

دفتر ملاحظات جوبيتر خيار مثالي للتعليم والتعلم

د. إياد حاتم* ، ظلال عطا هدهله**، كنانة عدنان سعيد***

* (هندسة الميكاترونك، جامعة تشرين)

iyadhatem@tishreen.edu.sy

** (هندسة الميكاترونك ، جامعة تشرين)

zihal.hadla@gmail.com

*** (هندسة الحاسبات والتحكم الآلي ، جامعة تشرين)

kenanasaeed11@gmail.com

الملخص

تعد البرمجة أداة أساسية في معظم التخصصات العلمية الحديثة وخاصة المجالات الهندسية وحتى الصناعية منها. وقد تزايدت في الآونة الأخيرة أعداد الراغبين بتعلم البرمجة بمختلف أشكالها ولغاتها لذلك ظهرت الحاجة لوجود أدوات تساعد على تعلم البرمجة وتسهيل فهمها على الطالب وبنفس الوقت تساعد المدرس على توصيل الأفكار بصورة واضحة ومن أبرز هذه الأدوات دفتر ملاحظات جوبيتر. وقد ذاع صيت دفتر ملاحظات جوبيتر مؤخرًا خصوصًا مع الانتشار الواسع لعلوم البيانات والذكاء الاصطناعي. نستعرض في هذه الورقة شرح تفصيلي لهذه الأداة بالإضافة إلى مجموعة من الأمثلة المتفرقة التي توضح إمكانياتها.

كلمات مفتاحية - دفتر جوبيتر - دفتر التفاعلي الرقمي - لغات البرمجة - أدوات التعليم

1. مقدمة

ولكن الاختلاف هو أنه في دفتر الملاحظات التفاعلي التقليدي ، يقوم الطالب بقص الأجزاء ولصقها في دفتر ملاحظات فعلي.

تعمل الدفاتر التفاعلية الرقمية على ترويج التعلم. عندما يرى الطلاب أن المحتوى والمهارات مهمة بالنسبة لهم "في الوقت الحالي وفي المستقبل" (الاحتياجات الفورية والمستقبلية) ، فإنهم يكونون أكثر استعدادًا وقدرة على الانخراط في تعلم محتوى ومهارات جديدة. التعليم هو عملية ذات اتجاهين. بالتأكيد يحتاج الطلاب

أصبحت الدفاتر التفاعلية الرقمية Digital Interactive Notebooks شائعة بشكل متزايد في جميع المجالات: القراءة والعلوم والرياضيات والتاريخ و الدراسات الاجتماعية وفنون اللغة والفن ودراسة اللغات الأجنبية والبرمجة. وتشبه الدفاتر التفاعلية الرقمية إلى حد كبير الدفاتر التفاعلية التقليدية، ما عدا أنها رقمية لاستخدام الأوراق. فيمكنها تغطية نفس المحتوى ونفس المهارات

إلى الإقلاع بشكل أبطأ ولكنها غنية بالميزات ومعقدة. يمكن لمحررات النصوص البسيطة عادةً تحرير التعليمات البرمجية فقط، ولا يمكنها تشغيل الرماز Code أما بالنسبة لبيئات التطوير المتكاملة فيمكنها تحرير التعليمات البرمجية وتشغيلها وتصحيح الأخطاء.

يوفر دفتر جوبيتر طريقة سريعة وبمبسطة للمهندسين والمبرمجين بعمل نموذج أولي للرمز ومشاركة التعليمات البرمجية بسرعة كما أنه يتيح للمبرمجين مشاركة حلول البرمجة مع أعضاء الفريق والمشرفين والعملاء.

يعد دفتر جوبيتر من أهم أدوات علماء البيانات اليوم حيث يقوم بإنجاز استكشاف البيانات ومعالجتها ضمن عدة خطوات ومراحل مما يسهل عملية تحليل البيانات وتنظيمها. أيضا يمكنه بناء مخططات عرض بصرية للبيانات بشكل أسهل وأسرع من محرر النصوص التقليدي أو بيئة التطوير المتكاملة مما يتيح سهولة أكبر في شرح النتائج لأصحاب المصلحة و الذي يعتبر غرض أساسي في عالم البيانات.

II. تنصيب دفتر جوبيتر

لتنصيب دفتر جوبيتر على نظام ويندوز 10 يمكننا استخدام بيئة أناكوندا Anaconda التي تحتوي على نسخة من دفتر ملاحظات جوبيتر وذلك من خلال:

<https://www.anaconda.com/products/distribution>

ثم كتابة الأمر التالي في نافذة أناكوندا:

[Jupyter-lab]

أو يمكن تنزيل دفتر جوبيتر لوحده على الحاسوب بدون الاستعانة ببيئة أناكوندا ولتقوم بذلك نتبع الخطوات التالية: **أولاً:** لنفترض أننا نريد استخدام دفتر ملاحظات جوبيتر من أجل البرمجة بلغة بايثون سنحتاج أولاً لتحميل حزمة خاصة بلغة بايثون من خلال الضغط على الرابط التالي <https://www.python.org> ثم من قائمة [Downloads] نقوم بتنزيل النسخة التي تلائم نظام

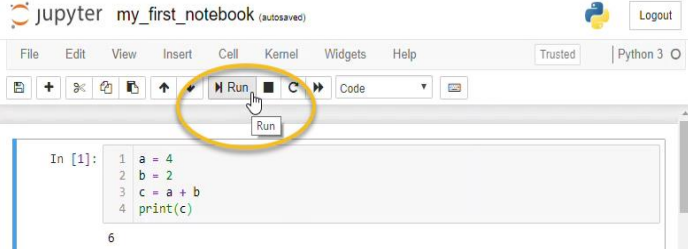
إلى مدخلات، لكنهم يقومون أيضًا بتصفية تلك المدخلات من خلال المعرفة والخبرات السابقة ويصنعون فهمهم الشخصي من تلك المدخلات. تزود الدفاتر التفاعلية الرقمية الطلاب بالاتصالات التي يحتاجونها بين العالم الخارجي للأفكار وعوالمهم الداخلية حول كيفية فهمهم لتلك الأفكار. عندما يمتلك الطلاب دروسهم التفاعلية في دفتر الملاحظات مع أهداف التعلم، فإنهم يتعلمون المحتوى والمهارات على مستوى أعمق ويحتفظون بمزيد من المعرفة.

توفر الدفاتر التفاعلية الرقمية الآليات التي يحتاجها المعلمون والطلاب لمعالجة المحتوى والمهارات الجديدة. الأمر كله يتعلق بالتعلم وليس التدريس. عندما يقوم الطلاب بإضافة أو إبراز الأفكار الرئيسية في ملاحظات المحاضرة، وتدوين التعليقات التوضيحية الهامشية على مقالات الدفاتر التفاعلية القصيرة، وتلخيص التعلم في المخططات الرسومية ثلاثية الأبعاد، فإنهم يقومون بمعالجة المعلومات والذي يساهم في تعزيزها حيث نعلم جميعاً مقدار التعلم المفقود عندما لا يتم تعزيزه على الفور. ويتم التدريب على استخدام المحتوى والمهارات في الدفاتر التفاعلية الرقمية مباشرة بعد المحاضرة التي توفر هذا التعزيز. مما يضع بصمة التعلم في الذاكرة طويلة المدى.

وبشكل مشابه، في عالم البرمجة وعلوم البيانات كان لاستخدام الدفاتر التفاعلية الرقمية تأثير كبير في رفع جودة التعلم حيث أصبح من السهل تقسيم المسائل لأقسام أصغر مما انعكس بشكل إيجابي على حلها، إضافة إلى تقديم وتوفير العديد من المخططات التي تعطي فهم أعمق للمسألة. هناك العديد من الدفاتر التفاعلية مثل أباتشي زيبلن Apache Zeppelin ، دفتر مود Mode Notebooks، غوغل كولاب Google Colab ، نيكست جورنال Nextjournal وغيرها، وسنتحدث في هذا المقالة عن دفاتر ملاحظات جوبيتر التفاعلية Jupyter Notebook الأكثر شهرة.

إن دفتر ملاحظات جوبيتر ليس مجرد محرر نص بسيط ولا بيئة تطوير متكاملة. بطريقة ما، فهو يحقق توازناً بين برامج تحرير النصوص البسيطة، والتي تتميز بالسرعة في الإقلاع والبساطة وسهولة التعديل عليها وبين بيئات التطوير المتكاملة التي تميل

يمكنك النقر فوق حقل الاسم وتغييره. لاحظ أن امتداد الملف [.ipynb] لا تتم طباعته في حقل اسم الملف، ولكن إذا نظرت في علامة التبويب الصفحة الرئيسية، فسترى أن الملف محفوظ بامتداد [.ipynb].



الشكل (1-1)

ضمن خلايا التعليمات البرمجية، يمكنك كتابة رماز اللغة البرمجية التي اخترتها، ثم تنفيذ الرماز ورؤية الناتج. ب. على سبيل المثال سنستخدم رماز بايثون هنا وللتنفيذ نضغط على الزر [Run] أو نكتب [Enter] + [Shift]. يؤدي الضغط على [Enter] عندما يكون المؤشر داخل خلية رماز إلى إحضار المؤشر إلى سطر جديد. كما هو موضح في الشكل (1-1).

بعد تنفيذ خلية التعليمات البرمجية، تظهر خلية خرج أسفلها كما هو موضح في الشكل (1-2)، وهي تحتوي على الخرج من التعليمات البرمجية أعلاه. لا ينتج كل رماز خرج، لذلك لا تنتج كل الخلايا البرمجية خلايا مخرجات. خلايا الإخراج. في حال كانت خلية الإخراج تنتج مخططات أو صوراً، فستظهر في خلايا الإخراج.

بعد تنفيذ خلية التعليمات البرمجية، تظهر خلية خرج أسفلها كما هو موضح في الشكل (1-2) وهي تحتوي على الخرج من التعليمات البرمجية أعلاه. لا ينتج كل رماز خرج، لذلك لا تنتج كل الخلايا البرمجية خلايا مخرجات. خلايا الإخراج. في حال

التشغيل المستخدم (إذا كان النظام ويندوز 10) ونضغط على [Windows]

ثانياً: من قائمة [Stable Releases] نختار النسخة التي تلائم المعالج المستخدم.

ثالثاً: نقوم بتشغيل الملف المراد تنزيله ولا ننسى التحقق من تحديد خيار [Add python to Path] وإلا فإنه سيعطي خطأ ثم نضغط على خيار [install now]

رابعاً: بعد الانتهاء من تثبيت حزمة بايثون يجب الآن تثبيت دفتر ملاحظات جوبيتر، نذهب أولاً إلى جهاز الحاسب [This PC] ثم إلى القرص [C] ثم إلى مجلد [Users] ملاحظة: نقوم بعمل عرض للعناصر المخفية من خلال تبويب [View] فيظهر لدينا مجلد اسمه [AppData] إذا قمنا بفتحه سنجد داخله مجلد اسمه [Local] وهذا المجلد يحتوي أيضاً على مجلد اسمه [Programs] نفتح، وسنجد مجلد بايثون حسب النسخة التي قمنا بتنزيلها

نفتح موجه الأوامر ونكتب التعليمة التالية:

[pip install jupyter]

(ملاحظة: يجب أن يتوفر اتصال بالإنترنت)

خامساً: نقوم بإنشاء مجلد جديد لحفظ البرامج التي سيتم إنشاؤها ضمن هذا المجلد

الآن نعود إلى موجه الأوامر ونكتب [jupyter notebook]

ننتظر قليلاً حتى يبدأ دفتر جوبيتر.

يعمل دفتر جوبيتر في مستعرض ويب مثل فايرفوكس أو غوغل كروم. يمكن كتابة برامج باستخدام لغات برمجة عديدة حيث يدعم دفتر جوبيتر أكثر من مئة لغة برمجة، يمكن استخدام لغة البرمجة المراد العمل بها من خلال تحديد النواة.

عندما نفتح دفتر جوبيتر جديد، سنرى الواجهة، في الجزء العلوي من دفتر الملاحظات، نرى شعار جوبيتر واسم ملف العمل الحالي.

يمكن تصدير الدفتر إلى مجموعة متنوعة من الصيغ بما في ذلك لغة ترميز النصوص التشعبية ونموذج الملفات المحمولة.

كانت خلية الإخراج تنتج مخططات أو صورًا، فستظهر في خلايا الإخراج.

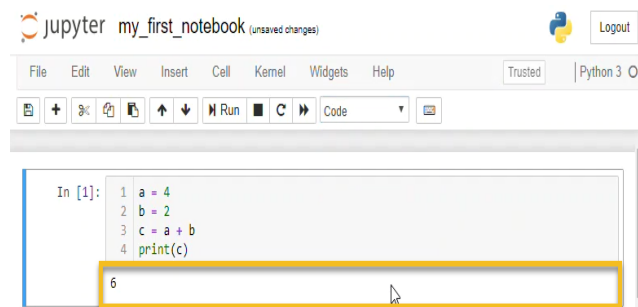
Exploring Jupyter notebook

python language

mechatronic engineering

AI and data science

الشكل (1-4)

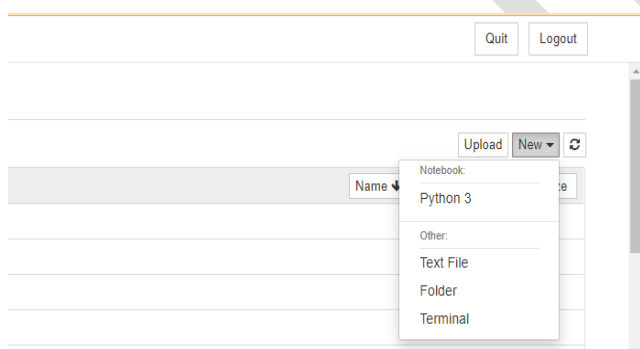


الشكل (1-2)

III. البرمجة التفاعلية في دفتر جوبيتر باستخدام بايثون

عند تشغيل دفتر جوبيتر نختار نواة بايثون التي تدعى python3 والتي تكون موجودة بشكل تلقائي كما هو موضح في الشكل

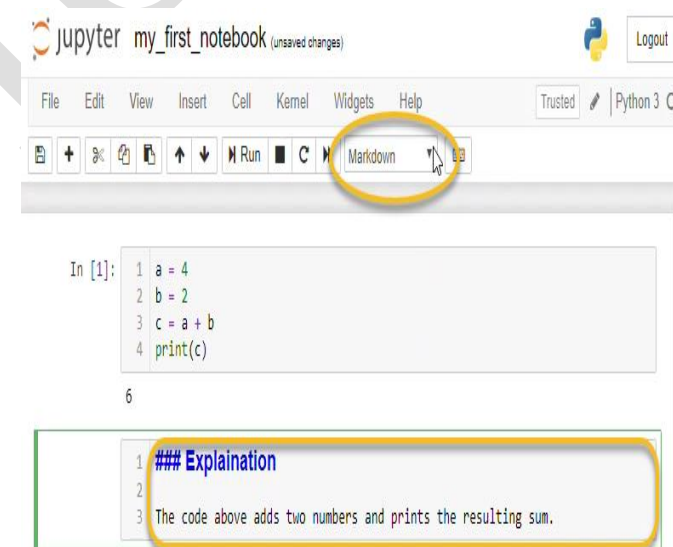
(2-1).



الشكل (2-1)

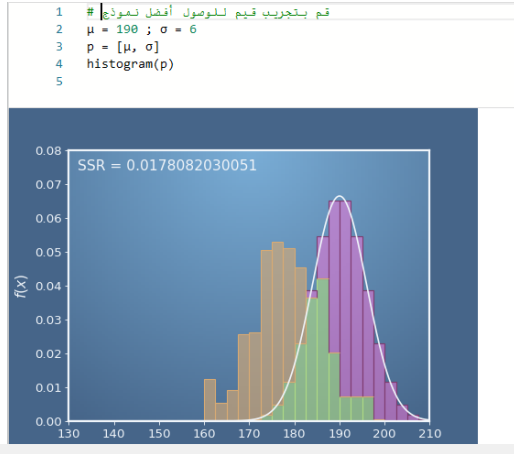
بعدها يمكن استخدام رمز بايثون بشكل طبيعي كما هو موضح في الشكل (2-2).

دفتر جوبيتر يحوي خاصية تنسيق النص حيث يتيح شرح الرمز من خلال الكتابة النصية بأحجام مختلفة، غامقة ومائلة. يمكن عرض الجداول والصور والمخططات ومقاطع فيديو والرموز معًا في دفتر جوبيتر نفسها عن طريق هذه الخاصية.



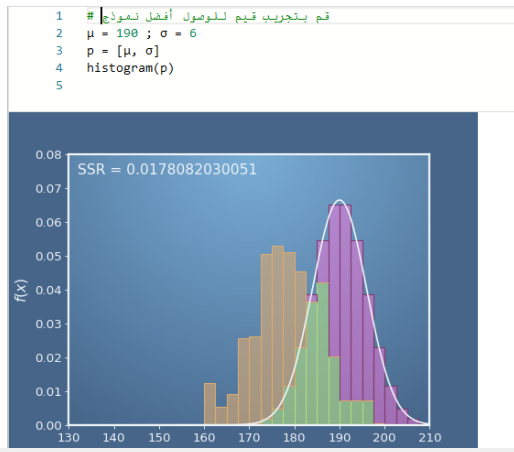
الشكل (1-3)

نلاحظ عند تنفيذ الخلايا السابقة نجد أن طريقة كتابة النص اختلفت حسب استخدام الوسوم كما هو موضح في الشكل (1-4)



الشكل (2-4)

ويمكن تكرار ذلك حتى الوصول للنموذج الأمثل

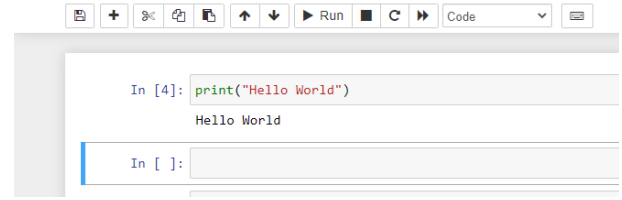


الشكل (2-5)

IV. البرمجة التفاعلية في دفتر جوبيتر باستخدام لغة سي بلس بلس

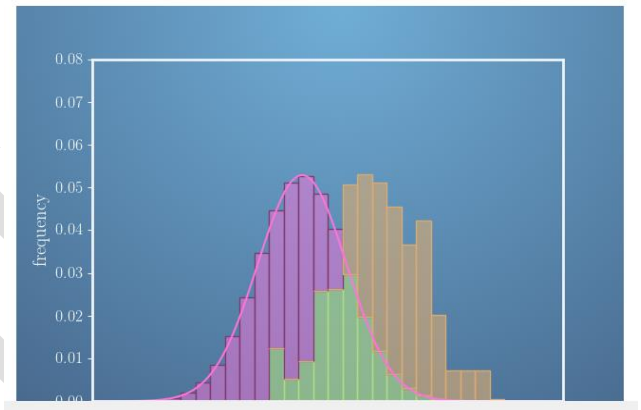
نعلم أنه في لغة سي بلس بلس يجب تصريف compiling البرامج أولاً ثم تنفيذها على عكس بايثون حيث أنها لغة تستخدم المفسر interpreter.

حتى نتمكن من كتابة برامج بلغة سي بلس بلس في دفتر جوبيتر يجب تثبيت نواة زيوس كلينج xeus-cling التي تسمح للمستخدمين بتشغيل رماز سي بلس بلس الخاصة بهم في دفتر جوبيتر. حيث



الشكل (2-2)

لإظهار امكانية العمل التفاعلي على دفتر جوبيتر لنأخذ المثال التالي: بفرض لدينا مجموعة عينات تمثل توزع أطوال مجموعة من الناس موزعة كما هو موضح (موضحة باللون البرتقالي) في الشكل (2-3).



الشكل (2-3)

نريد أن نتوقع عينات جديدة باستخدام النموذج الغوسي (موضح باللون الأرجواني في نفس الشكل) وبالتالي نحن بحاجة لمعرفة أمثل قيم للتباين σ والمتوسط μ (بحيث يغطي النموذج أكبر عدد من العينات (اللون الأخضر) كما هو موضح في الشكل (2-3). يمكننا اختيار قيم تجريبية ومشاهدة الناتج بشكل مباشر كما هو موضح في الشكل (2-4)

```
In [1]: #include <iostream>

In [2]: using namespace std;

In [3]: cout << "Hello World!" << endl;
Hello World!

In [4]: cout << "I am using Xeus-Cling!" << endl;
I am using Xeus-Cling!
```

الشكل (2-3)

هناك أدوات تفاعلية أخرى تدعم لغة سي بلس بلس في دفتر جوبيتر تسمح للمستخدم بناء واجهات رسومية وتصوّر البيانات التفاعلية المضمنة فيه. تعد اكس ويدجيت xwidgets التي لا تزال في مرحلة مبكرة من التطوير إحدى هذه الأدوات. وهي ليست مجرد مجموعة من عناصر واجهة المستخدم، ولكنها إطار يمكن البناء عليه لإنشاء مكونات مرئية عشوائية. يوضح الشكل (3-3) إنشاء زر وتجربته بشكل فوري

Button widget

```
In [10]: #include "xwidgets/xbutton.hpp"

In [11]: xw::button button;

In [12]: void foo()
{
    std::cout << "Clicked!" << std::endl;
}

In [13]: button.on_click(foo);
button.display()

Clicked!
Clicked!
Clicked!

In [13]: button.description = "button";
```

Widgets layout

الشكل (3-3)

V. البرمجة التفاعلية في دفتر جوبيتر باستخدام لغة جافا

حتى نستطيع استخدام لغة جافا Java في دفتر جوبيتر يجب تثبيت نواة آي جافا IJava كما يجب أن تكون نسخة جافا التي

أنها تستخدم مصرف تفاعلي يعمل على مبدأ (قراءة - تقييم - طباعة - حلقة) والذي تستخدمه أيضًا لغات البرمجة مثل جوليا.

أفضل طريقة لتثبيت نواة زيوس-كلينج هي إنشاء بيئة كوندا جديدة وتثبيت النواة هناك.

وذلك عن طريق موجه أوامر أناكوندا، لننشئ بيئة كوندا جديدة ونفعلها وذلك من خلال مايلي:

- سنستخدم مثبت حزمة كوندا لتثبيت زيوس كلينج.

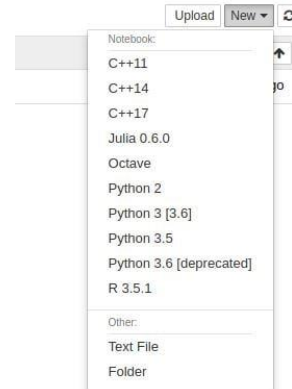
conda install -c conda-forge xeus-cling

- دعنا أولاً ننشط بيئة كوندا ثم نشغل دفتر جوبيتر

source activate xeus-cling

jupyter-notebook

سيبدأ خادم الكمبيوتر المحمول في العمل على المضيف المحلي الخاص بك. بعد ذلك، قم بإنشاء دفتر جوبيتر جديد وحدد نواة [C ++ 11]. كما هو موضح في الشكل (3-1)



الشكل (3-1)

بمجرد فتح دفتر جوبيتر، يمكننا البدء في العمل عليه باستخدام رماز سي بلس بلس كما هو موضح في الشكل (3-2).

بشكل افتراضي ، يتم تثبيت ipywidgets في Anaconda أو يمكنك أيضًا تثبيته يدويًا باستخدام conda. فيما يلي بعض الأمثلة على ipywidgets.

تسمح أداة النص للمستخدم بكتابة السلسلة المحرفية المبينة في الشكل (4-2).

تملكها على جهازك هي جافا 9 أو أحدث، لتنزيل نواة جافا لنتبع الخطوات التالية:

لنفتح موجه الأوامر ونكتب الأمر التالي:

Git clone https://github.com/SpencerPark/IJava.git

ثم نتوجه إلى ملف الجافا من خلال الأمر:

cd IJava

الأمر التالي يقوم بتنصيب النواة:

./gradlew installKernel

الآن نشغل دفتر جوبتير كما مر معنا ونختار نواة جافا كما هو موضح في الشكل (4-1)

```
from ipywidgets import widgets
title_textbox = widgets.Text(
    value = 'Hello World',
    description = 'Title:',
)
title_textbox
```

Output:

Title: Hello World

الشكل (4-2)

لإنشاء زر ، نكتب الرمز المبين في الشكل (5-2) ..

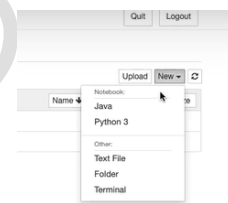
```
from ipywidgets import widgets
button = widgets.Button(
    description='Press Me',
)
button
```

Output:

Press Me

الشكل (5-2)

الشكل (6-2) يبين رمز برنامج لتصميم واجهة مستخدم لإدخال عددين وجمعهما.



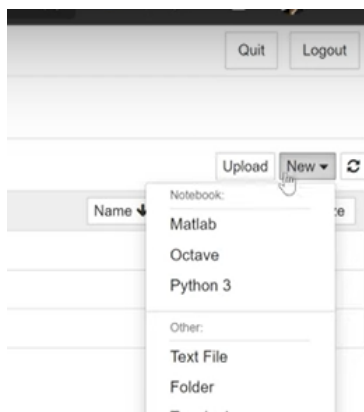
الشكل (4-1)

والآن يمكننا استخدام رمز جافا في دفتر جوبتير

```
In [16]: StringBuilder sb = new StringBuilder() ;
In [17]: sb.append("Hello World!!").append(" this is a test in jupyter notebook") ;
Out[17]: Hello World!! this is a test in jupyter notebook
In [ ]: |
```

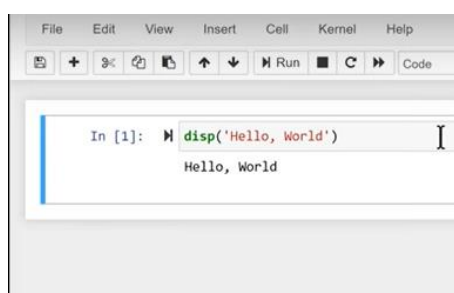
من الأدوات المثيرة للاهتمام الموجودة مع جافا أداة أيبويديجت ipywidgets التي توفر العديد من واجهات المستخدم الشائعة لاستكشاف التعليمات البرمجية والبيانات بشكل تفاعلي.

نفتح دفتر جوبتير ونختار نواة أوكتاف كما هو موضح في الشكل (5-1)



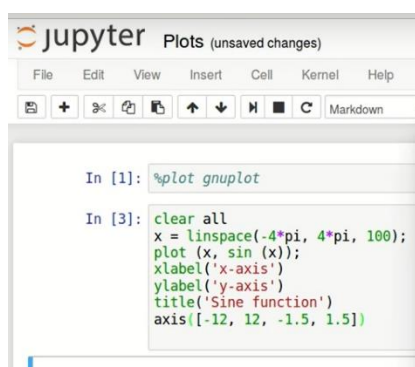
الشكل (5-1)

والآن يمكن استخدام رماز ماتلاب كما هو موضح في الشكل (5-2)



الشكل (5-2)

يوضح الشكلان (5-3) (5-4) رسم المخططات والحصول على خرج فوري على سبيل المثال قمنا برسم إشارة جيبية



الشكل (5-3)

```
from ipywidgets import widgets
lb1 = widgets.Label("Enter the First number")
display(lb1)
text1 = widgets.Text()
display(text1)
lb2 = widgets.Label("Enter the Second number")
display(lb2)
text2 = widgets.Text()
display(text2)
btn = widgets.Button(description = "add")
display(btn)
lb3 = widgets.Label()
display(lb3)
def add(x):
    a = int(text1.value)
    b = int(text2.value)
    lb3.value = 'result=' + str(a+b)
btn.on_click(add)
```

Output:

Enter the First number
10

Enter the Second number
10

add

result=20

الشكل (6-2)

VI. البرمجة التفاعلية في دفتر جوبتير باستخدام لغة أوكتاف

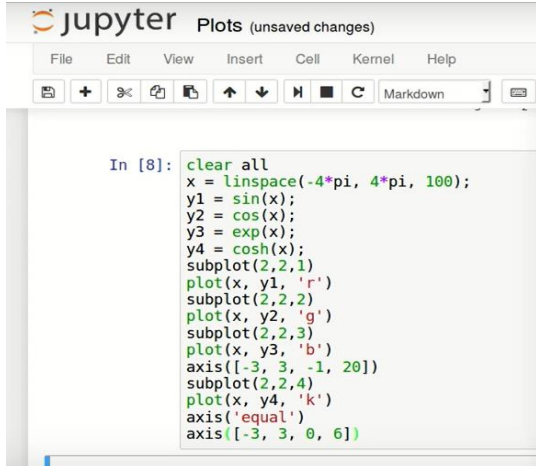
حتى نتمكن من استخدام أوكتاف Octave (وهي نسخة اللغة المفتوحة المصدر البديلة للغة الماتلاب) ضمن دفتر جوبتير يجب تثبيت نواة أوكتاف.

يتم ذلك عن طريق موجه الأوامر وذلك بكتابة الأمر:

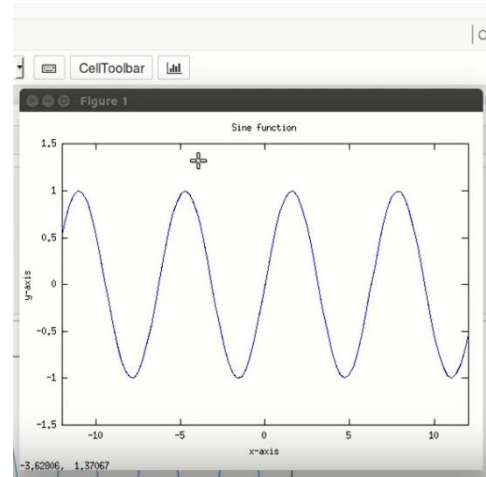
pip install octave-kernel

أو اذا أردنا استخدام موجه أوامر كوندا فيتم ذلك من خلال الأمر:

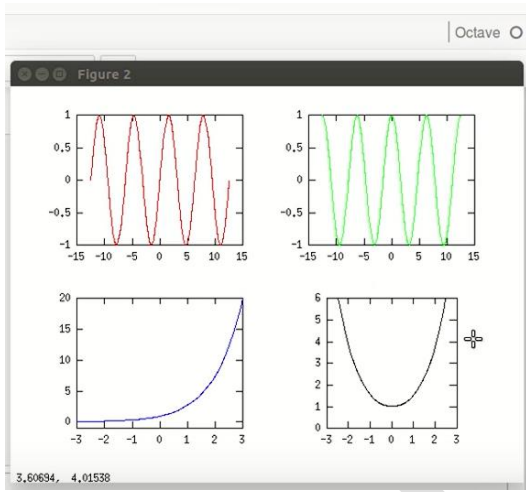
conda install -c conda-forge octave_kernel



الشكل (5-6)

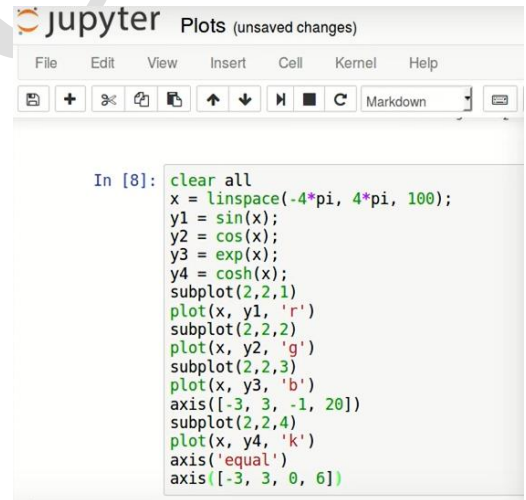


الشكل (5-4)



الشكل (5-7)

يوضح الشكل (5-5) رسم عدة مخططات بحيث كل مخطط على حدى



الشكل (5-5)

يوضح الشكلان (5-6) و (5-7) رسم عدة مخططات بحيث كل مخطط على حدى

VII. الخاتمة

تتوفر العديد من الأدوات والدفاتر المجانية لمساعدة المعلمين والمدرسين في مهامهم في تعليم دروس البرمجة وعلوم البيانات ، في هذه الورقة عرضنا دفاتر جوبيتر كخيار رائع لسهولة تنصيبها واستخدامها وإمكانياتها دون تكبد عناء تنصيب بيئة تطوير متكاملة . ورأينا كيف يمكن استخدام العديد من لغات البرمجة ضمنها والذي يفتح آفاق واسعة أمام المدرسين في المجال التربوي والأكاديمي.

منشورات المؤلف (د. إيد حاتم) :

- [13]. Dobler, S., Dalla, S., Wagschal, V., and Agrawal, A.A. (2012). Community-wide convergent evolution in insect adaptation to toxic ardenolides by substitutions in the Na,K-ATPase. *Proc. Natl. Acad. Sci. USA* 109:13040–13045.
- [14]. Dalla, S., Swarts, H. G. P., Koenderink, J. B., and Dobler, S. (2013). Amino acid substitutions of Na,K-ATPase conferring decreased sensitivity to cardenolides in insects compared to mammals. *Insect Biochem. Molec. Biol.* 43:1109–1115.
- [15]. Dalla, S. (2015). Analysis of resistance mechanisms in the Na,K-ATPase of cardenolide-adapted insects. *University of Hamburg*.
- [16]. Dalla, S., and Dobler, S. (2016). Gene duplications circumvent trade-offs in enzyme function: Insect adaptation to toxic host plants. *Evolution* DOI: 10.1111/evo.13077.
- [17]. Dalla, S., Baum, M., and Dobler, S. Substitutions in the cardenolide binding site and interaction of subunits affect kinetics besides cardenolide sensitivity of insect Na,K-ATPase. *Insect iochem. Molec. Biol.* DOI: 10.1016/j.ibmb.2017.08.005.
- [18]. Jennifer N. Lohr., Fee Meinzer., Safaa Dalla., Renja Romey-Glusing and Dobler, S. (2017). The function and evolutionary significance of a triplicated Na,K-ATPase gene in a toxin specialized insect. *BMC Evolutionary Biology* DOI 10.1186/s12862-017-1097-6.
- [19]. Safaa Dalla., Ola Rajab., Walaa Abdalla. (2021). Stem cells in the treatment of peripheral nerve injury. *Manara journal*.
- [20]. Suliman, M., Dalla. (2021). CRISPR technology and its promising use in gene therapy. *Manara journal*.

- [1]. Tools For Data Science course by IBM on Coursera
<https://www.coursera.org/learn/open-source-tools-for-data-science/home/welcome>
- [2]. Data Analysis with python course by IBM on Coursera
<https://www.coursera.org/learn/data-analysis-with-python/home/welcome>
- [3]. Machine Learning with python course by IBM on Coursera
<https://www.coursera.org/learn/machine-learning-with-python/home/welcome>
- [4]. Data Science with jupyter by Prateek Gupta-Ebook
- [5]. Using Octave kernel in a Jupyter Notebook
<https://dev.to/kherin/using-octave-kernel-in-a-jupyter-notebook-3im>
- [6]. Xeus-Cling: Run C++ code in Jupyter Notebook
<https://learnopencv.com/xeus-cling-run-c-code-in-jupyter-notebook/>
- [7]. Interactive Workflows for C++ with Jupyter
<https://blog.jupyter.org/interactive-workflows-for-c-with-jupyter-fe9b54227d92>
- [8]. Teaching Java with Jupyter notebooks
<https://blog.frankel.ch/teaching-java-jupyter-notebooks/>
- [9]. IJava
<https://github.com/SpencerPark/IJava>
- [10]. Jupyter Alternatives
<https://geekflare.com/best-data-science-notebooks/>
- [11]. 10 Reasons to Use Interactive Notebooks
<https://blog.penningtonpublishing.com/reading/10-reasons-to-use-interactive-notebooks/>
- [12]. Adding the Java Kernal to your installation
<https://subscription.packtpub.com/book/big-data-and-business->