprint,sys.stdout.write

يجب أن تعطى التصاريح حتى يعلم النظام أن هذا النص قابل للتنفيذ

الحلقه الثانيه سنتناول فيها إن شاء الله بعض أهم الدوال ف بايثون وطريقه تعريفها

## مقدمه عن الدوال

معنى الدوال : هى عبارة عن جزء من الكود تمت كتابته مره واحده ويتم إستخدامه كثيرا جدا فتم وضع ذلك الكود جاهزا بإسم داله

ملحوظه افضل إستخدامIDLE اثناء التجربه حتى تكون الإجابه لحظيه

يمكنك إستخدام محرر نصوص كما قلنا ولكنك ستضيف السطر التالى
```
#!usr/bin/python
```

### دالة len

بتعطيك عدد ما وبالمثال عدد حروف الكلمه

```
>>>a='word'
>>>len(a)
4
```

هناك إستخدامات متقدمه سنتحدث عنها قريبا

### دالة Range

```
>>>range(10)
[0,1,2,3,4,5,6,7,8,9]
>>>range(0,10,3)
[0,3,6,9]
>>>a=['ahmed','went','there']
        for x in range(len(a)):
            print x ,a[x]
0 ahmed
1 went
2 there
```

هذه الداله بتكون مصفوفه من عدد العناصر التى تحددها مثل المثال الأول

3 وفيها الأعداد تتزايد بمقدار10 إلى 0وف المثال الثانى تكونت مصفوفه من
وف المثال الثالث إستخدمنا جمله دواره for
وذلك لترتيب المصفوفه كل عنصر مع ترتيبه ف المصفوفه

## داله pass

من غير ضحك الداله ده مش بتعمل حاجه خالص سنتناولها ف تعريف الدوال

## دالةchr

هذه الداله تقوم بتحويل القيمه المدخله من جدول الآسكى إلى الرموز الأبجديه والأرقام

```
>>>chr(65)
'A'
>>>chr(97)
'a'
```

## دالةord

هذه الداله عكس سابقتها تقوم بتحويل المدخلات إلى القيم المناظره من جدول الآسكى

```
>>>ord('a')
97
>>>ord('A')
65
```

## دالةhelp

تقوم بها بالإستعلام عن أى شئ تريده مثلا
هذه المخرجات من على جهازى

```
>>>help('help')

Welcome to Python 2.3!  This is the online help utility.

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at http://www.python.org/doc/tut/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To quit this help utility and
```

return to the interpreter, just type "quit".

To get a list of available modules, keywords, or topics, type "modules", "keywords", or "topics". Each module also comes with a one-line summary of what it does; to list the modules whose summaries contain a given word such as "spam", type "modules spam".

دالة dir
دالة تستخدم ف عرض محتويات المكتبيه اللتى تريدها لاحظ المثال

```
>>>import sys,fibo
>>>dir(sys)
['__displayhook__', '__doc__', '__excepthook__', '__name__', '__stderr__',
 '__stdin__', '__stdout__', '_getframe', 'api_version', 'argv',
'builtin_module_names', 'byteorder', 'callstats', 'copyright',
'displayhook', 'exc_clear', 'exc_info', 'exc_type', 'excepthook',
'exec_prefix', 'executable', 'exit', 'getdefaultencoding', 'getdlopenflags',
'getrecursionlimit', 'getrefcount', 'hexversion', 'maxint', 'maxunicode',
'meta_path', 'modules', 'path', 'path_hooks', 'path_importer_cache',
'platform', 'prefix', 'ps1', 'ps2', 'setcheckinterval', 'setdlopenflags',
'setprofile', 'setrecursionlimit', 'settrace', 'stderr', 'stdin', 'stdout',
'version', 'version_info', 'warnoptions']
```

إستخدمنا import
حتى نستطيع إلحاق مكتبةsys,fibo

سنتعرض لهما إنشاء الله ف شرح بعض المكتبات
دور dir
قامت بعرض كل الدوال التى تحتويه المكتبيه

حسنا لقد تعرضنا لبعض الدوال لاحظ انه توجد ف المكتبيات دوال مهمه للغايه سنتعرض لبعضها حين نشرح بعض هذه المكتبات

## تعريف الدوال

لاحظ الأمثله التاليه

```
>>>def hi():                  #defining function called hi
        “””this function to print hi”””          #show what does it do
        print 'hi'          #this what the function do just print hi
>>>hi()                      #we used the function
hi
```

ماهى def؟

هى كلمه مفتاحيه نخبر بها بايثون بأننا نريد تعريف داله

ماهى:()hi؟

هى الداله التى نريد إنشاءها ويجب ان تعقب إسمها ب ()

ومتنساش النقطتين:

بعد كده وضحنا ماتقوم به الداله """ """ ذلك سيظهر على شكل تعليق ف حال إستخدام IDLE

وبعد كده اضفنا خصائص الداله وقد إخترنا ان نطبع الداله كلمه Hi

وبعد كده إستخدمنا الداله مباشره ()hi

وإطبعت كلمه hi

لاحظ المثال التالى على داله متقدمه شويه صغيرين

```
>>> def fib(n):    # write Fibonacci series up to n
...     """Print a Fibonacci series up to n."""
...     a, b = 0, 1
...     while b < n:
...         print b,
...         a, b = b, a+b
...
>>> # Now call the function we just defined:
... fib(2000)
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
```

بنجمع فيها الرقمين والناتج نجمعه ع التالت وكده

نشرح الكود بالتفصيل

عرفنا داله وداخلها متغير fib(n)1 والثانى =0وحددنا وظيفتها “””وبعد كده عرفنا جواها متغيرين الأول = n

جمله while الشرطيه إستخدمناها لتحديد الشرط هو انا طالما المتغير الثانى اكبر من متغير الداله

يطبع العدد السطر قبل الأخيرb ويأخذ العدد a قيمه العددb الذى اصبح يساوى مجموع العددين السابقين

1000لإإستخدمنا الداه وجعلنا قيمه المتغير داخلها =
وطبع الجهاز الناتج كما تراه

مثال على دالة pass

```
>>>def hi():
        pass
>>>hi()
>>>
```

قمنا بعمل داله لاتفعل شئ

بعض الردود عن أسئله حول الحلقه السابقه
لماذا يجب ان اتعلم البايثون؟
اخى لأن هذه لغه سهله كما اوضحت ف الحلقه السابقه ولأنك لن تتعب نفسك ف تتبع الأخطاء بتلك الطريقه كالتى مع C/C++

مالها لغات C#، C/C++ ,Java,Perl,Pascal,Basic

لن اتحدث عن لغة C/C++,Java,Pascal
فهى لغات عظيمه ولكن هناك شئ واحد انا هذه اللغات ان بحبها بجد ولكن من ناحية تتبع الخطأ تكون العمليه مرهقه ومتعبه للغايه وإسئلوا عن ذلك مبرمجى سى وجافا وباسكال

C#
لغه جميله ولكن من عيوبها انها لن تعمل على بيئة تشغيل غير ويندوز إلا ببرنامج مونو على لينكس
وبعض الأنظمة الأخرى كمـــــــاك وهو ليس بكفاءه دوت نت ولكن جارى التطوير

Basic
هذه اللغه بدأت فيها تعلم البرمجه وبحمد الله لم اكن افهم شئ ومالأخر انا باكره هذه اللغه اللتى تذهب العقل وأضف على ذلك انها لغه ركيكه حاولوا جعلها سهلها ولكنهم فشلوا بل محوها تماما وأرجو ألا يسألنى احد حيث انا بها عيوب تملأ كتابا ولكن ليس لى حق النقد إذا اردت ان تستخدمها انت حر ولكنها تعلم عادات برمجيه سيئه وايضا بالمقارنه بسى ليس للبيزيك اى لازمه معذره أخوانى مبرمجى بيزيك
وقد تكون فيجوال بيزيك دوت نت حلوه لكنى لن اتنازل عن بايثون
صحيح اخى إن كنت تسأل عن الفيجوال بيزيك كنت ارجو ان تسألنى على دلفى بيئة التطور للغه العريقه باسكاال
لن اقوم بالرد عن اسئله مقارنة بين لغات برمجيه
Pick ur programming language as u like

Perl
هى لغه جميله وتتميز ف السلاسل النصيه وإداره الأنظمه والويب وإن لم تعجبك بايثون إستخدمها ولكن كود بيرل اقرب للسى ولكنه سهل ويشرف بايثون ان تكون من اللغات القلائل التى تنافس بيرل
بإذن الله سيلى كتاب خواطر حول بايثون كتاب حول بيرل

اصحاب باقى الأسئله قد أرسلت لكم الردود على المنتدى معذرة لم اجد المجهود لإضافة باقى الأسئله

www.python.org
www.GNU.org
www.freetechbooks.com
www.linuxjournal.com
www.linux4arab.com

ملخص

تكلمنا ف هذه الحلقه عن بعض أهم الدوال الجاهزه
تناولنا تعريف الدوال
تحدثنا عن بعض لغات البرمجه

**المصفوفات ف بايثون**

المصفوفه ( هى عباره عن مجموعه متغيرات ) يعنى ممكن يكون العنصر الأول منها سلسله حرفيه وممكن يكون رقم اى شئ

اولا المصفوفات ف بايثون تتعامل مع اول عنصر فيها على انه رقم صفر والتالى مثله رقم واحد
ومن الناحيه الأخرى ف المصفوفه يبدأ العد بالسالب فاخر عنصر يساوى -1
لاحظ المصفوفه هى مجموعه من العناصر ولا يشترط ان تكون ارقام
قم بمراجعة الحلقه السابقه ف دالهrange

```
>>>a=[1,2,3,4,5,6,7,8,9]
>>>a
[1,2,3,4,5,6,7,8,9]
>>>a[0]
1
>>>a[2]
3
>>>a[-1]
9
```

لاحظ إذا وضعت ف المصفوفه كلمات تضعها بين علامتى تنصيص's'

ملحوظه لمبرمجىperl /php
مفيش فرق ف بايثون بين علامات التنصيص سواء احاديه أو ثنائيه او ثلاثيه
''' ''' ” “ ' '

```
>>>b=['python','c','c++','java','lisp']
>>>b[0]
python
>>>b[2]
c++
```

مثال ب إستخدام مصفوفه for

```
>>>a=[1,2,3,4,5,6]
```

```
>>> for x in a: print x
1
2
3
4
5
6

>>>a=[1,2,3,4,5,6]
>>> for x in a: print x,   # comma is used to avoid printing a newline
1 2 3 4 5 6
```

عندما نريد إضافة عناصر جديده للمصفوفه ماذا نفعل؟؟

الإجابه ببساطه نكمل على المثال السابق a.append(element) فيضاف العنصر الذى نريده وللترتيب نستخدم a.sort()

```
>>>a.append[0]
>>>a
[1,2,3,4,5,6,0]
>>>a.sort()
>>>a
[0,1,2,3,4,5,6]
```

بعض العمليات ع المصفوفات

```
>>>a=[1]
>>>a
1
>>>a.append(0)          #add 0 to a
>>>a.append(2)          #add 2 to a
>>>a.sort()             #sort a
>>>a                    #ask what is a now?
```

```
[0,1,2]
 >>>a[0:2] = [3, 12]   #repalce the first and the second elements with
                                                              0   and 1
>>>a
[3, 12, 2]
>>>a[:]=[]              #remove all elements in a
>>>a
[]
>>>len(a)               #counting  how many elements in a
0
>>>s=[0,1,2,3,4,5,6]
>>>s.append(9)
>>>s.pop()              #last in first out
9
>>>s
[0,1,2,3,4,5,6]
```

لاحظ هذا إستخدام اخر من إستخدمات دالهlen وذلك ف عد عدد عناصر المصفوفه

أرجو ان اكون اعطيتكم فكره ضئيله عن المصفوفات

هذا جدول لبعض اهم الوظائف

| | |
|---|---|
| *list*.[*a*:*b*]=list2 | تكوين مصفوفه جديده من عنصرين |
| *list*.append(*n*) | إضافة للمصفوفه |
| list.sort() | للترتيب المصفوفه |
| *list*.insert(n,*list*) | ةفاإضاقة مصفوفه داخل مصفوفه |
| *list*.index(*X*) | X =رقم العنصر الذى تريده |
| List.pop() | اخر عدد أضيف واول عدد يخرج من المصفوفه |
| del *list* | تمحو المصفوفه |
| *list*.pop(*n*) | nتعطى الرقم حسب الترتيب للمتغير الداخلى ف الداله |
| *list*.remove(*X*) | تحدد قيمه العدد الذى تريد محوه من المصفوفه |
| List.reverse | تعطيك المصفوفه بترتيب مخالف من اليمين للشمال |
| | ممكن لك ان تضيف عده وظائف اخرى فلنعد ذلك واجبا |

مع العلم ان المصفوفات المركبه والقواميس لن اتحدث عنها الآن
طيب ماذا الآن  سأقول لكم سنتحدث إنشاء الله عن السلاسل النصيه الحلقه القادمه ومكتبه sys
حسنا وحتى ذلك الحين إن كنت قد إستفدت اى شئ أرجو منك كمجرد محاوله كتابه كود لبرنامج بسيط باسوورد
كلمات سر ويقولك مرحبا لو باسوورد تانى يقولك إنت مش مسموح لك3بحيث يقبل البرنامج والإصدار الثانى برنامج باسوورد يطلب الإسم  و الباسوورد
البرنامج الثانى هو برنامج مقارنه بين عددين يخبرك اى منهما الأكبر
البرنامج الثالث برنامج يعطيك المضاعف المشترك الأكبر لعددين
اى مساعده ف الأكواد لاتتردد
وبعد الدرس القادم  إنشاء الله ستشعر بجد بالتقدم مع بايثون

بعض الردود عن  الأسئله

لماذا لاأستخدمC/C++
إستمع إلى
لأنى أجبت عن هذا السؤال من قبل البايثون لغه قويه للغايه وقد لمحت ف احد الردود انها توازى بيرل ف القوه ولكن اكثر مايميزها هو انها مناسبه جدا للمبتدأ وليس تعلمها ف صعوبة تعلم السى طب هقولك على حاجه ف لغه إسمها كوبول ده ممكن تكتب فيها برنامج 200 سطر ولكن سى تقوم بنفس الوظيفه ف 30 سطر والبايثون تقوم به  ف 20سطر كفرض طب ثوانى انا هغير قاعده وسأظهر لك بعض الأكواد من عدة لغات

C language
```
#include<stdio.h>

void main()
{
   char* msg="Hello World!";

   puts(msg);
}
```

C++ language
```
#include &ltstring.h>
#include &ltstdio.h>

class Message  // define a message class
{
private:
   char msg[50];
```

```
public:
   Message(char* s)
      {strcpy(msg,s);};
   void print()
      {puts(msg);};
};

void main()
{
   m = new Message("Hello World!");
   m->print();
   delete m;
}
```

## Perl language

```
#!usr/bin/perl
$msg = "Hello World";
print $msg;
```

## python language

```
#!usr/bin/python
print 'hello world'


#!usr/bin/python
class Hello{
   def hi(self):
       print 'hello world'

x=Hello()
x.f(self)
```

## Java language

```
class HelloWorld{                       //define Hello World class
static void main (String args[]){
System.out.println(“Hello world”)       //print Hello world
}
}
```

إيه رأيك اخى لقد جعلتنى اخالف قاعده هامه ولكن لايهمك الأن قرر مارأيك إيهم أسهل من هذه اللغات وتعلمه

والترتيب مره اخرى باللغات التى إذا اردت ان تتعلم البرمجه فهى تعد من أساسيات المبرمج وثقافته الأساسيه

Python,Perl,Java,C/C++,lisp

أرجو مره اخرى منكم اى حد عنده فكره عن الليسب او البرلوج يراسلنى للأهميه

لاحظ اخى ان من قوة البايثون تعد بها شركة ردهات التطبيقات بتاعتها وشركه ماندريفا تستخدم بيرل

لاحظ إنى بعد كل ذلك سأطلب منكم تعلم السى ولو أنك لن تكتب بها اى كود بل لمجرد التعلم من هذه اللغه العريقه

www.GNU.org
www.python.org
www.perl.org

ملخص
ف هذه الحلقه قمنا التعرض للعديد من خصائص المصفوفات

## السلاسل النصيه وبعض المكتبات

نكمل معكم الحلقه الرابعه إن شاء الله فيها سنتحدث بصوره بسيطه عن السلاسل النصيه تعريفها هى مجرد حروف مدموجه معا ( كلمة ) او مفرده تحاط بعلامات تنصيص احاديه او ثنائيه او ثلاثيه
' ' "" ''' '''

مثال عليهم

'this is a string'
"this is a string also"
'''This is anthor one'''

ياترى مجمعين؟؟ يعنى اى كلام داخل علامات تنصيص هو سلسله نصيه

## التعامل مع المتغيرات

```
>>>i=5
>>>print i
5
>>>i=i+1
>>>print i
6
>>>s="hello how are u?"
>>>print s
hello how are u
>>>print i;                #for C/C++ and perl programmers
6
```

ف المثال السابق عرفنا ثابتا هو I
وأعطيناه القيمه 5
على قيمته الأصليه1وكانت هى إضافة I وبعد ذلم وضعنا قاعد ه لمتغير يسمى
hello how are u? أيضا اعلنا عن سلسله نصيه
ألسيت بايثون سهله ف التعامل مع السلاسل النصيه والمتغيرات؟؟

ملاحظه لبمرمجى بيرل وسى وجافا مجرد عاده عند سى وبيرل وجافا إستخدام الفاصله المنقوطه ; او لأ إنت وراحتك ولكنها مجرد عاده عند سى وبيرل حتى لاتغير قواعدك الثابته عندك

**ندخل على مكتبية sys حالا فهى مكتبه جميله للغايه**

ناخد مثال؟؟

ارجو ان تستخدم IDLE ولاضرر ف محرر النصوص

```
>>>import sys
>>>n=input( "enter 0 to quit and 1 to advance")
>>>if n==0:
            sys.exit()
>>>elif n==1: print "hiiii"
```

لقد إستخدمنا وظيفه sys.exit ()الموجوده ف sys وهى نستغلها ف الخروج من البرنامج

**ملحوظه مبرمجى سى وجافا**

**لست مضطرا ف بايثون لإضافة الشرط بين اقواس**

```
>>>sys.platform
```

هيديك ناتج مختلف من شخص لأخر حسب نظام تشغيله بس هو هيديك نوع النظام
ممكن طريقه اخرى

```
>>>from sys import *   #it equals to import sys
>>>exit()               #we used the function directly with out sys
```

الطريقه ده عشان مش تقعد تكتب إسم المكتبه وبعد كده الوظيفه لكن تختار الوظيفه على طول لكن افضل الأولى لكن لك الإختيار فهنا لا إجبار

```
>>>sys.copyright
```

هيديك رقم الإصدار اللى معاك

```
>>>sys.getwindowsversion()
```

ده لمستخدمى ويندوز فقط هيديك الإصدار بتاعك إيه

```
>>>sys.version()
```

intel32 هيديك نوع الجهاز ف الغالب

كده تمام ؟؟

هناك وظائف اخرى لم اذكرها من اراد كل الوظائف ينظر إلى وثائق بايثون الرسميه ف موقعهم نصيحه إذا اردت الدعم هناك فى روماات البرمجه وانظمة التشغيل خاصه بتاعت الياااهو ده طبعا بعد ماتحاول لأن بجد سيرد عليك ردا لن تتقبله لأن هناك فلسفه تعنى عدم المحاوله ف شئ لم يحاول فيه مريده ولكن إن لم تعرف فليس هنااك مشكله لكن المحاوله اولا والدعم ثانيا

عوده للسلاسل النصيه

```
>>>name='ahmed'          #string
>>>if name.startwith('ahm'):
        print 'yes,the string starts with "ahm"
>>>if a in name:
            print 'yes, it contains an a
>>>L= '_**_'
>>>list=['python','is','extremly','powerfull']
>>>print L.join(list)
python_**_is_**_extremly_**_powerful
```

هناك بعض الوظائف الأخرى ستضاف بمشيئه الله ف الحلقات القادمه

لاحظ اخى إن لم تعمل معك اى من الأمثله انظر اولا ف التصاريح وإجعلها 775 وبعد ذلك راسلنى إن لم تعمل

برمجه التطبيقات هى مرحله متقدمه ولكن تعتمد بطريقه كبيره على البرمجه الكائنيه

برنامج صغير القاعد ه اخى التى تريدها كتبتهالك ف الأسفل *

برنامج يحول درجات الحراره عن طريق الدوال
ومره اخرى عن الطريقه المعتاده وقارن بينهما

(Celsius=(f-32)*(5/9

ملخص
لقد تعرفنا على مكتبيةsys
وبعض وظائف السلاسل النصيه,

## الأخطاء الشائعه ومكتبيه string

ف هذه الحلقه سنتحدث بمشيئة الله عن بعض ف بعض الأكواد وسنتناول فيها إن

شاء الله مكتبية string

```
>>>x=o
>>>while x<0  print x
```

لن يعمل ذلك الكود لأنك قد نسيت :

```
>>>while x<0: print x
>>>x=0
>>>if x==0 print  'wrong'
```

لن يعمل أيضا وذلك لأنك قد نسيت :

```
>>>if x==0: print 'wrong'

>>>a=[0,1,2,3]
>>>for x in a print a
```

لن يعمل ايضا وذلك لأنك قد نسيت:

```
>>>for x in a: print a


>>>Print 'hello'
```

لن يعمل ذلك لأن لغة بايثون حساسه جدا للحروف

```
>>>print 'hello'

>>>5*(2/0)
```

لن يعمل ذلك الكود لأنه لايمكن القسمه على 0

```
>>>'1'+1
```

لن يعمل ذلك الكود لأنه لايمكن جمع سلسله نصيه مع عدد
إلا ف حال إستخدام جمله()eval

```
>>>eval('3+4')
7
>>>eval('3'+'4')
34

>>>3+p*5
```

لن يعمل ذلك الكود وذلك لأنك لم تعرف المتغير او الثابتpفليست له قيمه تدخل ف العمليه الحسابيه

```
>>>def hi()
```

لن تعمل هذه الداله مع اننا قمنا بتعريفها ف حلقه سابقه وذلك لأننا نسينا :

```
>>>def hi():
```

هذه بعض الأخطاء الشائعه ولكن هناك جملهtry_except
سنقوم بشرحها هى وملحقاتها إنشاء الله ف حلقات قريبه قادمه
عوده السلاسل النصيه
ملخص لبعض الوظائف لهذه المكتبه

| | |
|---|---|
| ***String.upper(string)*** | ***تحول الحروف الصغيره لكبيره*** |
| ***String.lower(string)*** | تحول الحروف الكبيره لصغيره |
| ***String.capitalize(string)*** | تجعل اول حرف كبير |
| ***String.swapcase(string)*** | تجعل الحروف الكبيره صغيره والصغيره كبيره |
| | هناااك العديد من الوظائف مطلوب منكم إكمال الجدول |

```
>>>import string
>>>string.upper(ahmed)
AHMED
>>>string.lower(AHMED')
ahmed
>>>string.swapcase('AhMeD')
aHmEd
```

هذه كانت بعض الأمثله على مكتبيةstring
مع أنها امثله سهله ولكنها مفيده لاحظ انا لم اتحدث إلا عن اهم الوظائف لأن البحث واجب عليك فلن تعرض هذه الحلقات كل شئ عن بايثون ولاحظ انها مجرد مقدمه وحينما تتأقلم مع بايثون سوف تستطيع ان تجد المساعده والدعم كما قلنا

## ملخص

تعرفنا على بعض الأخطاء الشائعه
تعرفنا على بعض وظائف مكتبيةstring

# القواميس والمصفوفات ذات التركيب المرتب

هذه الحلقه السادسه ف بايثون
سنتناول فيها إن شاء الله
اولا ماهى القواميس؟؟
هى نوع من المصفوفات ولكن له مفتاح سنرى حالا
افضل التعامل معIDLE
حتى تكون النواتج لحظيه ولك الحريه ف غير ذلك بإستخدامك لمحررى نصوص

```
>>>dict1={“name”:”ahmed”,”age”:”83”}
>>>dict1
{“name”:”ahmed”,”age”:”83”}
>>>dict[“name”]
ahmed
>>>dict[“age”]
83
```

شرح الكود

مع ان الكود واضح إلا اننا سنشرحه
الأول عرفنا قاموس{}
وضعنا داخله عناصر كل منها له مفتاحا
المفتاح هو اللى ع الشمال بحيث إنك لما تحب تعرف إيه اللى بيشير إليه المفتاح على طول بتكتب الكلمتين
إسم القاموس وبداخله المفتاح
جرب كده تكتب العكس إسم القاموس ثم الشئ الدال عليه المفتاح ستظهرلك رساله خطأ
تقدر تضيف زى مانت عايز مفاتيح وقيم وتغيرهم كمان

```
>>>dict[“language”]=”python”
>>>dict
{“name”:”ahmed”,”age”:”83”,”language”:”python”}
>>>dict[“age”]=”30”
>>>dict
{“name”:”ahmed”,”age”:”30”language”:”python”}
```

الجدول التالى به بعض الخصائص المتعلقه بالقواميس

| ***Function*** | ***العمل*** |
|---|---|
| Dict.items() | يديك كل العناصر على صوره ازواج مرتبه من المفتاح والقيمه |
| Dict.clear() | محو كل العناصر |
| Dict.keys() | يديك كل المفاتيح |
| Dict.has_key(name) | تتحقق منطقيا من وجود المفتاح ام لا |
| Len(Dict) | يديك عدد عناصر القاموس |

انا بيتهيألى إن كده تمام القواميس صح؟؟
يارب تكون تمام

## tuple

سنسميها توبل بالعربى عشان إسمها كبير مصفوفه ذات تركيب مرتب
اولا ماهى التوبل؟؟؟؟؟
هى مصفوفه بس ثابته مش ليها اى خصائص غير التأكد موجود ولا لأ
ليه بنستخدمها بقى ؟؟؟
عشان هى اسرع م المصفوفات ولكن عادى ولايهمك هنا ليك الحريه البرنامج ممكن تكتبه ب10 طرق ع العموم إنت وراحتك
المهم يلاا نشوف التوبل ده مالها

```
>>>a=(1,2,”a”,”b”)
>>>a[1]
2
```

```
>>>a[1:2]
(2,"a")
>>> 1 in a
True
```

جرب كده اخى بعض خصائص المصفوفات على التوبل ولاحظ ماذا سيحدث
سوف نطلق إنشاء الله حلقه للغة سى ستجدونها بالمنتدى قريبا حتى يكون عندك فكره عما تقوم به هذه اللغه
ولن اطلب منك شيئا سوى ان تستطيع قراءه الكود البرمجى فقط المكتوب ولك مطلق الحريه ف إستخدام اى لغه تشاء

## ملخص

تحدثنا ف هذه الحلقه عن القواميس المصفوفات المرتبه

## lambda

هذه هى الحلقه السابعه ف بايثون وسنتناول فيها إنشاء الله عدة أشياء مثل

### lambda

بعد الطلب الكبير على عدة مميزات ف بعض لغات البرمجه وايضا ف ليسب تمت إضافة العديد منها لبايثون مثل هذه الكلمه
اظن انك لن تحتاجها ع العموم سنشرحها
لن اتناول عليها سوى مثالين حتى يتضح لك دورها والباقى لإبداعاتكم

```
>>>def inc(n):
    return lambda x: x+n

>>>f=inc(1)
>>>f(2)
3
```

شرح الكود
اولا عرفنا داله وبداخلها متغير ما
داخل متن الداله وضعنا القاعده وهى ان يتم تعريف متغير x
حيث قيمته هى مجموع متغير الداله الذى اصبح ثابتا والمتغير الجديد الذى نحدده
شرح الكود مره اخرى
لاحظ اننا قد عرفنا داله ووضعنا فيها متغير n
بعد ذلك وضعنا قواعد هذه الداله وهى ان تعيد مجموع المتغير السابق الذى يمكن ان نعده اصبح

ثابتا ف الداله و المتغيرx
مثال اخر

```
>>>def inc(n,l):
        return lambda x=x+n+l
>>>f=inc(4,3):
>>>f(5)
12
```

عرفنا داله اخرى ووضعنا بداخلها متغيرين
وجعلنا قاعدة الداله ان تجمع المتغيرين وأيضا العددx
لاحظ يتم ف ثالث سطر وضع قيم المتغيرين
وف الرابع يتم وضع قيمة المتغيرX
ننظر الآن لجزئيه ربما تكون مهمه للبعض وهى معرفة وثيقة الداله حيث حينما تعرضنا لتعريف الدوال
نعطى مثال للتذكير هيا بنا

```
>>>def function():
      “””doesn't do anything ok
       really it doesn't”””
     pass
>>>print function.__doc__
  “””doesn't do anything ok
       really it doesn't”””
```

معنى ذلك هو ان كل مايكتب بين"“ ”"""
هو وثيقه مبسطه للداله يوجد بها ماذا تفعل هذه الداله وكيفيه إستخدامها

مثاال واقعي

```
>>>import sys
>>>print sys.exit.__doc__
exit([status])

Exit the interpreter by raising SystemExit(status).
If the status is omitted or None, it defaults to zero (i.e., success).
If the status is numeric, it will be used as the system exit status.
```

If it is another kind of object, it will be printed and the system exit status will be one (i.e., failure).

هذا المثال على داله(sys.exit
فيعطيك كل ماتفعله الداله

## عودة للمصفوفات والدوال الجاهزه

هذه هى الحلقه الثامنه ف بايثون

سنعود ف هذه الحلقه إلى المصفوفات وسنتعامل إن شاء الله مع بعض الدوال الجاهزه الخاصه بها
وهذه الدوال هى
map(),reduce(),filter(),zip()

### اولا(filter

تستخدم لإعطاء مصفوفه مستثناه من عناصر تحدد انت طريقه الإستثناء عن طريق داله تقوم بإنشاءها
مثاال على هذه الداله

```
>>>def f(x): return x % 2 !=0 and x %3 !=0
>>>filter(f, range(2,25))
[5, 7, 11, 13, 17, 19, 23]
```

مالذى حدث؟؟؟
3 ولا على 2اولا قمنا بتعريف داله تعيد القيم التى لاتقبل القسمه على
إستخدمنا(filter
25 إلى 2بأننا وضعنا قاعده الداله السابقه وحددنا عناصر الصفوفه من
داخل مصفوفه3 ولا على 2 والتى لاتقبل القسمه على 25 إلى 2قامت الداله بإعاده كل القيم من
داله مفيده ف رأيى

```
>>>def f(x): return x % 5 !=0 and x % 6 !=0
>>>filter(f,range(2,25))
[2, 3, 4, 7, 8, 9, 11, 13, 14, 16, 17, 19, 21, 22, 23]
```

ولا 5ذلك مثال اخر حيث إستثنينا م المصفوفه المطلوبه كل الأعداد التى لاتقبل القسمه على
6على

## ثانيا()map

تعيد لك مصفوفه على هيئة متتابعه بمجرد ان تعرف داله وتضع القاعده
كلام كاللوغاريتمات صح؟؟
فلنرى بعض الأمثلة

```
>>>def square(x): return x**2 #u may use x*x
>>>map(square,range(1,11)
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

كوننا مصفوفه من مربعات الأعداد المحصوره بين 1 و 11
حيث حددنا ف الداله التى عرفناها اننا نريد مربعات هذه الأعداد
ف المثال التالى حددنا ف الداله المعرفه اننا نريد مكعبات هذه الأعداد

```
>>>def cube(x): return x**3 #u may use x*x*x
>>>map(cube,range(1,11))
[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]
```

مثاال اخر

```
>>>seq=range(9)
>>>def square(x):
        return x*x
>>>map(None,seq,map(square,seq)
[(0, 0), (1, 1), (2, 4), (3, 9), (4, 16), (5, 25), (6, 36), (7, 49),(8,64)]
```

ومربعاتها على هيئة أزواج مرتبه مكونه من العدد 11 إلى 1هذا المثال اعاد لك الأعداد من ومربعه
بمعنى اخر اننا حولنا زوج من المصفوفات إلى مصفوفه مكونه من أزواج
حيث المصفوفه الأولى range(9)
والثانيه هى مصفوفة الأعداد المربعه وأستخدمنا كلمهNone
للوقايه مثل النون ف اللغه العربيه ف بعض الأحيان

## ثالثا()reduce

هذه الدله سهله للغايه ف فكره عملها
حيث انك تعرف داله اولا ثم تستدعى تلك الداله من خلال دالةreduce
لنوضح ذلك بأمثله

```
>>>def add(x,y): return x+y
>>>reduce(add,range(1,11))
55
```

مالذى حدث ؟؟؟
الذى حدث اننا عرفنا داله تقوم بجمع عدد م المتغيرات
ثم إستدعينا دالتناreduce
ومررنا لها اننا سنستخدم الداله الأولى كقاعده وان المتغيرات التى ستسخدمها الداله الأولى هى
11 إلى 1الأعداد من
وكان حاصل جمعهم 55

رابعا zip()
مثال

```
#!usr/bin/python
#this is on zip(y,x) that matches y and x

questions=['name','age','favorite color']
answers=['ahmed','17','red']
 for q,a in zip(questions,answers):
        print 'What is your %s? it is %s' %(q,a)
```

ماهذا ؟
هذه الدالة فعلا خطيره كل اللى عملناه هنا هو إن إحنا حددنا مصفوفتين وقامت هذه الدالة بتشكيل خليط منهم لاحظ الناتج

www.python.org
www.c4arab.com
www.perl.org
www.perl.com

ملخص

عوده للدوال الجاهزه مثل

map(),reduce(),filter(),zip()
وتمت مناقشتهم تفصيليا

## لغة سى

هذه الحلقه التاسعه ف بايثون
وسنتحدث فيها عن لغه سى إجمالا
ستساعدك ف وضع قدمك على الطريق ف لغة سى لاحظ هناك اجزاء سأتجنبها
بالكامل  مثل المؤشرات وقد قام الأخ جيمى بشرحها على اكمل وجه
ومن يريد الأستفاضه ف لغة سى يمكنه مراسلتى بهذا الخصوص وسأدله على العديد مما يستطيع
قراءته والله ولى التوفيق
اولا ماهى لغة سى؟
هى لغه قياسيه وصممت ف الأساس لكتابة أنظمة التشغيل والبرامج الكبيره فتعطى كفاءه عاليه
وتحكم كبير للمبرمج ف الجهاز فممكن للبرامج المكتوبه بسى بأن تكون ف سرعه البرامج
المكتوبه
بالأسمبلى وذلك ف حال إستخدامك لكومبايلر جنو  gcc
هناك ايضا مترجمين جيدين مثل الذى من شركه بورلاند
لكن انصحك بألا تستخدمها إلا ف الحاجه فقط وغير ذلك فمن يبحث الأن عن البيتس والأن
المساحات بالجيجا والذاكره صارت كبيره فإذا اردت إستخدم لغه تهدر لك موارد النظام مثل
بايثون او بيرل
وإستخدم سى حينما تريد ذلك الكم الكبير من التحكم
لاحظ سنستخدم كومبايلر جنو ف الشرح وانت حر مع اى مترجم اخر لكنى افضله فهو مجانى
وهو الأفضل
اولا يجب ان تختار محرر نصوص يلونلك السكريبت اللى بتكتبه مثل
vi,emacs,Gedit,kwrite,pico,nano
تكتب السكريبت وتحفظه ف مكان ما بإمتدادc
filename.c
وتعطى التصاريح للقراءه والتنفيذ عن طريق

```
chmod +775 filename.c
chmod +xw filename.c
```

ولتشغيل البرنامج إكتب الأمر التالى

```
gcc filename.c -o filename
```

سنتحدث ف هذه الحلقه عن برامج بسيطه للغايه

```
#include<stdio.h>
#include<stdlib.h>
main()
{
puts (“hello world talking from c \n”);}
return 0;
}
```

اولا تعليمهinclude
وتسبق ب#
تعنى إضافة المكتبيه التى تليها فأضفنا مكتبتىstdio.h,stdlib.h
ف السطر الثانى قمنا بوضع الداله الرئيسيهmain
واتبعناها ببلوك البدايه{
ف السطر الثالث إستخدمنا تعليمة puts
ويمكن إستخدامprintf
أضفنا الفاصله المنقوطه لاحظ انا سى ليست ف سهولة بايثون
ولكن يجب ان تتعلم ولو ان تقرأالكود الذى امامك فقط
بعد ذلك اعطينا الأمر بالتنفيذ يمكن الاتطلب منك بعض الكومبيلرات إضافةreturn
على غير القاعده العامه وبعد ذلك اغلقنا البلوك1 والفشل =0التنفيذ={
لاحظ الفاصله لازمه بعد كل امر
تعريف المتغيرات ف المثال التالى

```
#include<stdio.h>
#include<stdlib.h>
main(){
int a;
puts(“Enter a number:”);
scanf(“%d”,&a);
printf(“The number is %d”,a);
return 0;
}
```

ماذا فعلنا ؟
مثل المثال الأول اضفنا المكتبيات
وعرفنا متغيرا سميناهa
اظهرنا رساله تطلب منك إدخال رقم
وبعد ذلك إستخدمنا تعليمهscanf
وهى تساوى ف بايثون input
ووضعنا فيها انها ستقبل عددا صحيحا وتم إعطاء ذلك العدد كقيمه للمتغيرa
وبعد ذلك اظهرنا رساله مفادها ان الرقم الذى تم إدخاله هو
الذى سيظهر

| | |
|---|---|
| ***%d (decimal)*** | تأخذ عددا صحيحا فقط |
| ***%f(float)*** | تاخذ عددا حقيقيا |
| ***%c(char)*** | تأخذ حرفا |
| ***%s(string)*** | سلسله نصيه |
| ***%e(input numbers in scientific notation)*** | 10تأخذ رقما مضروبا ف أس |

الدوال ف سى

```
#include<stido.h>
#include<stdlib.h>
message(){
puts (“hi what's up”)
}
main(){
message()
return  0;
}
```

المثال واضح للغايه
حيث قمنا بتعريف داله ما ووضعنا داخل متنها الخصائص المطلوبه وهى ف ذلك المثال ان تطبع كلاما ما
بعد ذلك إستدعينا الداله من خلال الداله الرئيسيه main

سنتحدث عن جمل الشرط والدوارات إجمالا بعرض امثله والشرح وفقنا الله

## if,else

```
#include<stdio.h>
main(){
int a;
printf(“please enter number  greater than 5”);
scanf(“%d”,&a);
if (a < 5);
printf(“the number u entered is less than five”);
else
printf(“the number u entered is greater than five”);
```

عرفنا متغير ما داخل متن الداله الرئيسيه واظهرنا رساله تطالب بإدخال ذلك الرقم وإستخدمنا دالة الإدخال scanf
ومررنا القيمه الداخله كقيمه للمتغيرa
تظهر رساله بذلك وإن لم يكن تظهر رساله تخبرنا 5وضعنا الشرط اما إذا كان الرقم اقل من بغير ذلك يمكنك وضع عدد كبير م الشروط أليست جميله هذه اللغه؟؟

## for

هذه الجمله قد إستخدمناها كثيرا ف بايثون لو لاحظت ع العموم فلنبدأ بمثال

```
#include<stdio.h>
main(){
int a;
for (a=1; a <=5;a++)
printf("Hello world from The C programming language");
return 0;
{
```

المثال واضح ف رأيى ولكن ماذا فعلنا؟؟؟
عرفنا داخل متن الداله متغيرا ما
وإستخدمنا جمله دوارهfor
ووضعنا بداخلها ان قيمه1= a
وان a مرات للشرط وهو طباعه جملة ما5 ولذلك فيجب التكرار 5اقل من
حتى تصل إلى قيمه الخمسه وذلك بزياده مقدارها واحدا

هكذا تعرفنا على لغة سى إجمالا بحيث ان حينما اتحدث عن وظيفه اساسيه ف بايثون وأقارنها بسى تستطيع التفاعل مع ذلك الكلام ونرجو من الله التوفيق
لاحظ تلك الحلقه ستعاد كتابتها بإذن الله عن طريق متخصص ف لغه سى ليس مثلى وسيكون أسلوب الشرح افضل من ذلك وفقنا ووفقكم الله

## Del & break

هذه هى الحلقه العاشره ف بايثون

بعد عوده من الكلام عن لغة سى ف الحلقه السابقه ولكن كان لابد منها حتى يكون عندك
فكره عنها
كلمة break نستخدمها لإيقاف تنفيذ البرنامج ف حال تحقق شرط ما

```
>>>for x in range(0,10):
                print x
            if x==5:
            break
0
1
2
```

```
3
4
5
```

ذلك مثال إستخدمنا فيه جملتى شرط إذا لاحظتfor, if
10 إلى 0قمنا بعمل دواره صغيره لعد الأرقام من
ثم وضعنا شرطا داخل جملةfor
وذلك بإستخدام شرطيةif
وهى ان يتوقف البرنامج حينماx=5

## جملة del

تستخدم على صورتها العاديه وهى للحذف سواء كان مصفوفه او متغير او ثابتا و صنف
او غيرهم

```
>>>i=5
>>>i
5
>>>del i
>>>i
```

السطر الأخير هيديك رسالة خطأ

```
>>>n=[1,2,3,4]
>>>n
[1,2,3,4]
>>>del n
>>>n
```

هيديك رسالة خطأ برده

## ملخص

تعرفنا على

break,del

## البرمجه الموجهه بالكائنات

الحلقه الحادية عشر ف بايثون

بعد العديد من الحلقات شاء الله ان يتم تقديم حلقات البرمجه الكائنيه وهى ستكون ف حدود حلقتين إلى ثلاثه

لقد أصبحت بحمد الله على درايه كبيره بالعديد من خصائص هذه اللغه وأنت معنا على مر هذه الحلقات ويمكنك ان تقوم ببرامج فعاالله بدون البرمجه الكائنيه
سطر ألن يحتاج 100ولكن هل لاحظت انك حينما تكتب برنامجا أكبر من للتنظيم والدقه وان تتبع الخطأ ناهيك ان هذه بايثون وليست سى فإن تتبع الخطأ أسهل
اولا ماهى البرمجه الكائنيه؟؟
هى تقسيم البرنامج لصنوف وتضع الخصائص داخل كل صنف ومن ثم تستدعيه ف البرنامج فيكون التعامل مع البرنامج اسهل ومن ناحية التعديل أيضا ف أى جزئيه فتستطيع تطويره بسهوله وتطويع البرنامج لصالحك مثلا كلمات10حينما تعدل كلمه واحده اليس أسهل من ان تعدل
لاحظ إن بايثون تنافس أقوى اللغات من ناحية البرمجه الكائنيه فهى تعد م منافسى جافا الذين يستحقوا التقدير

جـــافا

هى لغة برمجه كائنيه صافيه حتى أنك ف أسهل الأكواد يجب أن تستخدم الكائنات وكودها أسرع ف تصنيفه من البايثون وهناك مترجم ف لينكس يصنفها للغة الآله مباشرة مما يجعلها مقاربه للسى

عامة الجافا هى ثانى أفضل خيار ف تعلم البرمجه بعد البايثون فستكون بالنسبه لك

تجربه ممتعه

لاحظ الفيجوال البيزيك ليست لغه برمجه كائنيه حتى مع الذى يقوله الغير فينقصها العديد من خصائص البرمجه الكائنيه ممكن نقول عنها لغة برمجه ثلث كائنيه

ع العموم لن نتحدث عنها

الآن نبدأ سنرى أولا بعض الأكواد

```
>>>class Hello:
        i=12
        def hi(self):
              print “Hi there”
>>>x=Hello()
>>>x.i
12
>>>x.f()
Hi there
>>>Hello().i
12
>>>Hello().f()
Hi there
```

اولا عرفنا الصنف الذى نريده وسميناهhello

ووضعنا فيه متغيرا صنفيا أى لايمكن إستدعاؤه إلا من خلال الصنف الذى

عملناه
i=12
وبعد ذلك عرفنا داله وجعلنا بداخلهاself
لاحظself=This in java
لاحظself
ليس له أى تأثير ولكن كل داله تعرف داخل صنف يجب ان تحتوى عليه
الداله وظيفتها ان تطبع كلمه هاااى
لاحظ إستخدمنا حرف عادىx
وجعلناه يساوى الصنف ويمكن لك أن تتركه عادى مش مشكله
وبعد كده لو حبيت تستدعى اى شئ موجود داخل الصنف عادى بتكتب إسم
الصنف والشئ الذى تريده ولكن يجب أن يكون معرفا داخله

هذه هى الحلقه الثانية عشره ف بايثون
وثانى حلقات البرمجه الكائنيه
لقد توقفنا ف الحلقه السابقه عند صنفنا إذا تذكرتموه
والآن تعديل بسيط عليه

```
>>>class Hello:
            i=12
           def __init__(self,name):
           self.name=name
           def hi(self):
          print “Hi”,self.name
>>>x=Hello('ahmed')
>>>x.Hi()
Hi ahmed
```

def __init__(self,name)

الداله ده بإختصار شديد هى دالة بناء أى ان ماسيوضع ف الصنف بتاعنا هو إسم وقلنا ان self

تتجاهله
لاحظ ف إتنين ___ قبل وبعد
بعد كده أعطينا قيمه للأسم بتاعنا عشان نقدر نستخدمه ف الدوال الأخرى داخل الصنف
بمعنى اخر self هو حلقة الوصل بين الشئ الحقيقى والصنف والدوال التى داخله
ممكن تغيره براحتك self.name=name
بعد كده الداله بتاعتنا واخبرنا فيها ان سنستخدم متغيراselfله إمتداد ما اللى هو الإسم
بعد كده ساوينا بين صنفنا وداخله إسم وحرف ما
x=Hello('ahmed')
x هى حرف ما
Hello('ahmed')هو الصنف بداخله الإسم الذى اخبرنا الصنف عنه
بعد كده إستخدمنا الداله الداخليه اللى داخل الصنف
Hello('ahmed').hi()
وممكن
x.hi()
أمثله بدون شرح

```
>>>class message:
        def __init__(self,string):
           self.string=string
        def printIt(self):
         print self.string
>>>a=message('hi')
>>>a.printIt()
hi
>>>b=message('ahmed')
>>>b.printIt()
ahmed
>>>n=['a','b']
>>>for x in n:
          print x.printIt()

hi
ahmed
```

طبعا سنشرحه هههههههههههههههههههههههه

عرفنا صنفclass message
عرفنا دالة البناء بأننا سنضع داخل الصنف أى بين القوسين سلسله حرفيه
عشان نقدر نستخدمه ف باقى الموجودات ف الصنف كالدوالself.string=string
عرفنا داله عادى خالص وخلصنا الصنف
وعرفنا متغيرين وساويناهم بالصنف وداخله سلسله نصيه

```
a=message('hi')
b=message('ahmed')
```

وإستخدمنا الداله المعرفه داخل الصنف
لاحظ بنستخدم الداله كأنها إمتداد للصنف يعنى بنكتب الصنف الأول ونملاه باللى حددناه كالسلسله النصيه

```
a.printIt()
```

على فكره الجزئيه الأخيره

```
n=['a','b']
```

كانت مثال لمجرد الكسل البرمجى عادى لو مش عاجباك الطريقه بس عملنا مصفوفه من المتغيرين السابقين
وبإستخدام for
قمنا بإستدعاء الداله الداخليه ف الصنفprintIt
عشان نطبع الرسالتين ف وقت واحد

لنرى مثال اخر

```
>>>class C:
      def __init__(self,val):
         self.val=val
      def printIt(self): print “Hi my value is “,self.val

>>>a=C(21)
>>>b=C(34)
>>>a.printIt()
Hi my val is 21
>>>b.printIt
Hi my val is 34
```

Simple class

```
#!/usr/bin/python
# this is SimpleClass.py
# OOP and classes
class Simple:
# the self arg is just like this (you can't delete it)
  def __init__(self, str):
```

```
    print "Inside the Simple constructor"
    self.s = str
  # Two methods:
  def show(self):
    print self.s
  def showMsg(self, msg):
    print msg + ':',
    self.show() # Calling another method
# if we are runed as a prog (else then this file is used as  module)
if __name__ == "__main__":
  # Create an object:
  x = Simple("constructor argument")
  x.show()
  x.showMsg("A message" )
```

Area

```
#!usr/bin/python

class Square:
    def __init__(self, side):
        self.side = side
    def calculateArea(self):
        return self.side**2

class Circle:
    def __init__(self, radius):
        self.radius = radius
    def calculateArea(self):
        import math
        return math.pi*(self.radius**2)

list = [Circle(5),Circle(7),Square(9),Circle(3),Square(12)]

for shape in list:
    print "The area is: ", shape.calculateArea()
```

هذه الحلقه الثالثه عشره ف البرمجه بإستخدام بايثون
سنستكمل ق هذه الحلقه بمشيئه الله الصنوف والبرمجه الكائنيه
لاحظ هذا الكتاب ليس إلا مجرد دوره سريعه للغايه ف بايثون

هذه الحلقه تعد من الحلقات النهائيه ف ذلك الكتاب
لكن أرجو ان يقرأ هذه الحلقه من لديه خبره عن البرمجه بإستخدام الجافا وإن لم تكن فلا ضرر
عاااامة لن نخرج عن موضوع هذا الكتاب ألاوهو البرمجه بإستخدام بايثون ولن أظهر اكواد بلغة جافا لأنك البرمجه ولكن الحديث سيكون عن This لايمكن ان تحدد هذه اللغه خلال سطورا فهى تعد من أعظم لغات شيئ مميز لها وهو معامل
فإذا كنت من مبرمجى جافا فلاتغير قواعدك الراسخه فى ذهنك فإذا اردت
self=this
مثاال بإستخدام this

```
>>>class D:
       def __init__(this,realpart,imagpart)
                this.real=realpart
                this.imag=imagpart
      def printIt(this):
        print “The realpart equals:”,this.real
        print “The imagpart equals:”,this.imag
>>>x=input(“the realpart :”)
>>>y=input(“the imagpart: “)
>>>n=D(x,y)
>>>n.printIt()
```

أظن أنك مذهول بالفعل هذا ماتراه عيناك
لك الحق فى أن تكون فلك حريه الإختيار فى إستخدام بعض قواعد الجافا مثل مثالنا يمكن لك ان تقوم بتغيير thisإلىself
الآن لتدرك هل قمت بإدراك مفهوم البرمجه الكاائنيه أرجو منك تعديل المثال السابق ليصبح بإمكانك ان تأتى بحاصل الضرب والقسمه والمجموع والفرق لأعداد مركبه
لاحظ أى إجراء يتم فصله الحقيقى مع الحقيقى والتخيلى مع التخيلى ماعدا الضرب والقسمه

# التعامل مع الملفات File handling

الحلقه الرابعة عشره ف بايثون

سنتناول فيها إن شاء الله التعامل مع الملفات

## نبذة عن التعامل مع الملفات

المقصود هنا هو فتح الملف وإلاغلاقه وقراءته

لنبدأ بمثال بسيط ف محرر النصوص المفضل لديك إكتب فيه
this is a simple text about python
it's a really great programming language
whatever you do there's python for you

وإحفظ هذا النص ف أى مكان بأى إسم وليكنtext.txt
لو أنت بتستخدم ويندوز إحفظه ف مكان المفسر غير كده ف أى مكان
بس إحفظ المسار
هذا البرنامج من أدواتك الخاصه فإن شئت طورها فهى مثل
cat >>>In UNIX,  type>>>In Windows
إفتح IDLE
أو اى محرر نصوص
وإكتب الآتى

```
#!usr/bin/python
x=file(“/home/ahmed/text/txt”,”r”)
for line in x.readlines():
```

```
print line
x.close()
```

## شرح الكود

إستخدمنا دالة وهى ومخصوصه لفتح الملفات file()
هذه الدالة بتاخذ مقطعين الأول مسار البرنامج والثانى ماستفعله سواء قراءه أو كتابه
هنااك دالة أخرى لفتح الملفات()open

```
#!usr/bin/python
x=open(“/home/ahmed/text/txt”,”r”)
for line in x.readlines():
print line
x.close()
```

لاحظ معنى”read"r
يعنى ذلك أن الملف المفتوح سيكون للقراءه فقط

ثم إستخدمنا دوارةfor لعرض السطور من خلال دالة القراءه readlines()
وإخبرنا بايثون أن تطبع لنا السطور الموجود ف الملف
ثم ستخدمنا دالة الإغلاق()colse

## strip()

تستخدم هذه الدالة ف مسح المسافات بين الجمل يعنى لو سايب فاصل سطرين تقوم الدلة ده
تخليها سطر هتسأل طب وإيه الميزه؟؟
إنت هتلاحظها بنفسك

```
#!usr/bin/python
x=open(“/home/ahmed/text.txt”,”r”)
for line in x.readlines():
print line.strip()
x.close()
```

## كيف تكتب داخل ملف؟

سنستخدم دالة بطريقة بدائيه وسنستغلها ف الكتابة داخل ملف من ملف أخر كأنها دالة نسخ
إعمل ملف آخر فارغ وسميهtext2.txt

```
#!usr/bin/python
x=open(“/home/ahmed/text.txt”,”r”)
y=open(“/home/ahmed/text.txt”,”w”)
for line in x: y.write(line)

print “file is copied”
x.close()
y.close()
```

مثال سهل فعلا
لقد أحضرنا المخرجات من ملف القراءه ووضعناها منسوخه ف ملفنا الجديد
يالها من لغة

## تلميحات

تستطيع تطوير هذه الأدوات البرمجيه بل وأن تحفظها ف مجلد البرامج عندى انا مثلا/usr/bin
ثم تستدعيها كآداة رئيسيه
تستطيع تعديل هذه الآدوات بأن تجعل المدخلات هى مدخلات سطر الأوامر
commandline _arguments
عن طريق مكتبيةsys
من خلال وظيفة argv
يجب أن تفتح الملفات قبل إستخدامها
ويجب أن تغلقها بعد الإستخدام
لاتستطيع إستخدام خيار القراءه والكتابه معا
\n تعليق بسيط على متاعقبة السطر الجديد
هناك أنظمة يختلف إستخدام هذه المتعاقبه من نظام لآخر
'\r\n'>>>ويندوز
'\n'>>>يونكس وشبيهاته وأنظمة جنو خصوصا لينكس
'r'>>>ماك
وحاليا'\n'
إرجع للوثيقه الرسميه بهذا الخصوص ونظام تشغيلك
AIX,Irix,Solaris,Unix,Windows

## معاملات سطر الأوامر

هذه الجزئيه مع سهولتها الشديده ولكنى لن أقوم بشرحها لأنى أظن أنك ستستوعب الكود

## echo

```
import sys
for x in sys.argv:
print x,     # the comma is for not making a new line
```

الحلقة الخامسة عشر ف بايثون

سنتناول فيها بمشيئة الله كيف تسطيع إنشاء مكتبيه
ولكن أولا ماهى المكتبيه؟؟
المكتبيه ف أبسط كلمات هى مجموعه من الدوال الجاهزه والمتغيرات والثوابت التى قمت بكتابتها وتريد إستخدامها دون
كتابتها مرات عديده أو تنسخها من برامج أخرى ولكن تتسخدم لها أمر الأستدعاء import
نعم مثل مكتبيةsys
ولكن كيف ننشئ هذه المكتبيه؟؟
أليس يجب أن تكون عبقريا للقيام بكتابة مكتبيه؟؟
الرد سهل ليست معظم المكتبيات كتبها صاحب اللغه ولكن كتبها بعض المطورين أيضا
سننشئ ف هذا الفصل مكتبيه بها عدة دوال جاهزه لنستطيع التحويل الحروف من صغيره لكبيره
وأيضا بعض الوظائف المشابهه لوظائفC/C++

لتعرف أين يمكنك حفظ هذه المكتبه ثم إستدعائها إكتب الآتى

```
>>>import sys
>>>sys.path
['/home/ahmed', '/usr/bin', '/usr/lib/python23.zip', '/usr/lib/python2.3', '/usr/lib/python2.3/plat-linux2', '/usr/lib/python2.3/lib-tk', '/usr/lib/python2.3/lib-dynload', '/usr/lib/python2.3/site-packages', '/usr/lib/python2.3/site-packages/Numeric', '/usr/lib/python2.3/site-packages/PIL', ……..]
```

شكل المكتبيه العام

```
####module Da.py####
pi=3.14
j=-1**.5
def hi(): print “Hi”
##################
```

على سبيل المثال بعد حفظ المكتبيه ف home

إكتب الآتى

```
>>>import Da
>>>Da.Pi
3.14

>>>Da.hi():
```

“Hi”

وهكذا
أرأيت يمكنك بعد أن تستخدم دوالك أن تنشئ مكتبيه مكونه منها
مثال على مكتبيه حقيقيه

```
##written by : l1nUx3r

def  to_upper(string):
    ##it converts a string to upper case
upper_case=” “
for character in string:
  if 'a' <= character <='z':
location=ord(character)-ord('a')
new_Ascii=location+ord('A')
character=chr(new_ascii)
upper_case=upper_case+character
print upper_case
##..
def printf(string): print string
```

مكتبية جيده أليس كذلك بها وظيفة الإدخـــــــال الرئيسية ف سى
ودالة لتحويل الحروف الصغيرة لكبيرة

## بعض السكريبتات الخفيفه

سكريبت لحساب قيمة محدد

```
#!usr/bin/python
#This program is under FDL lisnece

x="""
|a b c|
|d e f|
|g h i|
"""
print x

r1=['a','b','c']
r2=['d','e','f']
r3=['g','h','i']
print "The rows are in lists []"
r1=input("1st row in [] :")
r2=input("2nd row :")
```

```
r3=input("3rd row :")
a=r1[0]
b=r1[1]
c=r1[2]
d=r2[0]
e=r2[1]
f=r2[2]
g=r3[0]
h=r3[1]
i=r3[2]

print "The value",(((a*e*i)+(b*f*g)+(c*d*h))-((b*d*i)+(a*f*h)+(c*e*g)))
```

تطبيق ع الكائنات والمتغيرات الداخليه (مهم)

```
#!usr/bin/python



def options():
        print "1-circle \n""2-square \n""3-recetangle \n""4-info \n""5-quit\n"
```

```
class circle:
	def __init__(self,radius):
		self.radius=radius
	def area(self):
		import math
		return math.pi*(self.radius**2)

class square:
	def __init__(self,side):
		self.side=side
	def area(self):
		return self.side**2

class recetangle:
	def __init__(self,height,width):
		self.height=height
		self.width=width
	def area(self):
		return (self.height)*(self.width)


options()


x=input("Number :")
if x==1:
	r=input("The radius: ")
	n=circle(r)
	print "The area: ",n.area()
elif x==2:
	r=input("The side : ")
	n=square(r)
	print "The area: ",n.area()
elif x==3:
	h=input("The height: ")
	w=input("The width: ")
```

```
        n=recetangle(h,w)
        print "The area",n.area()
elif x==4:
        print "it's made by l1nUx3r "
elif x==5 :
        import sys
        sys.exit()
```

lwc ( l1nUx3r word Counter)

عداد كلمات

```
        !#usr/bin/python

        import sys
        import string

        def numwords(s:(
        list=string.split(s(

        return len(list(



        n=sys.argv[0]   #we won't use it cause it returns the ac program
        x=sys.argv[1[
        f=open(x,"r("
        total=0

        for line in f:
        total=total+numwords(line(

        print x," has %d words" % total
```

## Cal viewer

```
#!usr/bin/python

import calendar
print "choose the year"
year=input("the year??:")
calendar.prcal(year)
```

## Cls

```
#!usr/bin/python

print "\n"*1000
```

## Cout ( C++ printing function)

```
#!usr/bin/python

def cout(string): print string

cout("using c++ function cout")
```

## Factorial دالة المضروب

```
#!usr/bin/python

n=input("the number?:")
def fac(n):
    if n<=1:
    print n,"! = 1"
   else: print n,"!=",n*fac(n-1)    #it should be "!= ", not = "!"=n*fac(n-1)
```

## fibonnacci sequence

```
#!usr/bin/python
```

```
#fibonnacci sequence

a,b=0,1
count =0
max_count = 20
while count < max_count:
count=count+1
old_a=a
old_b=b
a=old_b
b=old_a+old_b
print old_a
```

## Odd or EveN

```
#!usr/bin/python
#even
num=input("The number is: ")
if num % 2 ==0:
        print num,"is even"
elif num % 2==1:
        print num,"is odd"
else: print num,"is very strange"
```

## Rate

```
#!usr/bin/python

print "we are calcualting the time "
rate=input("The rate : ")
distance=input("The distance is : ")

print "The time is ",distance/rate
```

## unstoppable loop

```
#!usr/bin/python
#unstoppable loop
```

```
while 1==1:
    print "hi i'm a crazy loop"
```

## Size

```
#!bin/python

from os.path import*

file1=input("put the path of the file : ")
print "The size equals",getsize('file1')
```

## NewLine

```
#!usr/bin/python

def newline():
        print


print "hi"

newline()

print "This was an empty line"
```

## Password

```
#!usr/bin/python

password="l1nUx3r "
while password !="DaRkoOo":
    password=raw_input("Password: ")
print "welcome in"
```

## Abs

## القيمة المطلقه

```
#!usr/bin/python
#abs        means the absolute value
n=input("number? :")
if n < 0:
        print "the abs of ",n,"is",-n
else:
        print "the abs of ",n,"is",n
```

## تمارين عـــــــــــامة

قم بتطوير هذه المكتبية لتحتوى على لوظائف الإدخال الرئيسية فPascal, C/C++
طور الأدواتecho,cat
فتستطيع إستخدام معاملات سطر الأوامر ف مخرجاتهم
طور دوال وأضفها لمكتبيتك الخاصة للتعامل مع السلاسل النصية
إن كنت تهوى الجوانب الرياضيه قم بإضافة بعض الثوابت العلمية والعلاقات الرياضيه

## ماذا الآن

بعد أن أنهيت قراءة الكتـــــــــاب كل ماعليك هو أن تفتح محرر النصوص المفضل لديك وأن تكتب الكود التالى

```
#!usr/bin/python

print “I am a really cool python hacker”
```

## تعليقات بسيطه

حينما تكتب برنامجا الأفضل ان تضع تعليقا بسيطا يوضح كيفية عمله وألا تستخدم طرق صعبة ف فهم قارئ الكود

الآن يجب أن تحدد ماذا ستفعل؟؟

هناك عدة أشياء تستطيع فعلها مثل أن تتعلم عدة لغات برمجة أخرى وأرشح لك جـــافا أو بيرل كخطوة تالية
أو أن تتعمق أكثر ف بايثون فتستطيع إقتحام المجالات الرسومية
هنـــــــــاك عدة أنواع من بايثون إذا كنت مازلت معجبا بهذه اللغة أظن أنك ستبحث عنها

أخى لاتحاول أن تمنع النفع عن إخوانك
فمن أكثر الناس عذابا كاتم العلم وإنما أهلكنا وفنينا بكتماننا العلم عن إخواننا
فالعلم ليس حكرا على أحد فلا تصغى لهراء حقوق الملكية الفكرية وذلك الكلام العقيم
فلاأقصد بذلك المحاولة لإختراق القانون
فإن كانت هناك برامج تحتاج أن تدفع لها المئات من الدولارات حتى يكون لك حق الإستخدام إعترض ولا تستخدمها لأنه يوجد بدل هذه البرامج المئات والمئات من البرامج الحره والمجانيه التى هى أفضل من هذه البرامج الغالـــــــــية التى ستدفع أموالا طائله لإستخدامها وليس حتى تطويرها

تم بحمد الله كتـــــــــــاب إحترف بايثون

الكاتب
l1nUx3r

كلمة من الكاتب_ لاحظ أنك لست مطالب بقراءه هذه الجزئيه فهى ليست ضمن مقرر الكتـــــــــاب _
أشكر الله الذى وفقنى ف كتابة هذا الكتاب وتقديمه ف مثل هذا الشكل

أود أن أشكر العديدين ممن ساعدونى وألهمونى ف كتابة هذا الكتـاب وأن أتقدم لهم بالشكر بجانب والدى ووالدتى
الكتاب إهداء خاص ل

لطقم من عرباوى

LaMoR,ABOHELAL,Dr_LeeDo,Rock,Zanger9210,Egyptian_Lady,Storm_3arabawy,Dr.SaDa,Game MaSter

من SeCurity GuRuS

**Storm,ACiDWareZ,rOCkMaStEr,Ne0,Safa7,Untrust,DarkLinux,MySQL,HackoBacko**

هذا الكتـــــــــــــــاب مكتوب على Open Office
وتمت كتابتـــــــــــــه وتحويله لكتاب على منصة لينكس
وليكون الكلام دقيق
***GNU/Linux Mandrake 10.1,GNU/linux Mandrake9.2,GNU/linux Redhat 9***

***نرجو أن يكـــــون هذا الكتاب قد نال رضاكم وإعجابكم***
***نسألكم الدعوة الصالحة***
***برمجـــــــــــــــــــة ممتعة لكم مع بايثون***