Preview from Notesale.co.uk
Page 5 of 341

مقدمة

لماذا هذا الكتاب ؟

السبب الرئيسى هو أنه كتاب **مجانى** ومتكامل ولااعتقد انك بعد قرائته ستواجه صعوبات مع Python

ماذا تتوقع بعد قراءتك لهذا الكتاب؟

بعد قرائتك للكتاب ستكون قد علمت الكثير حول بايثون وكيفية توظيف اللغة بصورة جيدة وجدية، ستصبح متأقلم مع تطبيقات قواعد البيانات والشبكات -ربما تفكر فى كتاب خادم ويب خاص بيك؟- والتطبيقات الرسومية والكثير والكثير من طرق معالجة البيانات وغيرها الكثير.. ستكتشف هذا بنفسك!

المتطلبات

- جهاز عليه احد نظم GNU/Linux او Windows (يفضل XP or better)
- معرفة جيدة بكيفية إستخدام Text Editor رفة باشياء مثل
- معرفة Permissions, Groups, Users .. فى حال عدم المعرفة يفضل الإتجاه إلى LinuxAC ستجد العديد من المواضيع لتساعدك

Preview from Notesale.co.uk
Page 8 of 341

الإصدارات القادمة والتحديثات

- ستجد اخر الإصدارات والتحديثات على Programming-Fr3aks.NET

المساهمين

- Ahmed Youssef

حقوق النسخ

مسموح بالنسخ والتوزيع وإحتمالية التعديل تحت بنود **GNU FDL** مع وجود الأقسام الثابتة "حقوق النسخ ، الإصدارات القادمة ،المساهمين)

الاقتراحات والانتقادات يرجى إرسالها على guru.python[at]gmail.com

كل الشعارات و العلامات التجارية و الرموز المستخدمة في هذا الكتاب تعود ملكيتها إلى أصحابها.

Chapter 3 (Basics)

هنتناول فى الفصل دا مفاهيم اساسية زى ال variables وال Loops وال Conditions وهى جزء حيوى من كل اللغات

اولا ماهو ال variable (متغير)؟
هو قيمة متغيرة فى برنامج وبيهمك انك تتابعها وتتابع اى تغير يتم فيها.
مثلا حساب بنكى فيه فلوس "قيمة" قابلة للتغيير فلازم نعبر عنها بمتغير وليكن "money" ونشوف قيمته خلال برنامجنا سواء بالزيادة او النقص

افتح ال IDLE .. هشتغل بنظام تفاعلى فى الأكواد الصغيرة..

```
>>>balance=90000 #an integer
```

نطبع القيمة كالتالى

```
>>> print balance
90000
```

نضيف ليه قيمة ولتكن 100 (زيادة)

```
>>> balance = balance + 100
>>> print balance
90100
```

Preview from Notesale.co.uk
Page 14 of 341

ننقص منه اى قيمة وليكن 890 (نقص)

```
>>> balance = balance - 890
>>> balance
89210
```

على فرض اننا عندنا variable بإسم name بيعبر عن قيمة معينة لل name دا
ملحوظة ال variable ماهو الا alias لقيمة متغيرة فى برنامجك

```
>>> name="ahmed" #a string
>>> print name
ahmed
```

وهكذا تقدر تعمل متغير يعبر عن العمر مثلا age

```
>>> age=50
>>> print age
50
```

بتقوم بحذف ال chars من الستيرنج وفى حال عدم تحديدها هيعتبر انها المسافات

.lstrip([chars])

مثل الأولى ولكن من على اليسار فقط

rstrip([chars])

مثل الأولى ولكن من اليمين فقط

.isalpha()

بتختبر هل الستيرنج مكون من حروف الأبجدية او لأ

.isalnum()

بتختبر هل الستيرنج مكون من حروف من الأبجدية او ارقام او لأ

.isdigit()

بتختبر هل الحروف المكونة للستيرنج عبارة عن ارقام او لأ

.isupper()

بتختبر هل الستيرنج فى حالة uppercase او لأ

.islower()

بتختبر اذا كان الستيرنج فى حالة lowercase أو لأ

.isspace()

بتختبر هل الستيرنج دا مسافة

.istitle()

بتختبر هل الستيرنج ا title أو لأ

.splitlines()

تقوم بإعادة الأسطر فى قائمة List

Preview from Notesale.co.uk
Page 22 of 341

ويوجد العديد بكل تأكيد تقدر تطلع على باقى المميزات فى الوثائق الخاصة بالبيثون

ملحوظة هامة ال string مش mutable! -غير قابلة للتغيير-
لتستخدم mutable strings تابع قسم ال UserString

UserString

المشكلة:

```
somestring="hola"
print somestring, ", ", id(somestring)

somestring += " guyZ"
print somestring, ", ", id(somestring)
```

الناتج

```
hola ,  3084138464
hola guyZ ,  3084161896
```

Loops

while

مثل ال conditions معظم حياتنا مبنية على التكرار.. مثل كل يوم تصحى الصبح وتاخد دش وتفطر وتنزل شغلك وهكذا لحد الأجازة مثلا تنام للمغرب :D

طالما انت فى دراسة او شغل "مش فى اجازة"
اصحى الساعة 6
خد دش
افطر
انزل شغلك

الصيغة العامة

```
while condition  do
        while_suite
```

Preview from Notesale.co.uk
Page 32 of 341

تعالى نجرب ابسط لوب ممكن

```
>>> i=0
>>> while i<10:
        print "i: ", i
        i += 1


i:  0
i:  1
i:  2
i:  3
i:  4
i:  5
i:  6
i:  7
i:  8
i:  9
```

لاحظ ان الكود بيتنفذ كالتالى
الشرط هو ان ا تكون اقل من 10 وطالما الشرط دا حقيقى (صحيح) هيتنفذ البلوك التابع لل loop -الحلقة او الدوارة-

```
        print "i: ", i
```

```
Now count is  6
Now count is  7
t was found at word[7]
```

Our dance will continue

بتستخدم continue للهروب من الحلقة الحالية (ربما لعدم استيفاء عنصر الشروط المطلوبة للعمل عليه) واستكمالها على العنصر الذى يليه مثلا

```
tstvars=['123mx', 'hello', 'acc', '9']

for var in tstvars:
   if var[0].isdigit():
      continue #no work will be done on this item, maybe the next?
   else:
       print var, " => ", "is valid."
```

هنا نختبر كل عنصر من عناصر tstvars ماذا كان يصلح ان يكون اسم متغير فى بايثون
فنقوم بعمل حلقة على العناصر

```
for var in tstvars:
```

Preview from Notesale.co.uk
Page 36 of 341

ونختبر ماإذا كان يبدأ برقم (احد الشروط عدم بدا تسمية المتغيرات فى بايثون برقم) فإذا كان رقم نهرب من الحلقة الحالية ونستكمل على العنصر التالى فى القائمة tstvar

```
   if var[0].isdigit():
       continue #no work will be done on this item, maybe the next?
```

تدريب:
اكتب برنامج لتسجيل الدخول بالبينات التالية

user_name="Ahmed"
user_pass="123456"

ل 3 محاولات وفى حالة الفشل تطلع رسالة ب
!Account Suspended
*أستخدم raw_input للحصول على الداتا من المستخدم

```
>>> val=add(2, 3)
5
>>> print val
None
```

*ايه دا ؟ انا كنت متوقع ان val هتكون قيمتها 5!
دا هيتم فى حالة واحدة ان الطرف اليمين من ال expression تكون قيمته 5 لكن add(2, 3) قيمته None لأنه بيعمل print لمجموع الرقمين لكن مش بيعمل بيهم return

*ايه None ؟

```
None=null=nil=Nothing
```

جميل طب ازاى اخلى قيمة ال Function تساوى مجموع الرقمين ؟
بسيطة اعمل return بالمجموع!
كالتالى مثلا

```
>>> def add(first, second):
        return first+second

>>> val=add(2, 3)
>>> print val
5
```

Preview from Notesale.co.uk
Page 41 of 341

هنا return عبرت عن **قيمة** ال Function

كتير للأسف مش يعرف الفرق بين ال Functions وال Procedures
على كل حال اعتبرها كالتالى
ال Procedure هو اى Function ليس لها return

--لمبرمجى السى/++ والجافا اى Function ال return لها void يبقة اسمها Procedure

بايثون بتقدم ليك العديد من ال Functions الجاهزة مثل

raw_input(prompt)

للحصول على الداتا من المستخدم

input(prompt)

زى ماقلنا هى بتستدعى التالى

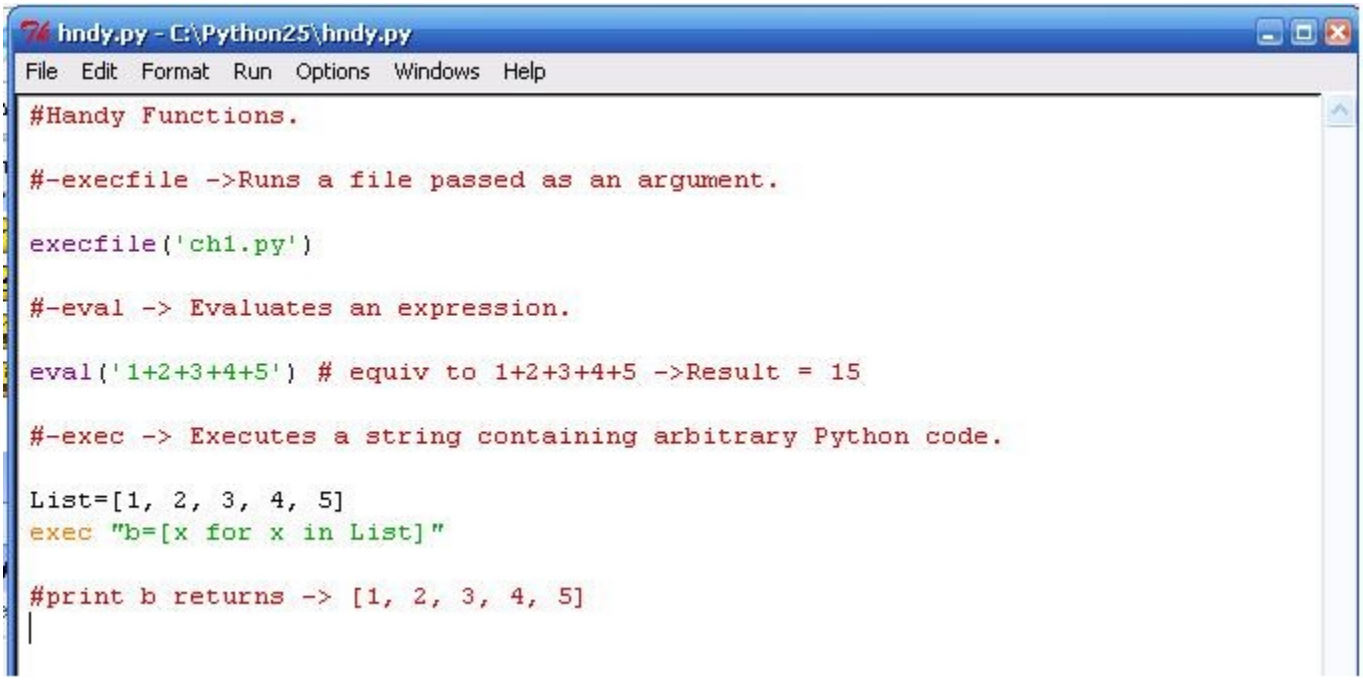
eval(raw_input(prompt))

eval(expression)

بتحقق قيمة ال expression

abs(number)

بتعيد ليك ال Absolute value -القيمة المطلقة- وهى العدد بدون اشارة



execfile(filepath)

تقوم بتنفيذ ملف بايثون

exec expr

تقوم بتنفيذ تعبير بايثون

Preview from Notesale.co.uk
Page 48 of 341

Lambda/Anonymous Functions

قراءتك لهذه الجزئية اختيارية

بايثون بتتيحلك استخدام الدوال المجهولة ودا باستخدام lambda مستعارة من لغة lisp

```
def getName(name):
    return name

anonyFunc=lambda name: name

print anonyFunc("Mido")
print getName("Mido")

#Output:
#Mido
#Mido
```

مابعد lambda هو ال args ومايليها هو ال return

```
    return self.work_hours

  def decrement_workinghours(self, hours):
     self.work_hours -= hours
    return self.work_hours

  def mul_workinghours(self, hours):
     self.work_hours *= hours
    return self.work_hours
```

حل آخر : هو إنك تعمل Overload للـ Operators الـ + و – و * كالتالى

```
  def __add__(self, hours):
     self.work_hours += hours
    return self.work_hours
  def __sub__(self, hours):
     self.work_hours -= hours
    return self.work_hours

  def __mul__(self, hours):
     self.work_hours *=hours
     return self.work_hours
```

Preview from Notesale.co.uk
Page 68 of 341

هيكون صورة الصف كالتالى

```
class Worker(object):

  def __init__(self, name, work_hours):
     self.name=name
     self.work_hours=work_hours

  def increment_workinghours(self, hours):
     self.work_hours += hours
     return self.work_hours

  def decrement_workinghours(self, hours):
     self.work_hours -= hours
     return self.work_hours

  def mul_workinghours(self, hours):
     self.work_hours *= hours
     return self.work_hours

  def __add__(self, hours):
     self.work_hours += hours
     return self.work_hours
```

```
    def __sub__(self, hours):
        self.work_hours -= hours
        return self.work_hours

    def __mul__(self, hours):
        self.work_hours *=hours
        return self.work_hours
```

اعمل Object من ال Class وليكن w

```
>>> w=Worker('EVAN', 4)
>>> w.increment_workinghours(3)
7
>>> w.decrement_workinghours(2)
5
>>> w.mul_workinghours(2)
10
```

انا شايف إن الإسلوب دا ممل جدا مع إنه ممكن يكون اسهل من بعض الشئ ولكنه ممل!

اعمل Object تانى وليكن w1

Preview from Notesale.co.uk
Page 69 of 341

```
>>> w1=Worker('ANN', 5)
>>> w1+2
7
>>> w1-4
3
>>> w1*5
15
```

جدول بكل المعاملات + ال Magic Methods الخاصة بيهم لتعريفهم

```
+ __add__, __radd__
- __sub__, __rsub__
* __mul__, __rmul__
/ __div__, __rdiv__, __truediv__ (for Python 2.2),
__rtruediv__ (for Python 2.2)
// __floordiv__, __rfloordiv__ (for Python version 2.2)
% __mod__, __rmod__
** __pow__, __rpow__
<< __lshift__, __rlshift__
>> __rshift__, __rrshift__
& __and__, __rand__
```

```
    self.__kw=kwargs
    for k, v in kwargs.items():
        self.__setattr__(k, v)
__str__=lambda self: str(self.__kw)
```

بحيث نقدر نستخدمها كالتالى

```
Days=KVEnum(sunday=0, monday=9, tuesday=2)
print Days
print Days.monday
```

Preview from Notesale.co.uk
Page 72 of 341

```
    def getColor(self):
        return self.__color


    def setColor(self, value):
        self.__color = value

    #properties...
    name = property(getName, setName, None, "Gets/Sets name.")

    sex = property(getSex, setSex, None, "Gets/Sets sex.")

    color = property(getColor, setColor, None, "Gets/Sets color.")

    __str__=lambda self: "<Human object: %s >"%self.__name

    __unicode__=__str__

    #methods..
    def eat(self):
        #Eating
        pass

    def drink(self):
        #Drinking
        pass

    def sleep(self):
        #Sleeping
        pass
```

Preview from Notesale.co.uk
Page 76 of 341

هنا عرفنا صنف جديد بيعبر عن الإنسان وليه متغيرات داخلية زى الإسم واللون والنوع وبعض الميثودز لمعالجتهم وميثودز اخرى مثل eat, drink, sleep

لاحظ ان النوع لازم يكون موجود فى Gender.Options المعرفة كالتالى

```
class Gender(object):
    Male, Female="Male", "Female" #0, 1 whatever!
    Options=(Male,Female)
```

Multiple Inheritance

مقدمة:

الوراثة المتعددة هى ميزة موجودة فى بعض اللغات الكائنية تسمح لصف بان يورث اكثر من صف كما فى المثال التالى سنجد ان هناك المستذئب يرث الإنسان والذئب ويجمع بين ميزاتهم

```
class Wolf(object):
    def __init__(self):
        self.__bite=True

    can_bite=lambda self:self.__bite

    def fullmoon(self):
        print "Wooooooooooooooooooooooooo"

    def bite(self, h):
        print "I'm cursed"

class Werewolf(Human, Wolf):

    def __init__(self, name, color, sex):
        Human.__init__(self, name, color, sex)
        Wolf.__init__(self)
    yasser=Werewolf("yasser", "black", Gender.Male)
    print yasser.__dict__
    print "Can bite? ", yasser.can_bite()
    yasser.bite(ahmed)
    print Werewolf.__bases__
```

Preview from Notesale.co.uk
Page 81 of 341

```
class Wolf(object):
    def __init__(self):
        self.__bite=True


    can_bite=lambda self:self.__bite

    def fullmoon(self):
```

Chapter 7 (Exceptions, Errors)

مش فى شئ كامل وطالما كتبت كود توقع انك ستجد فيه مشكلات! مثلا انقطاع الإتصال مع الداتابيز او مشكلات فى النتورك او مكتبيات غير متوافرة او كود خاطئ SyntaxError او بيانات دخلها المستخدم بصورة خاطئة او خطئ رياضي او او
فنحن لانقوم بكتابة software writing ولكن اعادة كتابة software rewriting لماذا نستخدم text editor وليس text writer فى رأيك :P ؟

اولا ايه هى ال exceptions ؟ هى شئ حصل يتسبب فى تغيير مسار برنامجك "اللى انت خططه" المفتاح للموضوع دا 4 حاجات (مش شرط يكون خطأ) ولكن شئ غير محسوب

1- try
2- except
3- finally
4- raise

try مش فيها شئ عجيب كل المطلوب منك هو انك تضيف الجزء اللى هتشك ان ممكن يحصل فيه exception

except: بتصطاد فيها ال exception وتعالجه

finally : الكود المرتبط بها هيتم تنفيذ دائما حتى لو مش عالجت ال exception بتكون غالبا لإغلاق المصادر المفتوحة (مثلا فايل او كونكشن معين)
raise: هى المسئولة عن اطلاق ال exceptions دى

تعالى نشرحها بمثال شهير جدا
هنحاول نقسم عدد على صفر

```
>>> 1/0
```

Preview from Notesale.co.uk
Page 84 of 341

رد بايثون

```
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    1/0
ZeroDivisionError: integer division or modulo by zero
>>>
```

لاحظ السطر ZeroDivisionError: integer division or modulo by zero
بيتكلم فيه عن حدوث Error للقسمة على صفر + وصف ال Error
طب جميل جدا انا كدا عرفت ان ممكن يتقسم على صفر طب انا عايز اصطاد ان حد حاول يعمل كدا
1- ضع الكود بتاعك فى try بلوك
2- هندل -تعامل مع- ال exception فى except بلوك

```
try:
        print 1/0
except ZeroDivisionError, e:
        print e.message
```

```
        except Exception:
                raise IntOnly("Integers only are allowed.")
```

3- اكتب كود برنامجك

```
>>> try:
        j=readInt()
except IntOnly, e:
        print e.message
```

هنا بنستدعى readInt ونسند قيمتها ل j فى ال try block
فى حالة حدوث ايرور(خطأ) من النوع IntOnly هنعمل منه كائن (اللى هو e) ونشوف e.message
ودا الأسلوب المفضل
تقدر تتعامل مع كذا اكسبشن بإستخدام except حسب ماتحب طالما الكود بتاعك فيه مشاكل

مثال على finally

```
#!/usr/bin/python


f = None #out of try block. As finally doesn't have try' context

try:
   f = file('somefile, 'r')
   lines = f.readlines()
   for line in lines:
      print line, #avoid \n\n !
except IOError, e:
   print 'IOErrorError'
finally: #cleaning up
   if f:
      f.close()
```

Preview from Notesale.co.uk
Page 86 of 341

كدا وصلنا للصيغة العامة وهى

```
try:
        suite
except EX1, ex1: suite

except EX2, ex2: suite

except EX3, ex3: suite

finally: suite
```

Chapter 8 (IO)

Basic IO

هنتكلم فى الفصل دا عن التعامل مع ال files وتحديدا ال IO اختصارا ل Input/Output
ال file type بيعبر عن صف مسئول عن التعامل مع الملفات (ممكن يكون سوكيت او غيره بما ان كل شئ عبارة عن file)

فى اكثر من طريقة لإنشاء file object ودا عن طريق open او file class
open هتعمل ريترن ب file object وهى معرفة كالتالى

```
open(name, mode, buffering)
```

حيث name هو مسار الملف
mode هو بيعبر عن حالة الوصول (الملف مفتوح للقراءة ، للكتابة ، للإضافة ؟) الإفتراضى هو r للقراءة

r → قراءة فقط
w → كتابة فقط (بيتم محو كل البيانات الموجودة)
a → للإضافة من عند النهاية ،مع عدم محو البيانات
r+ → قراءة وكتابة
w+ → كتابة وقراءة

اى من ال modes السابقة اذا لحقتها اختصارا ل binary سيتم التعامل مع binary read, binary write, .. etc

ال buffering لتحديد هل سيتم عمل اى buffering فى خلال التعامل مع الملف، الإفتراضى هو -1 (بيتم النقل لنظام التشغيل)

```
.close()
```

لغلق ال file object

Preview from Notesale.co.uk
Page 89 of 341

```
.read(num=None)
```

بتقوم بقراءة عدد معين من البايتات وفى حال عدم تحديده بيتم قراءة الملف بكامله

```
.readline()
```

قراءة سطر واحد

```
.readlines()
```

قراءة كل السطور (على صورة list)

```
.tell()
```

بتخبرنا بالمكان الحالى

```
.fseek(offset, whence=0)
```

بتنقل المكان الحالى الى offset معين بعد ال whence
whence ربما تكون 0 (اى بداية الملف) او 1 (المكان الحالى) او 2 (نهاية الملف)
عندنا ملف سميناه iotest.xcd فيه التالى 0123456789

```
.write(s)
```

يقوم بكتابة s فى الملف

```
.writelines(seq_of_strings)
```

يقوم بكتابه كل عناصر seq_of_strings فى الملف
--كقيامك بإستدعاء write على كل عنصر فى هذه ال sequence

```
.fileno()
```

الحصول على File Descriptor خاص بال file

```
.flush()
```

لتأكيد نقل ال buffer الداخلى كله على الملف

```
.name
```

Preview from Notesale.co.uk
Page 100 of 341

Preview from Notesale.co.uk
Page 108 of 341

```
def deleteUser(self, username):

    sqlstmt="DELETE FROM users WHERE username='%s'"%username
    try:
        self._sqlcursor.execute(sqlstmt)
        self._sqlconnection.commit() #commit
    except Exception, e:
            print e.message
```

تسجيل دخول

```
def login(self, username, pwd):

    md5edpass=self._md5(pwd)
    sqlstmt="SELECT username, password FROM users WHERE username='%s' AND password='%s'"%
(username, md5edpass)
    self._sqlcursor.execute(sqlstmt)
    if self._sqlcursor.fetchone():
            self.setState(State.Online, username)
```

Preview from Notesale.co.uk
Page 114 of 341

تغيير الحالة

```
def setState(self, state, username):

    sqlstmt="UPDATE users SET state=%d WHERE username='%s'"%(state, username)
    try:
        self._sqlcursor.execute(sqlstmt)
    except Exception, e:
            print e.message
```

عرض الكل

```
def getAllUsers(self):

    sqlstmt="SELECT username, state FROM users"
    self._sqlcursor.execute(sqlstmt)
    for row in self._sqlcursor.fetchall():
        yield row[0], row[1]
```

ملحوظة: انا هنا ناقشت MySQLdb من خلال مفهوم ال DB API بمعنى ان نفس المبادئ هتلقيها ثابتة فى اى انترفيس هتستخدمها
ومازلنا منتظرين DB API 3

```
class Person(object):

    def __init__(self, name, phone):
        self.name=name
         self.phone=phone
```

ننشئ connection و cursor ولكن ننبه ال database انها تعمل parse لل declared types زى ال Person مثلا .. هنغير شوية ونتعامل مع ال memory

```
#create a connection.
memConnection=SQLite.connect(':memory:', detect_types=SQLite.PARSE_DECLTYPES)

#cursor
cursor=memConnection.cursor()
```

الآن ننشئ table بحيث إنه ياخد 2 fields وهم ال ID, information

```
cursor.execute("CREATE TABLE test (id INTEGER PRIMARY KEY, p person)")
```

جميل جدا .. ناقص إننا نحدد إزاى ال object الخاص بنا يتحول لـ string وازاى نجمع ال data بتاعته تانى من ال string دا
ملحوظة: إحنا بنتكلم على مجرد text بإستخدم toString method مثلا ..مش serializing objects او Pickling

```
def adaptPerson(person):
    return "%s;%s" %(person.name, person.phone)
```

وكيفية التجميع .. بكل بساطة إحنا حولنا ال fields بتاعت ال Person object ل string ودمجناهم ب ;.. يبقة نقدر نجمعهم بإننا نفصل ال ; ونمرر ال قيم الخاصة بال fields دى لل Constructor وننشئ object منها

```
def convToPerson(text):
    name, phone=map(str, text.split(";"))
     return Person(name, phone)
```

بعد ماعملنا الميثودز الخاصة بالتحويل والتجميع .. كل اللى ناقص اننا نبلغ SQLite بكدا

```
SQLite.register_adapter(Person, adaptPerson)
SQLite.register_converter("person", convToPerson)
```

ننشئ بعض الكائنات

Preview from Notesale.co.uk
Page 120 of 341

```
    def __init__(self, name, phone):
        self.name=name
        self.phone=phone

    def __repr__(self):
        return "(%s;%s)" %(self.name, self.phone)

#define a method to register it..

def adaptPerson(person):
    return "%s;%s" %(person.name, person.phone)

def convToPerson(text):
    name, phone=map(str, text.split(";"))
    return Person(name, phone)

SQLite.register_adapter(Person, adaptPerson)
SQLite.register_converter("person", convToPerson)

p1=Person("ahmed", "12345678")
p2=Person("rul3z", "89745632")


#create a test table..
cursor.execute("CREATE TABLE test (id INTEGER PRIMARY KEY, p person)")

#add
cursor.execute('INSERT INTO test VALUES(null, ?)', (p1, ))
cursor.execute('INSERT INTO test VALUES(null, ?)', (p2, ))

#select..
cursor.execute('SELECT * from test')
for row in cursor:
    print row

#clean-up
cursor.close()
memConnection.close()
```

Preview from Notesale.co.uk
Page 122 of 341

وللمزيد راجع التالى :

http://www.devshed.com/c/a/Python/Us...ite-in-Python/
http://www.initd.org/tracker/pysqlite/wiki/basicintro
http://www.initd.org/pub/software/py...ined-functions

ملحوظة: sqlite3 اصبحت وحدة اساسية فى بايثون تقدر تطبق نفس الفصل عليها

```
...     name=Unicode()
...     npages=Int()
```

ننشئ قاعدة بيانات

```
>>> db=create_database("sqlite:")
```

ننشئ كائن مخزن (ليتعامل مع عناصر قاعدة البيانات)

```
>>> store=Store(db)
```

ننشئ الجدول المعبر عن الكتب

```
>>> store.execute("CREATE TABLE book (id INTEGER PRIMARY KEY, name VARCHAR,
npages INTEGER)")
```

ننشئ كائن

```
>>> rbook.name=u"Introduction to Ruby"
>>> rbook.npages=210
>>> print rbook.id, rbook.name, rbook.npages
None Introduction to Ruby 210
```

Preview from Notesale.co.uk
Page 125 of 341

فلنضيفه الآن

```
>>> store.add(rbook)
<__main__.Book object at 0xb78aecac>
>>> print rbook.id, rbook.name, rbook.npages
None Introduction to Ruby 210
>>>
>>> pybook.name=u"PyGuide"
>>> pybook.npages=230
>>> print pybook
<__main__.Book object at 0xb78ae80c>
>>> store.add(pybook)
<__main__.Book object at 0xb78ae80c>
```

الحصول على سجل ما

```
>>> pythonbook=store.find(Book, Book.name==u"PyGuide").one()
>>> pythonbook.name
u'PyGuide'
```

Preview from Notesale.co.uk
Page 129 of 341

ايه المتغيرات دى كلها ؟
self._total للتخزين المجموع الكلى للأسعار
self._curel لتخزين اسم العنصر اللى بيتم معالجته
self._curid لتخزين اخر id تم قرائته
self._booksInfo تخزين معلومات عن الكتاب مكونة من ال name, id
self._authors تخزين اسماء الكتاب

```
def getTotal(self):
    return self._total

def getBooksInfo(self):
    return self._booksInfo

def getAuthors(self):
     return self._authors
```

عرفنا getters للوصول للمتغيرات الداخلية
ملحوظة يفضل تستخدم properties مع lambda

```
booksinfo=property(fget=lambda self: self._booksInfo)
authors=property(fget=lambda self: self._authors)
total=property(fget=lambda self: self._total)
```

Preview from Notesale.co.uk
Page 138 of 341

```
def startDocument(self):
    #print "Starting Document."
        pass
```

لو حبيت تضيف اى رسالة او اى حاجة على هواك يتم تنفيذها عند بداية قراءة الملف

```
def endDocument(self):
    #print "Ending Document."
        pass
```

نفس السابقة ولكن عند انتهاء القراءة

```
def startElement(self, el, attrs):

    #print "Starting ", el
    self._curel=el
        if el=="book":
                #get the id..
                self._curid=attrs.getValue("id") #attrs["id"]
```

```
ARGV:  ['echo.py', 'Hello World']
Argv[0]: echo.py
Argv[1]: Hello World
```

ماذا عن enumerate ؟

هى دالة تقوم بإعادة index (لعدد الدورات) وقيمة من container (مثلا list ك argv)

Gimme usage!

تطبيق جيد ايضا ان تضع دالة بإسم usage توضح كيفية استخدام البرنامج
قم دائما بإختبار عدد المعاملات اللتى تم ارسالها للسكريبت فأى عدد غير مقبول قم بعرض ال usage

Preview from NotesSale.co.uk
Page 157 of 341

```
def usage():
        """My fancy usage helper"""
        ....

def consoleMain():
        if len(argv) != DEFINED_LENGTH:
                GIMME_USAGE

if __name__=="__main__":
        consoleMain()
```

GIMME optparser!

بايثون كالعادة توفر لك العديد والعديد لمساعدتك فتوفر لك اكثر من وحدة لمعالجة معاملات سطر الأوامر لاحظ الإستخدام التالى

```
striky@striky-desktop:~$ python mufhrs.py -f http://linuxac.org/forum/forumdisplay.php?f=23 -l 1 -u 3 -s 1 -t vb > pgfihrsx2.txt
```

3- اضافة اسماء المعاملات بإستخدام الطريقة add_option واللتى تأخذ معاملات عديده اهمها
1- الصورة المختصرة لإسم الإختيار -f
2- الصورة الطويلة (الكاملة) لإسم الإختيار -forum
3- اسم من اختيارك للحصول على قيمته وليكن forumlink مثلا
4- قسم المساعدة الخاص بال option

```
optsparser.add_option("-f", "--forum", dest="forumlink", help="Forum Section")
optsparser.add_option("-l", "--lower", dest="lower", help="Lowest page")
optsparser.add_option("-u", "--upper", dest="upper", help="Upper page")
optsparser.add_option("-s", "--step", dest="step", help="Step")
optsparser.add_option("-t", "--type", dest="ftype", help="Forum type (e.g) vb")
```

عند استدعاءك للبرنامج فى وضع المساعدة help يتم تنفيذ شئ مشابه للتالى

```
Options:
  -h, --help            show this help message and exit
  -f FORUMLINK, --forum=FORUMLINK
                        Forum Section
  -l LOWER, --lower=LOWER
                        Lowest page
  -u UPPER, --upper=UPPER
                        Upper page
  -s STEP, --step=STEP  Step
  -t FTYPE, --type=FTYPE
                        Forum type (e.g) vb
striky@striky-Desktop:~/workspace/python/src$ python mufhrs.py --help
Usage: mufhrs.py [options]

Options:
  -h, --help            show this help message and exit
  -f FORUMLINK, --forum=FORUMLINK
                        Forum Section
  -l LOWER, --lower=LOWER
                        Lowest page
  -u UPPER, --upper=UPPER
                        Upper page
  -s STEP, --step=STEP  Step
  -t FTYPE, --type=FTYPE
                        Forum type (e.g) vb
```

Preview from Notesale.co.uk
Page 159 of 341

رائعة اليس كذلك ؟
للحصول على قيم الإختيارات

```
forumlink=optsparser.values.forumlink
lower=int(optsparser.values.lower)
```

نبدأ من اليمين لليسار الطريقة communicate تعيد لنا tuple تشمل الخرج والخطأ فنأخذ العنصر الأول وهو الخرج (ناتج تنفيذ العملية)
Popen هو صف يأخذ اول عنصر args الأمر والمعاملات
stdout, stdin, stderr هى الخرج والإدخال والخطأ الخاصين بالعملية ويأخذو قيم None او PIPE (قيمة خاصة لتعلم بوجوب فتح انبوبة pipe) او file object او رقم ليعبر عن file descriptor موجود
shell قيمة منطقية (True لتعبر عن تنفيذ البرنامج من خلال الشيل او False ليتم التنفيذ من خلال excevp (لاتهتم الآن))

Replacer

ربما لديك مجموعة كبيرة من الملفات تحتاج القيام بتعديل سريع عليها جميعا ؟
فى هذه الجزئية سننشئ سكربت للتعديل على تلك الملفات بإستخدام
1- القواميس dict
2- ال recursion (وهى استدعاء الدالة لنفسها)

Preview from Notesale.co.uk
Page 162 of 341

حسنا هاهو السكربت

```
#!bin/python


#Imports#
import os
import os.path as op
import time
from sys import argv



class Replacer(object):

    def __init__(self, parentDir, dic={}, exts=[]):
        self.__parentDir=parentDir
        self.__dic=dic
        self.__exts=exts
        self.__replacingTimes=0


    def __getExt(self, s):
```

```
    #idx=s.rfind(".")
    #ext=s[idx:]
    #return ext
        return op.splitext(s)[1]

def replace(self):
    start=self.__parentDir
    for e in os.listdir(start): #foreach entry in os.listdir..
        path=start+op.sep+e
        if os.path.isfile(path):
            ext=self.__getExt(path)
            if not ext in self.__exts:
                continue #Re-Loop..
            #replace..
            f=file(path, "r")
            src=f.read()
            f.close()
            for key in self.__dic:
                self.__replacingTimes += src.count(key)
                src=src.replace(key, self.__dic[key])

            f=file(path, "w")
            f.write(src)
            f.close()
        elif os.path.isdir(path):
            #RECURSE..
            rep=Replacer(path,self.__dic, self.__exts)
            rep.replace()
        else:
            continue
```

Preview from Notesale.co.uk
Page 163 of 341

الفكرة هى اعطاء مجلد لتعديل الملفات اللتى بداخله
وإذا كان داخله مجلد يتم فتح ذلك المجلد للتعديل على مافى داخله وهكذا

نبدأ ب

```
def __init__(self, parentDir, dic={}, exts=[]):
    self.__parentDir=parentDir
    self.__dic=dic
    self.__exts=exts
     self.__replacingTimes=0
```

تحديد مجلد الأب parentDir
تحديد الكلمات القديمة والجديدة فى قاموس ليتم استبدالهم
تحديد الإمتدادات القابل العمل عليها

```
import socket
import threading


class ChatServer(object):
    '''Indexer...'''

    def __init__(self, port):

        self.port=port
        addr=('', self.port)
        self._bufsize=2048

        self.listener=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.listener.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1) #Quick restarts.
        self.listener.bind(addr)
        self.alSocks=[]
        #self.tListening=threading.Thread(target=self.listeningHandler, args=[])
        #self.tListening.start()

            self.listeningHandler() #Start listening...



    def listeningHandler(self):
        self.listener.listen(5)
        print "Server started..."
        while True:
            clientSocket, clientAddr=self.listener.accept()
            #Handle the client in a new thread...
            self.tHandleClient=threading.Thread(target=self.clientHandler, args=[clientSocket])
            self.tHandleClient.start()


    def clientHandler(self, clientSocket):

        self.alSocks += [clientSocket]
        print "connection from: ", clientSocket.getpeername()
        self._bufsize=2048
        try:
            while True:

                data=clientSocket.recv(self._bufsize)
                if not data:
                    break
                #handle sending all recieved in another thread..
                #serverToAll=threading.Thread(target=self.serverToAll, args=[clientSocket, data])
                #serverToAll.start()
                self.serverToAll(clientSocket, data)
```

Preview from Notesale.co.uk
Page 175 of 341

```
  alias=raw_input("Alias: ")
  sharedFiles=os.listdir(os.getcwd())
  peer=Peer(alias, ('localhost', 8080), sharedFiles)
```

طبعا تقدر تظبطها بحيث إنك تمرر ال addr الخاص بال server من ال command line او حتى من prompt الأكواد

```
Discovery Server
Code:
#!bin/python
import socket
import sys
import os
import threading
from utils import serialize, deserialize, PeerInfo
class NotSupportedCommand(Exception):
  pass
class DiscoveryServer(object):
  '''Indexer...'''
  def __init__(self, port, maxPeers=5):
    self.port=port
    addr=('', self.port)
    self.maxPeers=maxPeers
    self.supportedCommands=["/register", "/setNick", "/setSharedFiles", "/showall", "/query"]
    self.listener=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    self.listener.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    self.listener.bind(addr)
    self.tListening=threading.Thread(target=self.listeningHandler, args=[])
    self.tListening.start()
    self.alSocks=[]
   # {clientAddr:peerInfo Object}
    self.db={}
    self.log=[]
    self.BUF=2048

  def listeningHandler(self):
    self.listener.listen(self.maxPeers)
    print "Server Started..."
    while True:
      clientSocket, clientAddr=self.listener.accept()
```

Preview from Notesale.co.uk
Page 190 of 341

```
import socket
import sys
import os
import threading
from utils import serialize, deserialize, PeerInfo
import random as rnd

class Peer(object):
  def __init__(self, alias, serverAddr=(), sharedFiles=[]):

    self.alias=alias
    self.serverAddr=serverAddr
    self.sharedFiles=sharedFiles
    self.tcpClient=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    self.tcpClient.connect(self.serverAddr)
    self.BUF=1024
    self.listeningPort=rnd.randint(8081, 10000)
    self.pInfo=PeerInfo(self.alias, self.sharedFiles, self.listeningPort)

    print "\nConnected to server..."
    self.tClientToServer=threading.Thread(target=self.clientToServerHandler, args=[])
    self.tClientToServer.start()
    #listen for connections in background..
    self.addr=('', self.listeningPort)
    self.listener=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    self.listener.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    self.listener.bind(self.addr)
    self.tListening=threading.Thread(target=self.listeningHandler, args=[])
    self.tListening.start()

  def registerAtServer(self):
    msg=("/register", self.alias, self.sharedFiles, self.listeningPort)
    self.tcpClient.send(serialize(msg))

  def clientToServerHandler(self):
    print "Start Chatting.."
    #first register the files...
    self.registerAtServer()
    while True:
      tServerToClient=threading.Thread(target=self.serverToClientHandler, args=[])
      tServerToClient.start()
      data=raw_input()
      if not data: continue
      if data.lower()=="exit": exit()

      if data.split(" ")[0]=="/fetch":
```

Preview from Notesale.co.uk
Page 194 of 341

```
        f.write(buf)
      except EOFError, eofErr:
        print "EOFError: ", eofErr
      except Exception, e:
        print "Error: ", e
        break
    print "File Downloaded!"
    f.close()
    fetchTCPClient.close()


  def listeningHandler(self):
    self.listener.listen(5)
    while True:
      clientSocket, clientAddr=self.listener.accept()
      tClientHandling=threading.Thread(target=self.clientHandler, args=[clientSocket])
      tClientHandling.start()

  def clientHandler(self, clientSocket):
    rcvd=clientSocket.recv(self.BUF)
    data=deserialize(rcvd)
    if isinstance(data, tuple):
      if data[0]=="/fetch": #go on..
        fileneeded=data[1] #(/fetch, fileneeded, from)
        print "File Request: "+fileneeded
        f=file(fileneeded,"rb")
        while True:
          try:
            buf=f.read(self.BUF)
            if not buf:
              break
            clientSocket.send(buf)
          except Exception, e:
            print "Error: ", e
            break

        f.close()
        clientSocket.close()
        print "Copied!"



if __name__=="__main__":
  alias=raw_input("Alias: ")
  sharedFiles=os.listdir(os.getcwd())
  peer=Peer(alias, ('localhost', 8080), sharedFiles)
```

Preview from Notesale.co.uk
Page 196 of 341

XMLRPC what?

اذا لم تكن مهتما ب XML-RPC فكن حرا للإنتقال للفصل القادم
فى ابسط الصور XML-RPC هو بروتوكول Remote Procedure Call عبر بروتوكول HTTP
للمزيد http://en.wikipedia.org/wiki/XML-RPC

Preview from Notesale.co.uk
Page 201 of 341

نوضح الموضوع بمثال بسيط

```
#!bin/python

from SimpleXMLRPCServer import SimpleXMLRPCServer, SimpleXMLRPCRequestHandler
from SocketServer import ThreadingMixIn

class Greeter(object):

    def hi(self):
        """Returns hi message"""
        return "Hi"

    def bye(self):
        """Returns bye message"""
        return "Bye"

    def say(self, what):
        """Returns Simone says message"""
        return "Simone says: "+ what

class MyServer(ThreadingMixIn, SimpleXMLRPCServer):

    pass
```

```
from DocXMLRPCServer import DocXMLRPCServer, DocXMLRPCRequestHandler
from SocketServer import ThreadingMixIn

class Greeter(object):

    def hi(self):
        """Returns hi message"""
        return "Hi"

    def bye(self):
        """Returns bye message"""
        return "Bye"

    def say(self, what):
        """Returns Simone says message"""
        return "Simone says: "+ what

class MyServer(ThreadingMixIn, DocXMLRPCServer):

    pass

def test():
    addr=('', 40002)
    srvr=MyServer(addr, DocXMLRPCRequestHandler)
    ##server methods...
    srvr.set_server_title("My First DocXMLRPC Server")
    srvr.set_server_name("Greeter DocXMLRPC Server")
    srvr.set_server_documentation("Greeter DocXMLRPC Server is used for learning XML-RPC")
    srvr.register_instance(Greeter())
    srvr.register_introspection_functions()
    srvr.serve_forever()
    #print "Started..."

if __name__=="__main__":
    test()
```

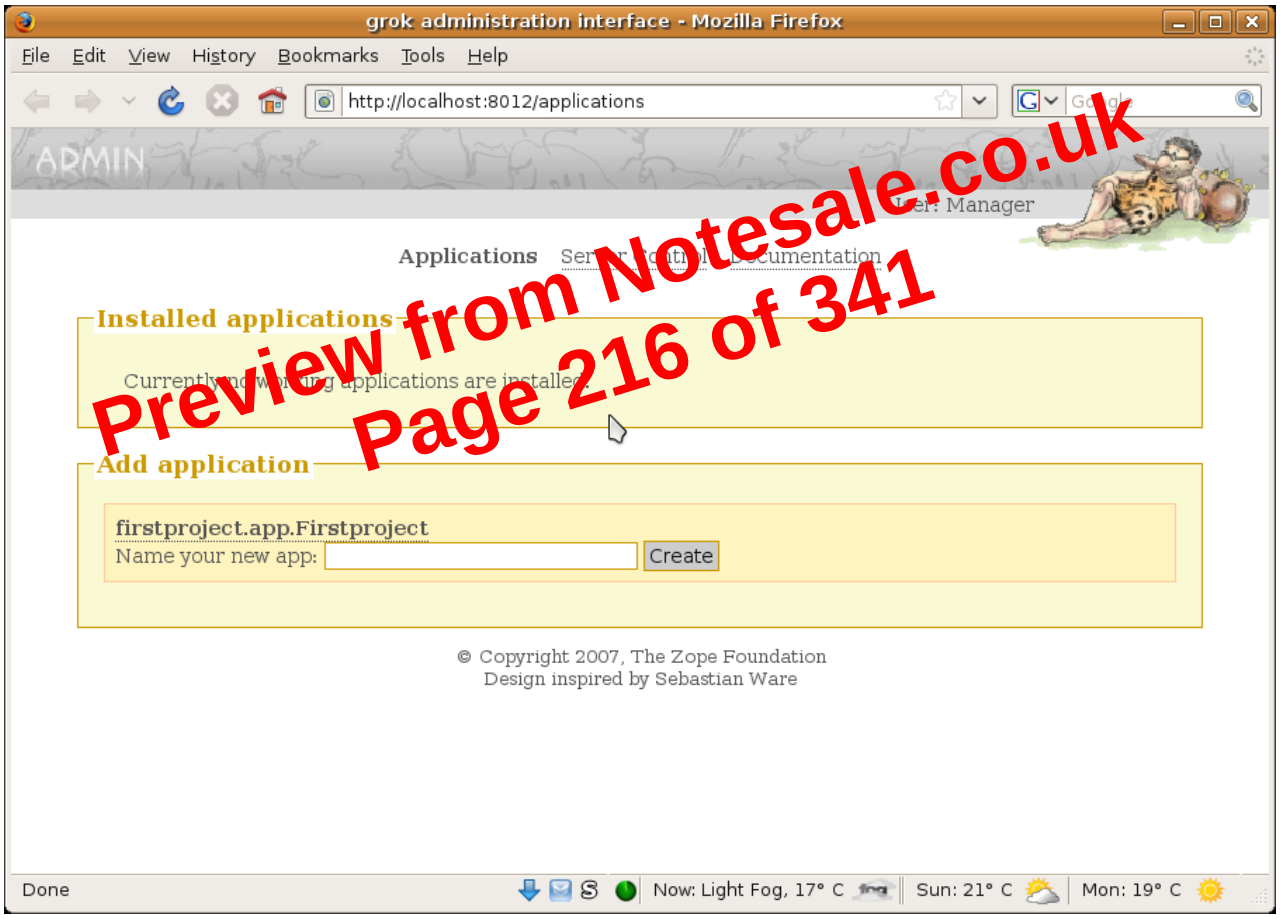
Preview from Notesale.co.uk
Page 206 of 341

Preview from Notesale.co.uk
Page 212 of 341

```
|  `-- firstproject
|      |-- __init__.py
|      |-- app.py
|      |-- app.txt
|      |-- app_templates
|      |   `-- index.pt
|      |-- configure.zcml
|      |-- ftesting.zcml
|      |-- static
|      |   `-- README.txt
|      `-- tests.py
`-- versions.cfg

13 directories, 32 files
```

الهيكلية واضحة ملفات تنفيذيه و templates وملفات static واعدادات الخ الخ

قم بتشغيل zope



Preview from Notesale.co.uk
Page 216 of 341

قم بتسمية التطبيق مثلا hello

الآن التطبيق لديه view واحدة وهى index التى يتم عند استدعاءها عرض ال template فى الملف index.pt
افتح الملف app.py ستجده مشابه للتالى

```
import grok

class Firstproject(grok.Application, grok.Container):
   pass

class Index(grok.View):
   pass # see app_templates/index.pt
```

قم بإضافة 2 views وهما Hi, Bye

```
class Hi(grok.View):
        pass #renders app_templates/hi.pt

class Bye(grok.View):
        pass #renders app_templates/bye.pt
```

Preview from Notesale.co.uk
Page 221 of 341

قم بإنشاء الملفات hi.pt و bye.pt حيث عرضها عند استدعاء ناتج ال views
الملف hi.pt

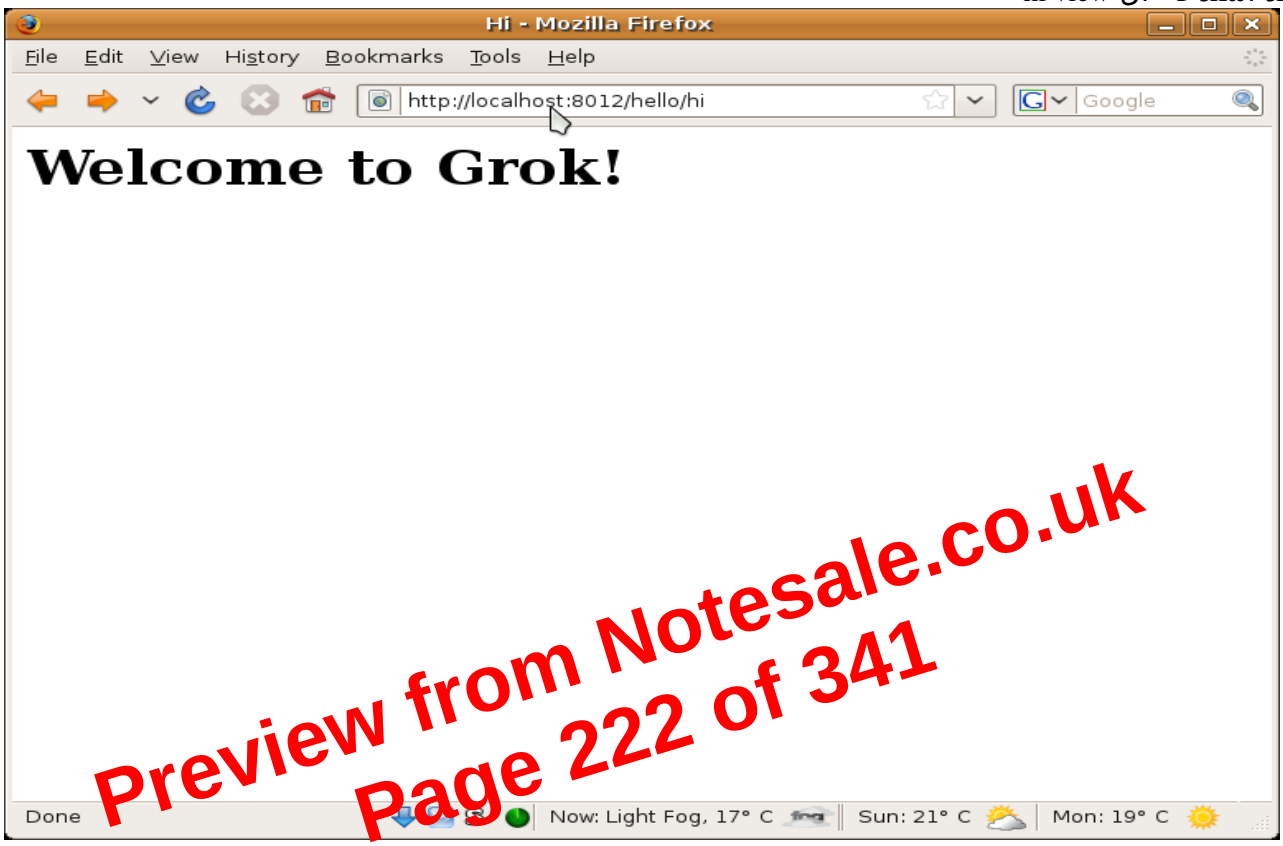
```
<html>
        <head>
                <title>Hi</title>
        </head>
        <body>
                <h1>Welcome to Grok!</h1>
        </body>
</html>
```

الملف bye.pt

```
<html>
        <head>
                <title>Bye</title>
        </head>
        <body>
                <h1>Have to go :(</h1>
```

```
        </body>
</html>
```

عند استدعاء ال hi view



Preview from Notesale.co.uk
Page 222 of 341

عند استدعاء ال bye.view

```
Variables:
  egg:      helloworld
  package:  helloworld
  project:  helloworld
Enter template_engine (mako/genshi/jinja/etc: Template language) ['mako']:
Enter sqlalchemy (True/False: Include SQLAlchemy 0.4 configuration) [False]:
Enter google_app_engine (True/False: Setup default appropriate for Google App Engine) [False]:
Creating template pylons
Creating directory ./helloworld
  Recursing into +package+
    Creating ./helloworld/helloworld/
    Copying templates/default_project/+package+/__init__.py_tmpl to ./helloworld/helloworld/__init__.py
    Recursing into config
      Creating ./helloworld/helloworld/config/
      Copying templates/default_project/+package+/config/__init__.py_tmpl to
./helloworld/helloworld/config/__init__.py
      Copying templates/default_project/+package+/config/deployment.ini_tmpl_tmpl to ./helloworld/helloworld/
config/deployment.ini_tmpl
      Copying templates/default_project/+package+/config/environment.py_tmpl to
./helloworld/helloworld/config/environment.py
      Copying templates/default_project/+package+/config/middleware.py_tmpl to
./helloworld/helloworld/config/middleware.py
      Copying templates/default_project/+package+/config/routing.py_tmpl to
./helloworld/helloworld/config/routing.py
    Recursing into controllers
      Creating ./helloworld/helloworld/controllers/
      Copying templates/default_project/+package+/controllers/__init__.py_tmpl to
./helloworld/helloworld/controllers/__init__.py
      Copying templates/default_project/+package+/controllers/error.py_tmpl to
./helloworld/helloworld/controllers/error.py
    Recursing into lib
      Creating ./helloworld/helloworld/lib/
      Copying templates/default_project/+package+/lib/__init__.py_tmpl to
./helloworld/helloworld/lib/__init__.py
      Copying templates/default_project/+package+/lib/app_globals.py_tmpl to
./helloworld/helloworld/lib/app_globals.py
      Copying templates/default_project/+package+/lib/base.py_tmpl to ./helloworld/helloworld/lib/base.py
      Copying templates/default_project/+package+/lib/helpers.py_tmpl to ./helloworld/helloworld/lib/helpers.py
    Recursing into model
      Creating ./helloworld/helloworld/model/
      Copying templates/default_project/+package+/model/__init__.py_tmpl to
./helloworld/helloworld/model/__init__.py
    Recursing into public
      Creating ./helloworld/helloworld/public/
      Copying templates/default_project/+package+/public/bg.png to ./helloworld/helloworld/public/bg.png
      Copying templates/default_project/+package+/public/index.html_tmpl to
./helloworld/helloworld/public/index.html
      Copying templates/default_project/+package+/public/pylons-logo.gif to
./helloworld/helloworld/public/pylons-logo.gif
```

Preview from Notesale.co.uk
Page 228 of 341

```
|  |-- lib
|  |  |-- __init__.py
|  |  |-- app_globals.py
|  |  |-- base.py
|  |  `-- helpers.py
|  |-- model
|  |  `-- __init__.py
|  |-- public
|  |  |-- bg.png
|  |  |-- index.html
|  |  `-- pylons-logo.gif
|  |-- templates
|  |-- tests
|  |  |-- __init__.py
|  |  |-- functional
|  |  |  `-- __init__.py
|  |  `-- test_models.py
|  `-- websetup.py
|-- helloworld.egg-info
|  |-- PKG-INFO
|  |-- SOURCES.txt
|  |-- dependency_links.txt
|  |-- entry_points.txt
|  |-- not-zip-safe
|  |-- paster_plugins.txt
|  |-- requires.txt
|  `-- top_level.txt
|-- setup.cfg
|-- setup.py
`-- test.ini
```

Preview from Notesale.co.uk
Page 230 of 341

ننشئ controller (وهو مايحوى ال actions مثل ال views اللتى فى grok او webpy)

```
striky@striky-desktop:~/workspace/pytut/src/nettut/helloworld$ paster controller hello
/usr/lib/python2.5/site-packages/RuleDispatch-0.5a1.dev_r2506-py2.5-linux-i686.egg/dispatch/__init__.py:98:
Warning: 'as' will become a reserved keyword in Python 2.6
/usr/lib/python2.5/site-packages/CherryPy-2.3.0-py2.5.egg/cherrypy/lib/profiler.py:54: UserWarning: Your
installation of Python doesn't have a profile module. If you're on Debian, you can apt-get python2.4-profiler
from non-free in a separate step. See http://www.cherrypy.org/wiki/ProfilingOnDebian for details.
  warnings.warn(msg)
/usr/lib/python2.5/site-packages/RuleDispatch-0.5a1.dev_r2506-py2.5-linux-
i686.egg/dispatch/predicates.py:239: Warning: 'as' will become a reserved keyword in Python 2.6
/usr/lib/python2.5/site-packages/RuleDispatch-0.5a1.dev_r2506-py2.5-linux-
i686.egg/dispatch/predicates.py:263: Warning: 'as' will become a reserved keyword in Python 2.6
/usr/lib/python2.5/site-packages/RuleDispatch-0.5a1.dev_r2506-py2.5-linux-
i686.egg/dispatch/predicates.py:281: Warning: 'as' will become a reserved keyword in Python 2.6
Creating /home/striky/workspace/pytut/src/nettut/helloworld/helloworld/controllers/hello.py
Creating /home/striky/workspace/pytut/src/nettut/helloworld/helloworld/tests/functional/test_hello.py
striky@striky-desktop:~/workspace/pytut/src/nettut/helloworld$
```

تمام افتح الملف فى المسار helloworld/controllers/hello ستجده مشابه للتالى

```
import logging

from pylons import request, response, session, tmpl_context as c
from pylons.controllers.util import abort, redirect_to

from helloworld.lib.base import BaseController, render
#from helloworld import model

log = logging.getLogger(__name__)

class HelloController(BaseController):

    def index(self):
        # Return a rendered template
        #   return render('/template.mako')
        # or, Return a response
        return 'Hello World'
```

يتم التعامل مع العناوين كالتالى

mysite.com/controller/view

مثلا عن ارسال

mysite.com/hello/index

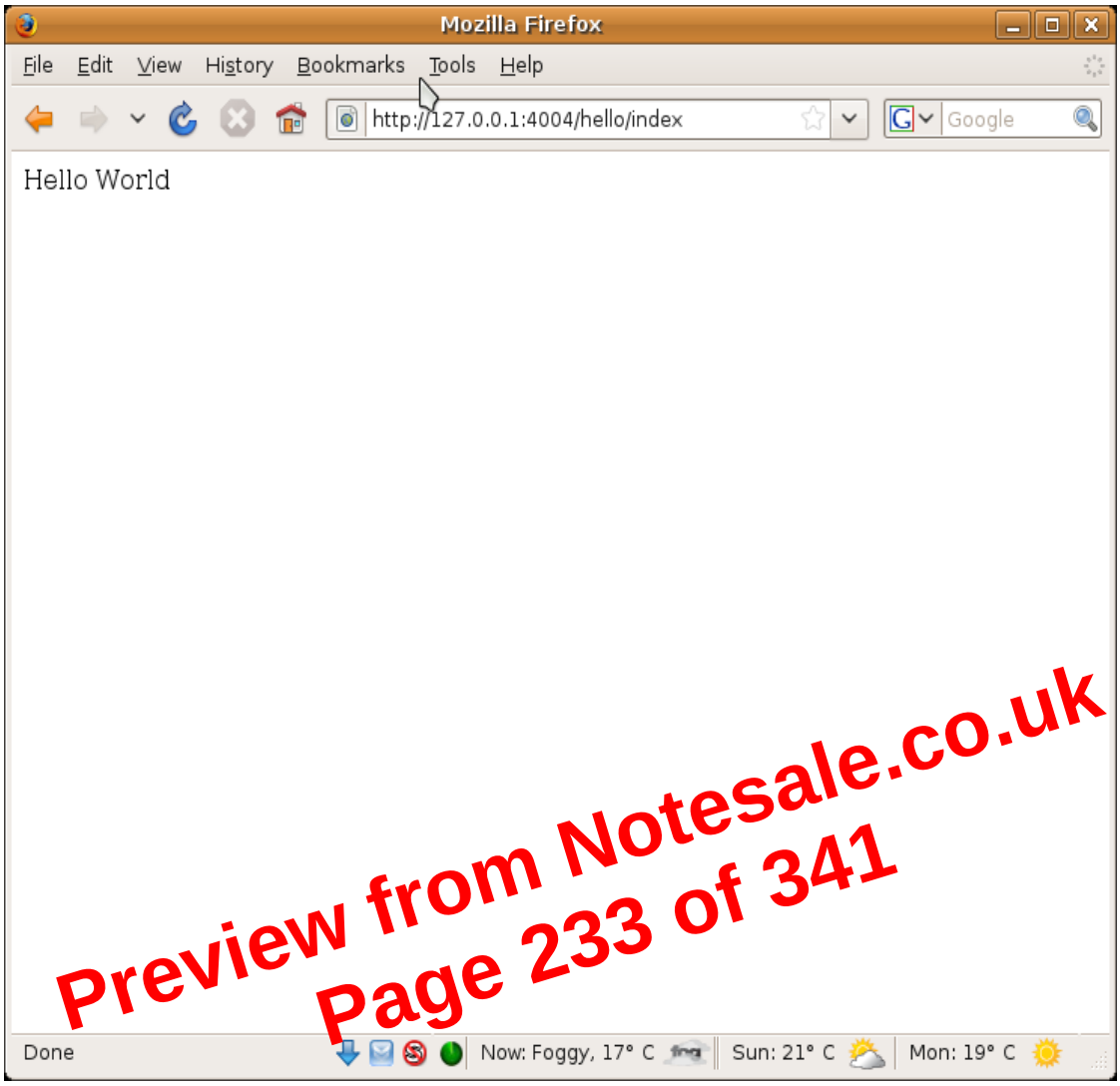
فيتم استدعاء الكنترولر (المقسم) hello ومنه يتم اختيار ال action المناسبة

قم بتشغيل السرفر

```
striky@striky-desktop:~/workspace/pytut/src/nettut/helloworld$ paster serve --reload development.ini
```

ملف ال development.ini يحوى معلومات عن البيئة كالهوست والبورت ومتغيرات التطبيق الخ الخ

Preview from Notesale.co.uk
Page 231 of 341



Preview from Notesale.co.uk
Page 233 of 341

```
Moving Extremes-1.1-py2.5.egg to /usr/lib/python2.5/site-packages
Adding Extremes 1.1 to easy-install.pth file

Installed /usr/lib/python2.5/site-packages/Extremes-1.1-py2.5.egg
Searching for PyProtocols>=1.0a0dev-r2302
Reading http://pypi.python.org/simple/PyProtocols/
Reading http://peak.telecommunity.com/PyProtocols.html
Reading http://peak.telecommunity.com/dist/
Best match: PyProtocols 1.0a0dev-r2302
Downloading http://files.turbogears.org/eggs/PyProtocols-1.0a0dev_r2302-py2.5-linux-i686.egg
Processing PyProtocols-1.0a0dev_r2302-py2.5-linux-i686.egg
Moving PyProtocols-1.0a0dev_r2302-py2.5-linux-i686.egg to /usr/lib/python2.5/site-packages
Adding PyProtocols 1.0a0dev-r2302 to easy-install.pth file

Installed /usr/lib/python2.5/site-packages/PyProtocols-1.0a0dev_r2302-py2.5-linux-i686.egg
Finished processing dependencies for TurboGears==1.0.8
```

Hello World: TG

tg-admin quickstart لإنشاء تطبيق سريع كل ماعليك هو تنفيذ

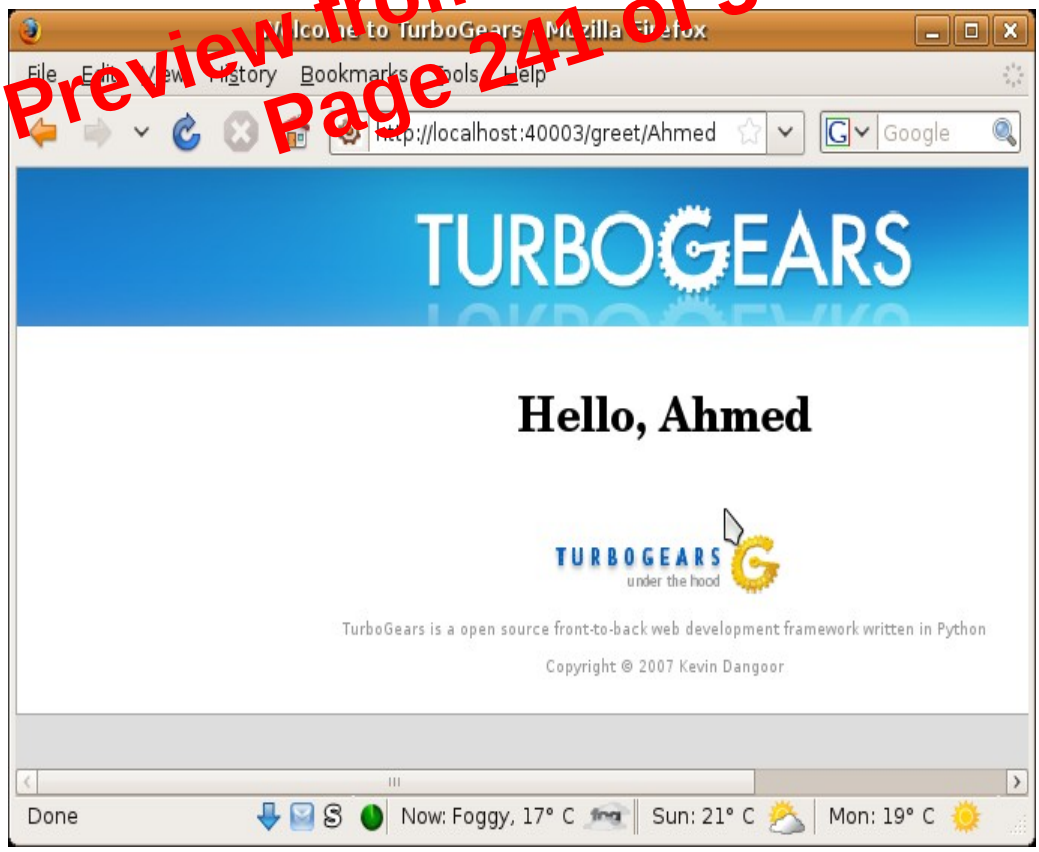
```
striky@striky-desktop:~/workspace/pytut/src/hellotut$ tg-admin quickstart
/usr/lib/python2.5/site-packages/CherryPy-2.3.0-py2.5.egg/cherrypy/lib/profiler.py:54: UserWarning: Your
installation of Python doesn't have a profile module. If you're on Debian, you can apt-get python2.4-profiler
from non-free in a separate step. See http://www.cherrypy.org/wiki/ProfilingOnDebian for details.
  warnings.warn(msg)
/usr/lib/python2.5/site-packages/RuleDispatch-0.5a1.dev_r2506-py2.5-linux-i686.egg/dispatch/__init__.py:98:
Warning: 'as' will become a reserved keyword in Python 2.6
/usr/lib/python2.5/site-packages/RuleDispatch-0.5a1.dev_r2506-py2.5-linux-
i686.egg/dispatch/predicates.py:239: Warning: 'as' will become a reserved keyword in Python 2.6
/usr/lib/python2.5/site-packages/RuleDispatch-0.5a1.dev_r2506-py2.5-linux-
i686.egg/dispatch/predicates.py:263: Warning: 'as' will become a reserved keyword in Python 2.6
/usr/lib/python2.5/site-packages/RuleDispatch-0.5a1.dev_r2506-py2.5-linux-
i686.egg/dispatch/predicates.py:281: Warning: 'as' will become a reserved keyword in Python 2.6
Enter project name: hello
Enter package name [hello]: hello
Do you need Identity (usernames/passwords) in this project? [no]
Selected and implied templates:
  TurboGears#tgbase      tg base template
  TurboGears#turbogears  web framework

Variables:
  egg:              hello
  elixir:           False
  identity:         none
  package:          hello
```

Preview from Notesale.co.uk
Page 235 of 341

عند الإستدعاء بدون معاملات

Preview from Notesale.co.uk
Page 241 of 341

بإستخدام معاملات



بعد انهاء اعلام بايثون عن الدوال اللى محتاجينها نقوم بتقديم الدوال البايثونية السلسة فى الإستخدام

```
def guess(filepath, flags):
        mc=magic_open(flags)
        magic_load(mc, None)
        res=magic_file(mc, filepath)
        magic_close(mc)
        return res
```

المشروع كامل تجده على
http://bitbucket.org/xmonader/libmagicbindings/
بإسم pymagic وتجد فيها ايضا مكتبات اضافية لسى شارب وباسكال لإستخدام libmagic

لمثال اكبر ومتعمق فى ctypes تقدر تطلع على binding لمكتبة PCRE على
http://bitbucket.org/xmonader/pypcre/

Preview from Notesale.co.uk
Page 256 of 341

```
Visitors:
        Ahmed
>>> someparty("Ahmed", "Mido")
Visitors:
        Ahmed
        Mido
>>> p=partial(someparty, "Ahmed")
>>> p("Mido")
Visitors:
        Ahmed
        Mido
>>> p
<functools.partial object at 0xb79d8d74>
>>> p("Mido", "Youssef")
Visitors:
        Ahmed
        Mido
        Youssef
```

Preview from Notesale.co.uk
Page 260 of 341

احد الإستخدامات العملية لها فى التعامل مع PyQt4 كالتالى مثلا -لأنك ستواجه صعوبة فى تمرير المتغيرات مع connect لل slot

```
self.connect(btn, SIGNAL("clicked()"),
        partial(self.onClicked, ARG))
```

طبعا تقدر تستخدم lambda بنفس الكيفية!

هنا مثلا نقوم بربط الإشارة clicked للزر button2 ب slot -تغلف self.anyButton اللتى تأخذ معامل واحد اللتى يتم تنفيذها عند محاولة استدعاء ال slot- هكذا قد تغلبنا على العملية اليس كذلك ؟

cookie => قيمة مميزة تستخدم لربط حدثين معا مثل IN_MOVED_FROM و IN_MOVED_TO
لأن كل منهم بيمثل نصف عملية النقل وهما الإتنين بيعبرو عن العملية. وهنا استخدمنا ال dict المسمى mv_cookies للربط بينهم

المثال كامل

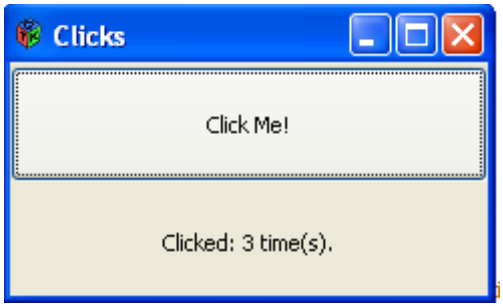
```
 import os
from pyinotify import *

wm=WatchManager()
mask=IN_DELETE | IN_CREATE | IN_MOVED_FROM | IN_MOVED_TO #to watch..

#no need..  use .pathname instead..
def fullname(event):
  return os.path.join(event.path, event.name)

class PDir(ProcessEvent):

  def my_init(self, **kwargs):
    self.mv_cookies={} # {cookie: src}

  def process_IN_DELETE(self, event):
    print "Remove: %s"%event.pathname

  def process_IN_CREATE(self, event):
    print "Create: %s"%event.pathname


  def process_IN_MOVED_FROM(self, event):
    print "Moving FROM: %s"%event.pathname
    self.mv_cookies[event.cookie]=event.pathname

  def process_IN_MOVED_TO(self, event):
    print "Moving TO: %s"%event.pathname
    print " WAS MOVED FROM: ", self.mv_cookies[event.cookie] #same unique number


#notifier
p=PDir()
notifier=Notifier(wm, p)

#add a watch..
watch=wm.add_watch("/home/striky/Desktop", mask, rec=True)

notifier.loop()
```

Preview from Notesale.co.uk
Page 268 of 341

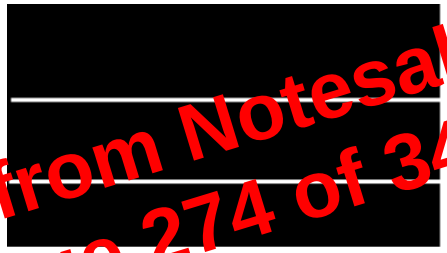
دا كان المختصر المفيد للمزيد حول pyinotify تقدر تراجع
http://seb.dbzteam.org/swp/pages/pyinotify.html

Preview from Notesale.co.uk
Page 269 of 341

Clicks



لاحظ ان النافذة متقسمة لجزئين راسي اول جزء فيه button والتانى فيه label بيتكتب عليه عدد مرات الضغط على ال button
احنا ممكن نكتبها بالطريقة المعتادة وممكن نكتبها بطريقة كتير افضل بإستخدام ال OOP
تمام الأول عشان نخليهم متقسمين فى شكل معين افقى او رأسى بنستخدم Hbox (اختصار ل Horizontal Box) او Vbox (اختصار ل Vertical Box) تمام ؟ قشطة



Preview from Notesale.co.uk
Page 274 of 341

ال vertical box بياخد ال widgets او بوكسز تانية فى صورة rows صفوف -صندوق رأسى-

ال horizontal box بياخد ال widgets او البوكسز التانى فى صورة Columns عواميد -صندوق افقى-

لإنشاء box بتبدأ اولا ب homogeneous ودى معناها هل كل الأجزء متساوية فى العرض او الطول ليها ترجمة بإسم متجانسة اعتقد مناسبة ؟ والمعامل التانى لتحديد عرض الفاصل

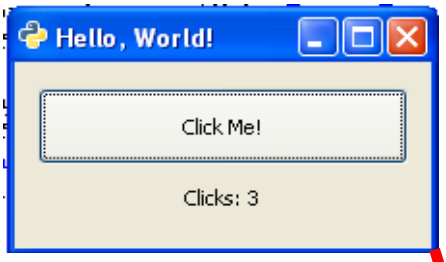
```
vbox=gtk.VBox(False, 2)
```

pack_start(child, expand, fill, padding)
لوضع ال widget من الشمال لليمين او من فوق لتحت "صورة فطرية!"
pack_end(child, expand, fill, padding)
لوضع ال widget من اليمين للشمال او من تحت لفوق وهى موجود لل Hbox وال Vbox

xpadding: الهامش من ناحية ال x
ypadding: الهامش من ناحية ال y
لو انت مرضى مع الخيارات الأساسية تقدر تستخدم attach_defaults
```
attach_defaults( child, left_attach, right_attach, top_attach,
bottom_attach )
```

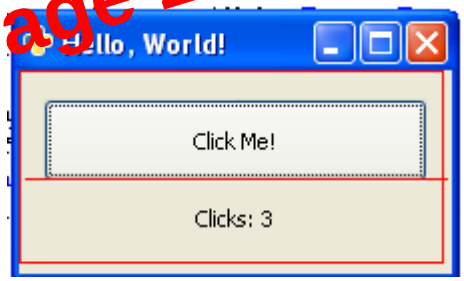
ودى هتخليك تدخل ال left, right, top, bottom attach فقط والباقى هيكون بالإفتراضى لل x,y options
gtk.FILL | gtk.EXPAND
وال x,y padding هيكون 0
فين ال fixed ؟ موجود fixed لعرض ال ويدجتس بتحديد ال مكان على الفورم ولكن "استخدام السابق افضل من حيث حماية طريقة وضعك لل ويدجتس من حيث التمدد والإنكماش وكدا"

تعالى نعمل مثال clicks بصورة واضحة



Preview from Notesale.co.uk
Page 276 of 341

التصميم



المستطيل الأحمر الكبير عبارة عن Vbox
وجواه صفين الصف الأول فيه button والتانى فيه label
الأول عندنا متغير __clicks بيعبر عن عدد الضغطات

```
class Window(gtk.Window):

    def __init__(self):
        super(Window, self).__init__(gtk.WINDOW_TOPLEVEL)
        self.__init_comp()
        self.__clicks=0
```

لتحديد ال fraction وهى الكم المنتهى

.set_orientation(**orientation**)

لتحديد اتجاه ملء الزيادة

```
  PROGRESS_LEFT_TO_RIGHT من اليسار لليمين
  PROGRESS_RIGHT_TO_LEFT من اليمين لليسار
  PROGRESS_BOTTOM_TO_TOP من اسفل لأعلى
  PROGRESS_TOP_TO_BOTTOM من اعلى لأسفل
```

.get_text()

للحصول على النص الظاهر على ال progressbar

.set_text(to)

تحديد النص الظاهر على ال progressbar إلى to

.pulse()

لتشير حدوث تغيير على ال progressbar

```
def progress_timeout(pbobj):

    if pbobj.activity_check.get_active():
        pbobj.pbar.pulse()
    else:
        # Calculate the value of the progress bar using the
        # value range set in the adjustment object
        new_val = pbobj.pbar.get_fraction() + 0.01
        if new_val > 1.0:
            new_val = 0.0
        # Set the new value
        pbobj.pbar.set_fraction(new_val)

    # As this is a timeout function, return TRUE so that it
    # continues to get called
    return True
```

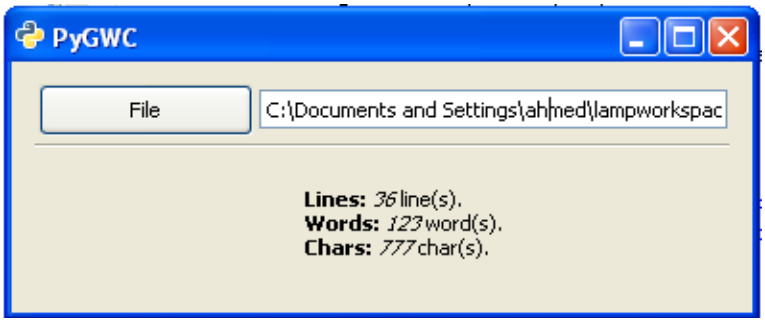
Preview from Notesale.co.uk
Page 295 of 341

.activity_check.get_active()

هل progressobject نشط اولأ

PyGWC

مثال لعد الأسطر والأحرف والكلمات



الأول نكتب الخدمة

```
class WordCounter(object):

    def __init__(self):
        pass

    def set_file(self, p):
        self.filepath=p
        fileobj=file(p, "r")
        self.txt=fileobj.read()
        fileobj.close()

    def get_nwords(self):
        return self.txt.count(" ")+1

    def get_nlines(self):
        return self.txt.count("\n")

    def get_nchars(self, count_spaces=True):
        if count_spaces:
            return len(self.txt)
        else:
            return len(self.txt)-self.get_nwords()

    def get_info_as_markup(self, spaces=True):
        s="""
        <b>Lines:</b> <i>%s</i> line(s).
        <b>Words:</b> <i>%s</i> word(s).
        <b>Chars:</b> <i>%s</i> char(s).
        """%(self.get_nlines(), self.get_nwords(),
self.get_nchars(spaces))
        return s
```

Preview from Notesale.co.uk
Page 297 of 341

التى عرفناها كالتالى مثلا

```
    def _on_item_activated(self, w, action):
        print "Calling %s"%action
```

7- اخيرا ضم العنصر اللذى يحوى القائمة الى شريط القوائم

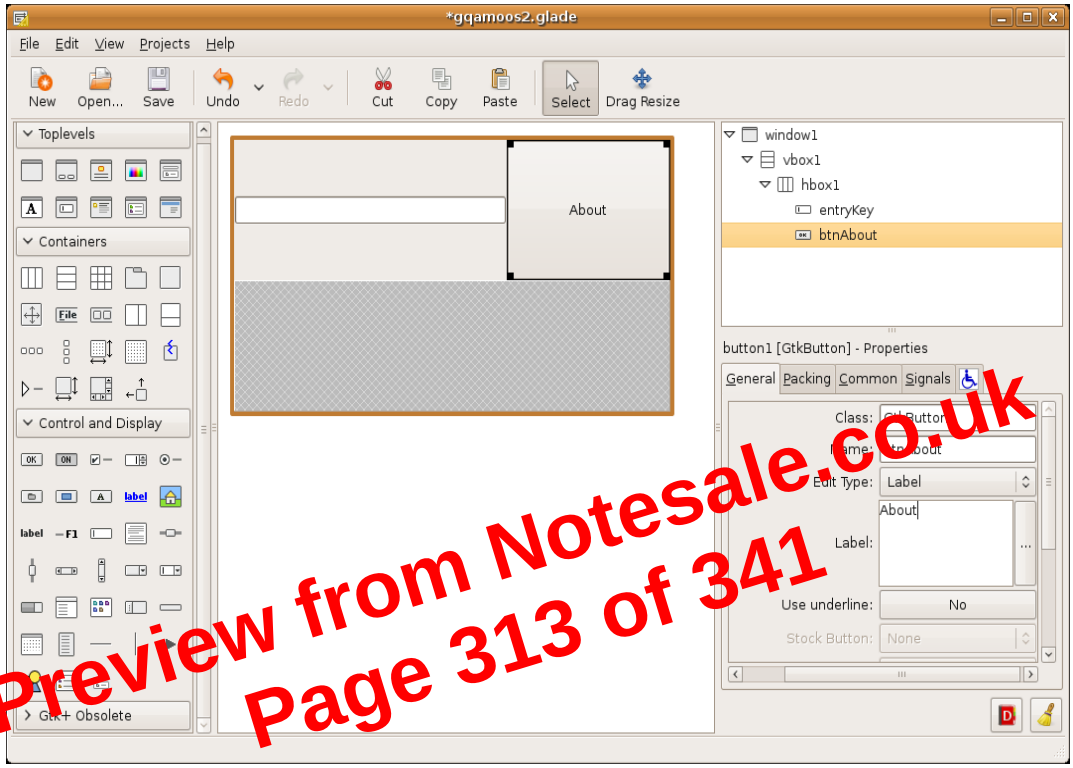
```
        mbar.append(file_item)
```

يوجد طريقة اسهل وهى بإستخدام ال UIManager توفر عليك الكثير من الكتابة

Preview from Notesale.co.uk
Page 306 of 341

من قسم ال General بأسفل اليمين قم بتغيير اسم الويدجت ل entryKey

قم بإضافة زر Button من اللوحة على يمين ال Entry



Preview from Notesale.co.uk
Page 313 of 341

```
self.inpentry=Entry(self, textvariable=self._inputfile)
```

لاحظ ربطنا لل textvariable بالمتغير self._inputfile بحيث ان يكون التغيير ديناميكى

لإنشاء ال Listbox

```
self.lbplugins=Listbox(self, height=5, selectmode=BROWSE, relief=SUNKEN)
```

نحددها على الشبكة (الصف والعمود. الخ)

```
self.lbplugins.grid(row=2, column=1, columnspan=3, sticky=E+S)
```

نضيف العناصر لها بإستخدام الطريقة insert وتأخذ اول معامل المركز الذى سيتم الإدخال عنده والثانى هو القيمة المدخلة

```
if len(getplugins())>=1:
    for plugin in getplugins():
        self.lbplugins.insert(0, plugin)
```

نحدد الإختيار الإفتراضى بإستخدام الطريقة select_set اللتى تأخذ معامل قيمته المركز

```
self.lbplugins.select_set(0)
```

تأخذ عدة معاملات مثل الإرتفاع ونوع الإختيار ولاحظ ايضا relief (اللتى جعلنا قيمتها SUNKEN)لجعلها غائصة

ال callbacks للزرين input, output

```
def _getinputfile(self, *args):

    self._inputfile.set(tfd.askopenfilename())
    #self.inpentry.delete(0, END)
    #self.inpentry.insert(0, self._inputfile)


def _getoutputfile(self, *args):

    self._outputfile.set(tfd.asksaveasfilename())
    #self.outentry.delete(0, END)
    #self.outentry.insert(0, self._outputfile)
```

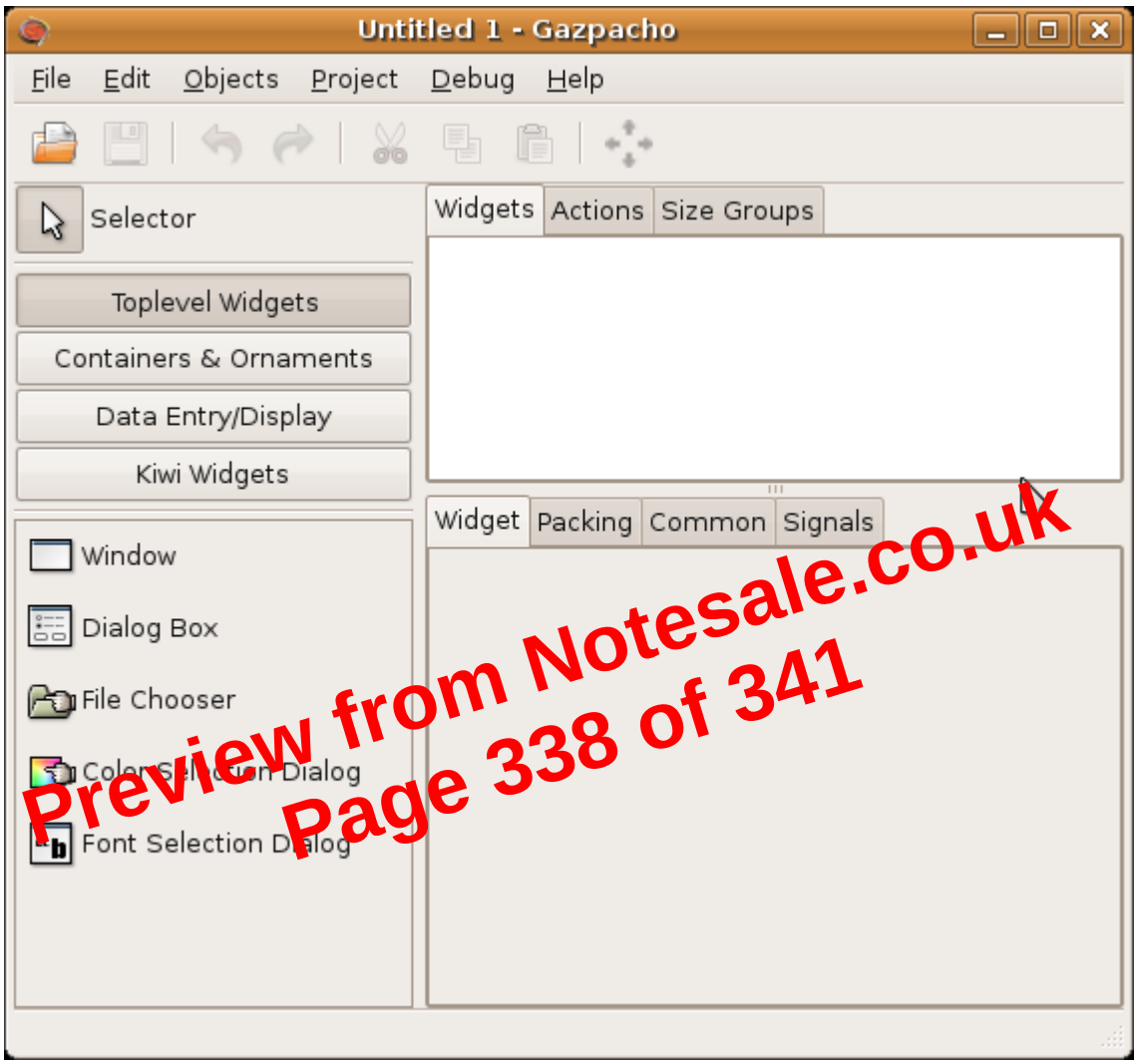
تقدم لنا الوحدة tkFileDialog الدوال اللازمة للتعامل مع الملفات (اختيار ملف مثلا)
نقوم بإستخدام tfd.askopenfilename() للحصول على مسار الملف المطلوب فتحه ونخزن قيمته بإستخدام الطريقة set الخاصة بال StringVar object
وبنفس الكيفية نحصل على مسار ملف الحفظ بإستخدام tfd.asksaveasfilename()

تقدم لنا الوحدة tkMessageBox الدوال اللازمة للتعامل مع الرسائل مثل showerror, showinfo,.. etc

Preview from Notesale.co.uk
Page 324 of 341

Gazpacho
http://gazpacho.sicem.biz/

الجيل الجديد من glade مكتوب بيايثون و gtk



Preview from Notesale.co.uk
Page 338 of 341