#أتعلم_بايثون (الدرس الأول)

الأفكار

١) تحميل وتثبيت بيئة العمل على البايثون
٢) شرح التعامل مع البرنامج
٣) كتابة أول برنامج
٤) التعامل مع دوال الإدخال والاخراج
٥) التعامل مع المتغيرات والتحويل بينها

تثبيت بيئة البايثون

افتح الموقع الرسمى للبايثون وحمل أى اصدار من المتوفرين ع الموقع
/https://www.python.org/downloads

من الإصدارات المتوفره إصدار ٣.٤.٣ والاصدار الأقدم ٢.٧ وليك انك تتسأل ايه الفرق بينهم : D

بس الموقع وال documentation الرسميه مجاوبنلك عن الفرق مابينهم بس اعرف ان اللغه هى اللغه وبس فيه فروق بسيطه فى الـ syntaxفى بعض الدوال

#EX

عشان تطبع جمله فى بايثون ٢.٧ كنت بتكتب :D

```
print "Hello Python"
```

الكلام ده لوعملته فى بايثون ٣.٤ هيديك syntax error عشان استخدام الداله بقى كده

```
Print ("Hello Pytho")
```

يا دوب حطلك أقواس ^_^ ..

ع العموم الفروق مابين 2.x و 3.x مش كتير .. لو كنت مبرمج سابق على ٢.٧ يهمك انك تعرفه عشان تقدر تشتغل على ٣.٤ لو بدايتك مع ٣.٤ ملكش دعوه بالإصدارات السابقه ^_^

#هنشتغل_بأيه ؟ ☺

تسأؤل جميل بس خايف اوضح ان اغلب الدورات والشروحات ع النت وخصوصاً كود اكاديمى متناولين فى الشرح بايثون ٢.٧

ولهذا السبب وعشان كده هشتغل على بايثون ٣.٤ :D ☺

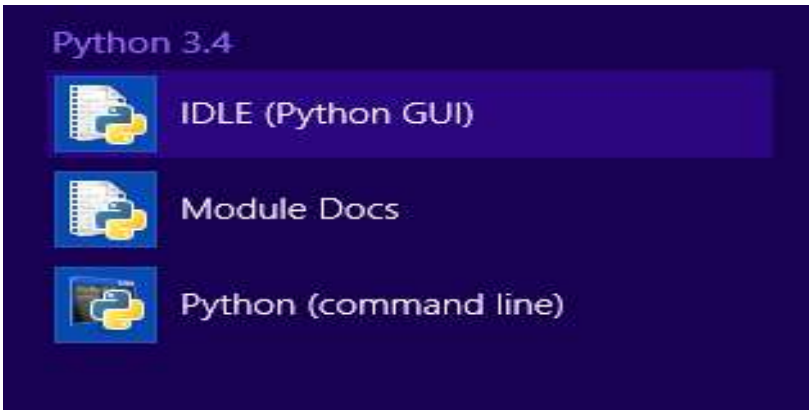
نرجع للتحميل وفرضنا انك استقريت على بايثون ٢.٧ هتدوس على رابط التحميل وهتدخل للصفحه اللى بعدها واخر الصفحه عتلاقى الهيدر files واللى تحتيه هتختار هتحمل بايثون لأنهى منصة تشغيل ... لو من مستخدمين ويندوز

Files

Version	Operating System
Gzipped source tarball	Source release
XZ compressed source tarball	Source release
Mac OS X 32-bit i386/PPC installer	Mac OS X
Mac OS X 64-bit/32-bit installer	Mac OS X
Windows debug information files	Windows
Windows debug information files for 64-bit binaries	Windows
Windows help file	Windows
Windows x86-64 MSI installer	Windows
Windows x86 MSI installer	Windows

أختار Windows x86-64 MSI installer وحمل بيئة البايثون ^_^

بعد ما تثبت البرنامج افتح قائمة أبدا وابحث عن python3.4 وافتح IDLE



IDLE : ده الانتربريتر interpreter الخاص بلغة بايثون ...

قولنا ف الموضوع الاول ان بايثون interpreted language يعنى يتم تحويل الكود الكود للغة الاله اثناء التشغيل on runtime على خلاف لغة السى ..

✓ مميزات IDLE

من الوثيقة الرسمية للغة .. بيقولك ان البرنامج ده برنامج بايثون خالص : D معمول بمكتبة الـ gui الرسيمه الخاصه باللغه اسمها tkinter ومن مميزاته انه بيقدملك code editor(المكان اللى بتكتب فيه الكود) لأكتر من مشروع ف نفس الوقت ده غير انه بيشتغل على اكتر من منصة تشغيل (نظام تشغيل)

على كل حال الميزه الاساسيه للـ IDLE ده :

بيقدملك واجهه رسومية للتعامل مع برامج البايثون

(بعيداً عن شاشة الكونسول الكئيبه ^_^

#خلى_بالك

IDE : integrated Development Environment

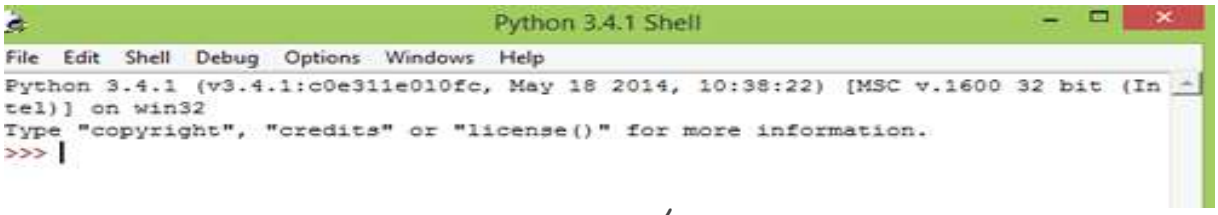
يعنى بيئة التطوير المتكامله .. اللى بتوفرلك code editor محرر نصى تكتب فيه كود وانتربريتر interpreter يترجم الكود و debugger يطلعلك أخطاء البرنامج ما الى ذلك ..

25.5. IDLE

IDLE is the Python IDE built with the `tkinter` GUI toolkit.

IDLE has the following features:

- coded in 100% pure Python, using the `tkinter` GUI toolkit
- cross-platform: works on Windows, Unix, and Mac OS X
- multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, and many other features
- Python shell window (a.k.a. interactive interpreter)
- debugger (not complete, but you can set breakpoints, view and step)



عشان منروحش بعيد اللى ظاهر قدامك ده بايثون idle

المؤشر >>> ده بيمثلك شاشة الاوامر التفاعليه اللى قولنا دى من مميزات البايثون انك تقدر تكتب الكود وتتعامل مع البرنامج اثناء التشغيل

^_^

اللى يهمنا فى القوائم

✓ File>>New File

ودى من خلالها بتفتح مشروع جديد اختصارتها

CTRLE+N

✓ File>> OPEN

ودى من خلالها بتفتح مشروع قديم : D كنت حافظه قبل كده

CTRLE+O

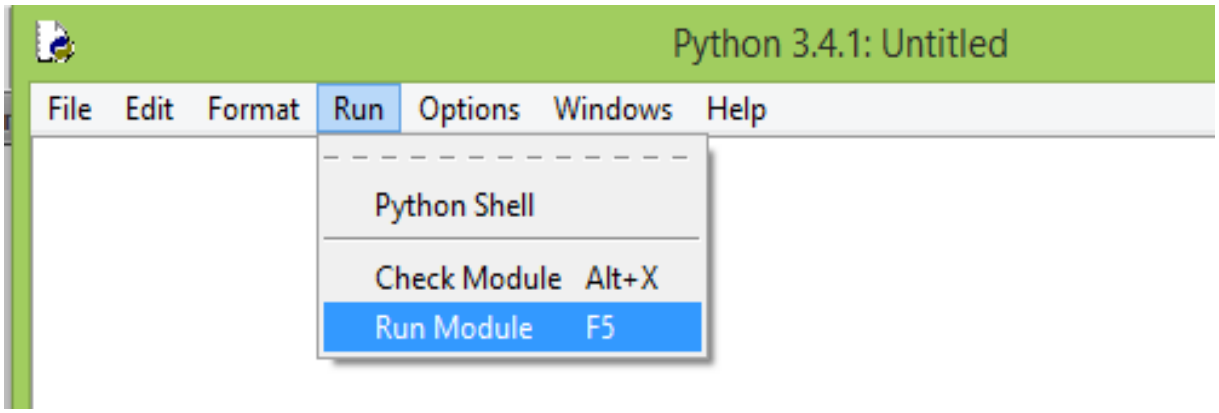
✓ Help>>Python Docs

ودى بتفتحلك الوثيقه الرسميه للبايثون اللى قرفتكم بيها وذكرتها اكتر من مره ^_^ اختصارتها F1

شايفين طبعاً انها بتقدم كل حاجه عن اللغه .. ايه الجديد فى الإصدار ده والفرق بينه وبين اللى قبله وقبل اللى قبله :D واهم حاجه Tutorial واللى هتلاقى فيها شرح كامل للغة ...

#مشروع_جديد

دوس CTRLE+N او من قائمة File اختار NEW



ظهرتلك صفحة كتابة الكود ☺ وعشان اكمل باقى القوائم اللى كلمتكم عنهم فوق ^_^

✓ Run>> Run Module

ودى من خلالها بتشغل المشروع اللى ان شاء الله هتكتبه

اختصارتها ع الكيبورد

F5

✓ Run>>Check Module

ف نفس القائمه دى بتشوفلك الاخطاء اللى ف الموديول اللى كتبته ولا لا ^_^

✓أول مشروع بايثون

الواحد قرى قد ما قرى ف كتب لغات البرمجه ويصادفنا مشروع hello world وازاى تبطع جمله على شاشة الكونسول

قبل ما اكتب الكود واشرح الكود بعدين كالمعتاد : D

لازم نرحب بأول داله معانا print()

print#

دالة print من الدوال الـ built in ف اللغه يعنى بتستدعيها مباشرة مش محتاج تنادى على مكتبات او تضيف موديولات زى ما كنت بتعمل ع السى

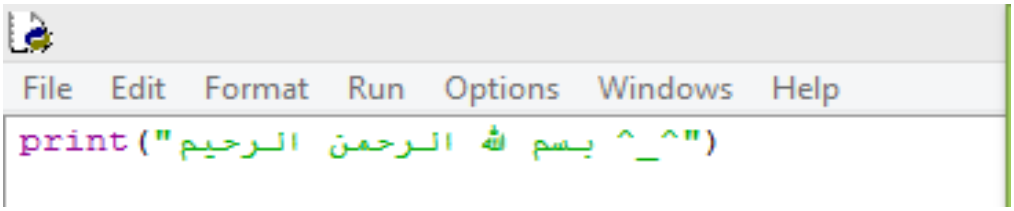
من غير رغى كتير افتح الموديول الجديد اللى لسه فتحتناه واكتب

```
print("بسم الله الرحمن الرحيم ^_^")
```

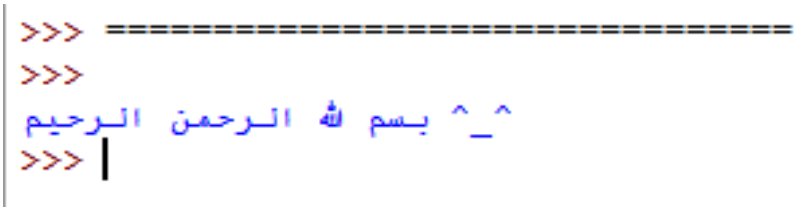
لو كنت من مستخدمين بايثون ٢.٧ فميش داعى انك تحط اقواس كنت هتكتب

```
print "بسم الله الرحمن الرحيم ^_^"
```

يبقى شكل صفحة الكود هيكون كده



وبعد ما تدوس run او F5 هيظهر الخرج بتاعك فى شاشة الاوامر التفاعليه



#أستخدامها_أزاى

print(حاجه)

يبقى دالة print تطلع خرج على شاشة المستخدم اما يكون ..

نص بين علامتين تنصيص ""

او نص بين single quote ' '

او رقم من غير "" ..

او متغير وهنشوفه قدام ..

أفتح صفحة الكود السابقه واكتب

```
print("نص")
print('نص')
print(1)
```

الخرج هيكون كده ..

```
>>>
نص
نص
1
>>>
```

تلاحظ انى بكتب النص باللغه العربيه ☺ وده عشان اوضحلك بس ان بايثون IDLE بيدعم العربى على عكس شاشة الكونسول ..

✓الانتربريتر التفاعلى

انت من مميزات البايثون انى فيه حاجه اسمها interactive interpreter وقولت ده بيمكنك انك تتعامل مع الكود اثناء تشغيل البرنامج ازاى ؟

تمام .. اقفل الموديول او المشروع اللى كنا شغالين عليه دلوقتى وخليك ف الصفحه الاولى للـ idle

```
Python 3.4.1 (v3.4.1:c0e311e010fc, Ma
tel)] on win32
Type "copyright", "credits" or "licen
>>> ================================
>>>
نص
نص
1
>>>
```

هتلاقى الـ cursor انه مستعد للكتابه وانك تتكب كود ينفذه الانتربريتر مباشرة :D

اكتب كده جلمة تطبع اى حاجه ..

```
print("Ezay")
```

```
>>> print("Ezay")
Ezay
>>>
```

ممكن تكون دلوقتى مش عارف ايه فايدة او قيمة انك تتفاعل مع الـانتربريتر مباشرة :D

بس تخيل انك مشغل مكتبة السيريال او بتتعامل مع اردوينو مباشرة من البايثون هيكون الكود الشغل عملى ازاى ..
وغير كده هيفوفر عليك وقت انك تفتحت مشروع جديد وتعمل save قبل ماتعمل run وبردك هيفتحلك نفس الانتربريتر التفاعل :D

Python Syntax

هحاول أتكلم عن بعض القواعد المهمه واللى هتتعامل معاها فى كتابة الكود وهتيجى مع الزمن بس بشىء مجمل والتفصيل هتعرفه بعدين :D

✓ التعليق comment

معروف ان الكومنت سطر او فقره بكتبها ف الكود وبيتجاهلها الانتربريتر ومش بيعتبرها تبع الكود ..

طيب بعملها ليه ؟

لتوضيح جمله او حته ف الكود او سطر برمجى مش اكتر ..

فى الـ C كنا بنعمل التعليق بـ٢ سلاش // ولو فاكر التعليق متعدد الاسطر
/* تعليق */

#ف_البايثون

- لو سطر مفرد نسبقه بعلامة هاش (شباك) #

#Just Comment ^_^

- لو عدة سطور بنحطهم بين ٣ double quote """تعليق """

شوف الكود والخرج فى البرنامج التالى

```
#تعليق سطر واحد :V

"""تعليق متعدد الأسطر
الانتربريتر مش هيشوفني ^_^
"""
print("learn Python ")
```

```
Type "copyrigh
>>> ==========
>>>
learn Python
>>>
```

No Semicolon ; ✓

طبعاً من المعتاد فى لغة السى ومشتقاتها ان نهاية اى سطر برمجى بينتهى بـ ; سيمى كولون عشان الكومبايلر او الإنتربريتر يعرف ان السطر ده خلص

ولو لاحظت الكود اللى فات بتاع جملة print انه منتهاش بيها ..

ده يعرفك ان السطور البرمجيه ف البايثون مش بنختمها بسيمى كولون وده زى لغة الفيجوال بيسيك

وعشان ابقى صادق احنا ممكن نستخدم السيمى كولون ف اكتر من موضع منها

اننا نحط جملتين 2 statement فى سطر واحد

#EX

عاوز تطبع جملتين بدالة print اما تعمل كده

```
print("first")
print("second")
```

لو عاوز تخلي الجملتين فى سطر واحد

```
print("first") ; print("second")
```

- No Braces مفيش اقواس { }

من الاختلافات التانيه اللى بتتفرد بيها لغة البايثون عن السى ان البلوكات blocks اللى بتحتوى على اكتر من statement لا يتم الجمع بينهم بكيرلى بركاتس { }

زى ما بيحصل فى لغة السى فى جمل الشرط والدوال والكلاسات .. مثال على البلوكات فى لغات البرمجه المختلفه ..

- مثلاً لو عندنا فى لغة السى جملة الشرط if لو شرطها تحقق هتنفذ السطر اللى تحتها لو فيه بلوك تنفذه كله والـ syntax كالأتى

```
If(condition){
ده بلوك
نفذ اللى هنا لو الشرط اتحقق
}
```

- فى لغة الفيجوال مش بتستخدم {} اقواس وتعمل بلوكات زى السى بس بتستخدم كلمة end فى نهاية اى تجمع من السطور البرمجيه (بلوك يعنى ^_^)

```
If (condition) then
نفذ السطور دى
End if
```

- طيب لغة البايثون بتعمل ايه ؟

استحملنى : D بس ده مثل رخم لتوضيح فكرة البلوكات فى لغات البرمجه وانت مش مطالب غير باللغه اللى انت عارفها ☺

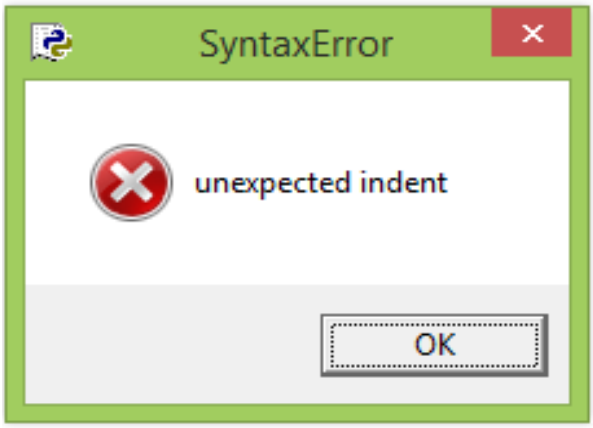
بايثون لغة حساسه لموضع السطر :O

يعنى كل السطور اللى ليهم نفس المحاذاه دول تبع بلوك واحد :D

ارجع لأى مثال فى بتوع print ودوس مسطره (ادى مسافه) قبل كلمة print وبعدها دوس RUN

هتلاقى البرنامج اداك syntax error بحجة انت واخد المسافة اللى معلم لك عليها بالاحمر دى ليه ؟

```
print("learn Python ")
```



يعنى كل اللى تعرفه دلوقتى ان بايثون لغه حساسه لموضع السطر واى مجموعة سطور تحت بعض ليهم نفس المحاذاه بيكونو بلوك واحد وهنفهم اكتر لما ناخد جمل الشرط والتكرار وده شكل بلوك لجمل if فى البايثون

```
if True:
    print "Answer"
    print "True"
```

جملة if هتعرف السطور اللى هتنفذها منين ؟ ان ليهم نفس المحاذاه ..

المتغيرات

المتغير فى اى لغة برمجه .. المتغير ده مكان بتحجزه فى الذاكره ram عشان تخزن فيها داتا

وشايفنى بقول الرام ☺ يعنى التخزين بيكون اثناء فترة شغل البرنامج واثناء عمل الجهاز بس ..

#أنوع_البيانات_فى_البايثون

- Numbers : ده لو هتخزن قيمه عدديه رقم صحيح او عشرى

ومعروف انواع المتغيرات الرقميه

Integer متغير يخزن رقم صحيح بقيم + او سالب زى ١٠

float متغير يخزن رقم فيه كسور زى ١٠.٥

- Strings

لو عاوز تخزن نص فى الميمورى اما بيكون

Char حرف واحد

String مجموعة حروف

#أزاى_أعرف_متغير ؟

لو كنت من مبرمجين السى أكيد عندك علم عن طريقة تخزين متغير عددى فى الذاكره

اول حاجه تعرف نوعه وبعدين تحدد اسمه وتعمله assign وتديه قيمته

مثلاً فى لغة السى C عاوز اخزن قيمة ١٠ فى متغير اسمه x ونوعه عدد صحيح

```
int x=10;
```

ومتنساش السيمى كولون :D

#python

فى لغة البايثون الموضوع ابسط من كده بكتيره ^_^

مش معهم تقول للإنتربريتر نوع العدد او الداتا تايب اللى هتخزنه

أهم حاجه النيه : D انك تحط ف دماغك نوع الداتا تايب وتعمله assign ينعى تديه قيمته

#EX

عاوز تخزن رقم صحيح نوعه int واسمه x وقيمته ١٠

هتكب

```
x=10
```

انت محددتش نوعه بس المترجم هيفهم انه نوعه int

لو كتبت الكود التالى :

```
X1=10
X2=10.5
X3="Learn Python"
```

على كده انت عرفت متغير x1 نوعه صحيح وقيمته ١٠ الانتربريتر عرفه من قيمته

عرفت متغير x2 نوعه متغير عشرى وقيمته ١٠.٥ الانتربريتر عرفه من قيمته

عرفت متغير ٣X نوعه نص وقيمته ١٠ الانتربريتر عرفه بردك من قيمته ^_^

افتح الانتربريتر التفاعلى : D واكتب الكود اللى فى الصوره

```
>>> x1=10
>>> x2=10.5
>>> x3="python"
>>> print(x1,x2,x3)
10 10.5 python
>>>
```

وبعد ماتعرّف كل متغير وتديه قيمته دوس انتر عشان تعرّف اللى بعده

عشان تطبع قيمة المتغير قولنا هنستخدم دالة

print (حاجه)

وقولت بدرى ان الحاجه دى ممكن تكون نص بين "" او رقم او ممكن تكون متغير او اسماء عدة متغيرات مفصول بينهم بفصله

```
print(x1,x2,x3)
```

لو كنت عاوز تطبع كل متغير لوحده مافيش مشكله خالص انك تكتب

```
print(x1)
print(x2)
print(x3)
```

حلو جداً ^_^ دلوقتى لو عندى متغير وعاوز اعرفه بديه قيمه والانتربريتر بيعرف نوعه..

بس لو انت قولتلى انا مخزنلك متغير اسمه x1 وعاوزك تعرفلى نوعه : D
وانا مش هوريك قيمته :3

تقدر تعمل ده بسهوله بدالة type

```
type#
```

دالة بترجع نوع المتغير شكلها type(variable)

مثلاً ارجع للإنتربريتر التفاعلى بتاعنا : D واسأله عن نوع التلت متغيرات اللى عرفناهم بدرى بس ياريت متكونش قفلته عشان ميكونوش طاروا من الميمورى ^_^

```
>>> x1=10
>>> x2=10.5
>>> x3="python"
>>> print(x1,x2,x3)
10 10.5 python
>>> type(x1)
<class 'int'>
>>> type(x2)
<class 'float'>
>>> type(x3)
<class 'str'>
>>>
```

زى ما فى الصوره رجعلك نوع كل واحد منهم سواء intاو strاو floatو..

#التحويل_مابين_المتغيرات ..

دلوقتى لو عندك متغيرين نصيين قيمتهم ١٠ و ١٠

X1="10"

X2="10"

عاوز تعمل عليهم عمليه حسابيه .. جمع مثلاً .. نشوف الكود ..

```
>>> x1="10"
>>> x2="10"
>>> x1+x2
>>> print(x1+x2)
1010
>>>
```

اللى حصل ده دمج concat مش جمع ..

العمليات الحسابيه مش بتتم غير على المتغيرات العدديه

يبقى عشان تحول x1 ,x2 لمتغيرات عديه عندنا دوال

تحول من متغير نصى لمتغير صحيح او كسرى

int(variable

float(variable)

او من متغير عددي لمتغير نصى

str(variable)

يبقى عشان نجمع المتغيرين النصيين اللى فوق هنحولهم لأرقام وبعدين نعمل عليهم العمليه الحسابيه ..

```
>>> x1="10"
>>> x2="10"
>>> print(int(x1)+int(x2))
20
>>>
```

إدخال البيانات

عرفنا جملة الإخراج فى البايثون print

طيب دلوقتى لو المستخدم عاوز يدخل حاجه من الكيبورد ويعرضها ع الشاشه يبقى لازم يستخدم دالة الإخراج ..

#input

هى دالة input .. بترجع للمستخدم return اى حاجه يكتبها المستخدم على لوحة المفاتيح .. بس مدام بترجع : D لازم استقبل اللى بترجعه فى متغير واحفظه لحين اخراجه على الشاشه

لو كنت من مستخدمين السى ++ : D مستحيل انك تعمل cin>> المشابهه لجملة input هنا من غير متغير تدخل فيه اللى بيكتبه المستخدم

```
x=input("Write Any Thing ")
```

النص اللى بكتبه جوه اقواس input ده هيظهر للمستخدم جنب المؤشر اللى بيظهر ويختفى عشان تكتب وتدوس انتر ..

ممكن متكتبش حاجه بين الاقواس وهيظهرلك المؤشر لوحده من غير جمله تطلب انك تدخل حاجه ..

```
x=input()
```

```
>>> X=input("Write Any Thing : ")
Write Any Thing :
```

طيب انت بعد ما تكتب اللى عاوز تكتبه وتدوس enter هيروح فين ؟

السؤال ده ميتسألش : D اكيد فى المتغير x اللى نوعه string !

طيب يفرق معاك نوعه ؟

ايوه جداً .. لو عاوز تعمل عمليه حسابيه عليه او اى حاجه لازم تحول مابين المتغيرات زى ما قولت فوق :D

عاوز تسترجع او تعمل طباعه للقيمه اللى دخلتها

Print(x)

```
>>> X=input("Write Any Thing : ")
Write Any Thing : ٣: اتعلم بايثون
>>> print(X)
٣: اتعلم بايثون
>>>
```

لو شغال فى مشروع او موديول مش الانتربريتر التفاعلى هيكون الكود والخرج كده

```
X=input("Write Any Thing : ")
print(X)
```

```
>>>
Write Any Thing : ٣: اتعلم بايثون
٣: اتعلم بايثون
>>>
```

مثال الموضوع

ده المثال الاخير اللى بيلخصلك الموضوع كله ..

#الهدف : بيعملك ازاى تتعامل مع الادخال والاخراج والمتغيرات والتحويل مابينهم ..

#المطلوب : اعمل برنامج يجمعلك رقمين يدخلهم المستخدم ويطبع قيمتهم ع الشاشه ..

X1=input("دخل الرقم الاول : ")

X2=input("دخل الرقم التانى : ")

print(x1+x2)

```
x1=input("دخل الرقم الاول : ")
x2=input("دخل الرقم التانى : ")
print("Sum : ", x1+x2)

>>>
دخل الرقم الاول : 5
دخل الرقم التانى : 5
Sum :  55
>>>
```

ايه ده ؟ 5+5=55 ☹

شوفت انك وقعت فى خطأ حسابى كبير : D قولتلك فوق انى الداله دى بترجع متغير نصى وهنعرف بعد كده عشان تدمج نص مع نص بتستخدم +

يعنى يادوب وصلك الحرف 5جنب اخوه 5

#الحل

نحول قيمة متغيرين الدخل قبل مانطبعهم ..

انك تحول المتغير النصى لمتغير عددى سواء صحيح او كسرى.. لاحظ انى بحول الى int.. انت ربما تدخل قيمه فيها كسور تقدر تحول float

الكود الصحيح هيبقى كده ..

```
x1=input("دخل الرقم الاول : ")
x2=input("دخل الرقم الثانى : ")
print("Sum : ", int(x1)+int(x2))
```

```
>>>
دخل الرقم الاول : 5
دخل الرقم الثانى : 5
Sum :  10
>>>
```

ملخص

- يبقى فى الدرس ده عرفنا ازاى نحمل بيئة البايثون ونشغلها على الجهاز
- عرفنا ازاى نتعامل مع الـ IDLE ونشوف مميزاته
- عملنا اول مشروع بايثون
- عرفنا ازاى نطبع جمله ع الشاشه
- ازاى ندخل نص من الكيبورد
- ازاى نتعامل مع المتغيرات ونحول مابينهم ..

خاتمه

إنتهى الدرس الاول فى سلسلة #اتعلم_بايثون

تابع الدروس ع صفحتى

https://www.facebook.com/earthuino

الأكونت بتاعى ع الفيس

https://www.facebook.com/EngMa7moud3ly

لو شايف اى حاجه مش مزبوطه فى طريقة العرض او الشرح او اى حاجه : D يسعدنى جداً انك تبلغنى ☺

شكراً للمتابعه