



Women Techmakers  
Saudi Arabia

العطاء الرقمي  
Attaa Digital



بالعطاء نبرمج

عالم البرمجة ولغاتها

الجزء الثاني



>Hello world!

>\_

Ruqiya Bin Safi

WTM Ambassador

 @Ru0Sa

START

العطاء الرقمي  
Attaa Digital



Women Techmakers  
Saudi Arabia

# المحاور

- التركيز على البايثون Python كمثال على لغات البرمجة
- ماهي البايثون؟ نبذة عنها
- لماذا البايثون؟ مميزاتها
- ماذا يمكن أن تفعل بالبايثون؟ مجالات استخدامها
- تطبيق عملي: مشروع بسيط بالبايثون



# التركيز على البايثون Python كمثال على لغات البرمجة

A person is seen from behind, sitting at a desk and working on a computer. The computer monitor displays a code editor with lines of code on the left and a line chart with green bars on the right. The background is dark and slightly blurred. In the top-left corner, there are three overlapping brush strokes in blue, cyan, and green. The text 'ماهي البايثون؟ نبذة عنها' is overlaid in white on the right side of the image.

# ماهي البايثون؟ نبذة عنها





# ماهي البايثون؟ نبذة عنها



صنف موقع PYPL لتصنيف لغة البرمجة Python على أنها لغة البرمجة الأولى التي اكتسبت شهرة كبيرة في عام 2019

Worldwide, Jan 2020 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	29.72 %	+4.3 %

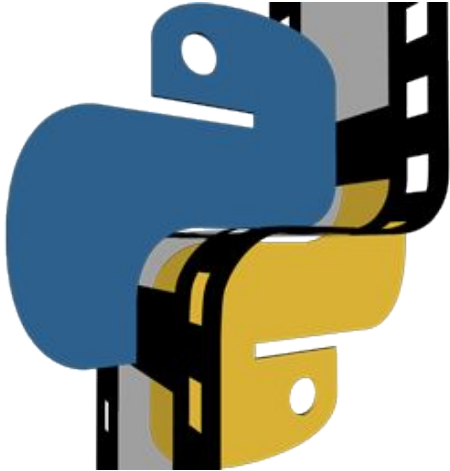
المصدر: PYPL



# ماهي البايثون؟ نبذة عنها



- ظهرت Python في 1991 عندما قام Guido van Rossum بتطوير كمشروع جانبي له ولم يعتقد أحد أنها ستكون لغة البرمجة الأكثر شعبية في يوم من الأيام.
- لم تشهد في البداية ارتفاعًا كبيرًا في الشعبية مثل Java أو C / C ++.
- ركزت على تبسيط البرمجة على المطورين وحتى يتمكن طلاب المدارس من كتابة الاكواد.
- في عام 2008 ، خضعت Python لعملية إصلاح وتحسين كبيرة من خلال تقديم Python 3.
- اليوم موجودة في كل مكان وتستخدم في العديد من مجالات تطوير البرمجيات.







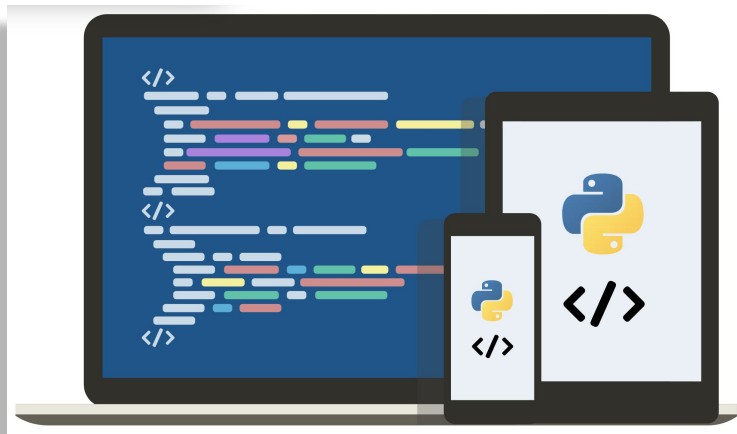
# لماذا البايثون؟ مميزاتها



# لماذا البايثون؟ مميزاتها

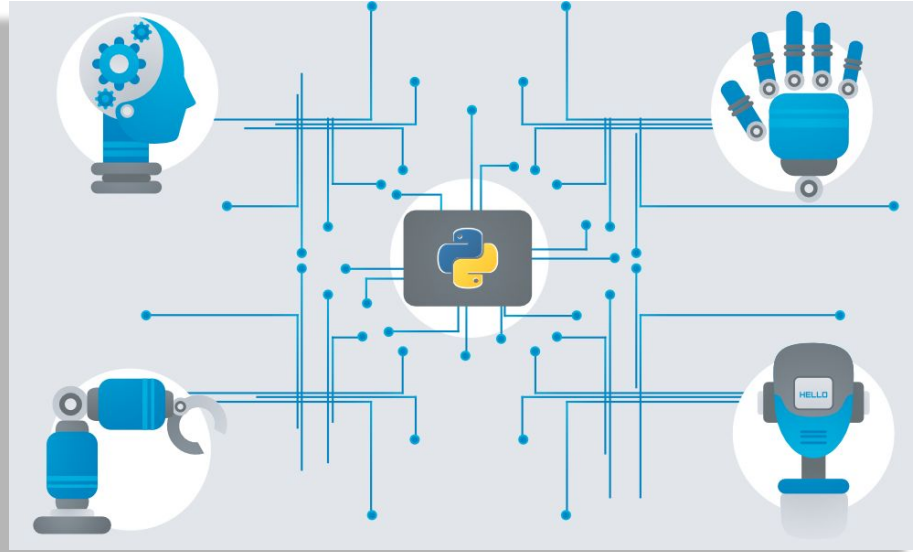


- تحتوي على بنية مشابهة للغة الإنجليزية وتصميم لغتها بسيط ولكنه قوي.
- تسمح للمطورين بكتابة البرامج اسطر أقل من بعض لغات البرمجة الأخرى.
- تستخدم مفسر Interpreter مما يعني أنه يمكن تنفيذ التعليمات البرمجية بسرعة.
- وجود مجموعة كبيرة من المكتبات التي يمكنك تضمينها والاستفادة منها.
- تمتلك بايثون مجتمع كبير داعم ونشيط جدًا.
- تعمل على منصات مختلفة (Windows و Mac و Linux و Raspberry Pi ...).
- لديها تكامل مع C / C ++ فاكسبها قوة.





ماذا يمكن أن تفعل بالبايثون؟  
مجالات استخدامها



- علوم البيانات
- تحليلات البيانات
- الذكاء الاصطناعي تعلم الآلة والتعلم العميق
- تطبيقات المؤسسات
- تطوير الويب
- Data Science
- Data Analytics
- Artificial Intelligence, ML and DL
- Enterprise Application
- Web Development

# تطبيق عملي

```
int main()
{
    char *ptr, *p = NULL, q[20];
    int x;

    /* initialize our dialog setup */
    init_dialog();

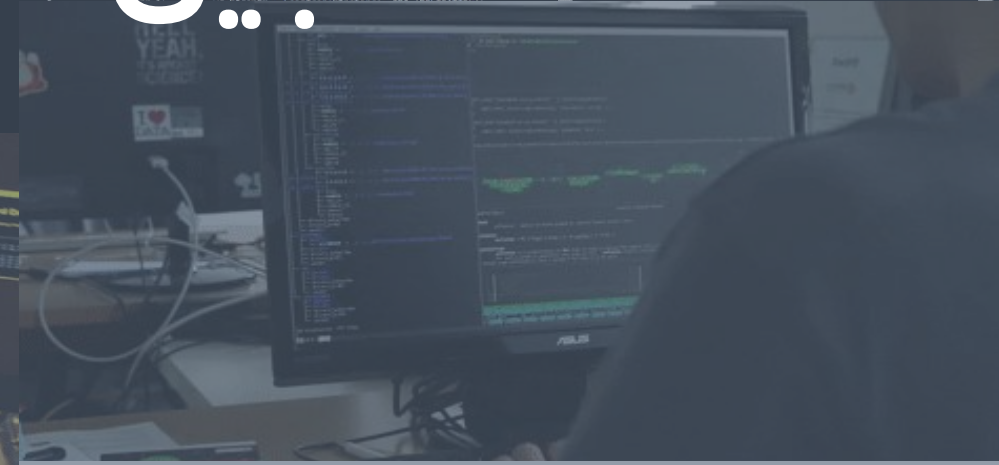
    /* dialog_prqbox(unsigned char *title, const unsigned char *line,
    * int height, int width, int pause, int use_shell);
    *
    * title: is the command we want to execute
    * pause: confirmation upon termination
    * use_shell: use 'sh' instead of 'exec()';
    */
    dialog_prqbox("Example #1", "id", 7, 90, 1, 0);
    crt(NULL);

    /* This contains a message and a OK button */
    dialog_msgbox("Example #2", "inExample using dialog_msgbox()", 10, 30);
    crt(NULL);

    /* This contains a message and YES/NO buttons
    * the return value is stored to x; YES = 0, NO = 1 */
    x = dialog_yesno("Example #3", "inExample using dialog_yesno()", 10, 30);
    if (x == 0)
        crt("yes");
    else
        crt("no");

    /* array is now sorted */
    /* sorted A[]; break; */

    /* Reduce upper limit on array since the larger values have
    been swept to the end. */
    u1 ;
}
```



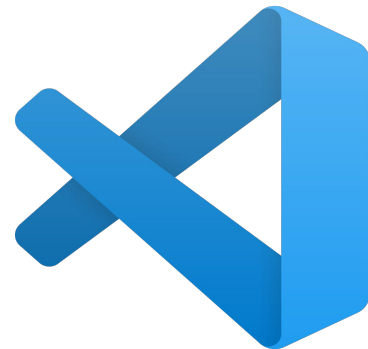




# أدواتنا في ورشة اليوم



Flask



HTML



CSS



GitHub



heroku

# Beginner's Python Cheat Sheet

## Variables and Strings

*Variables are used to store values. A string is a series of characters, surrounded by single or double quotes.*

### Hello world

```
print("Hello world!")
```

### Hello world with a variable

```
msg = "Hello world!"  
print(msg)
```

### Concatenation (combining strings)

```
first_name = 'albert'  
last_name = 'einstein'  
full_name = first_name + ' ' + last_name  
print(full_name)
```

## Lists

*A list stores a series of items in a particular order. You access items using an index, or within a loop.*

### Make a list

```
bikes = ['trek', 'redline', 'giant']
```

### Get the first item in a list

```
first_bike = bikes[0]
```

### Get the last item in a list

```
last_bike = bikes[-1]
```

### Looping through a list

```
for bike in bikes:  
    print(bike)
```

### Adding items to a list

```
bikes = []  
bikes.append('trek')  
bikes.append('redline')  
bikes.append('giant')
```

### Making numerical lists

```
squares = []  
for x in range(1, 11):  
    squares.append(x**2)
```

## Lists (cont.)

### List comprehensions

```
squares = [x**2 for x in range(1, 11)]
```

### Slicing a list

```
finishers = ['sam', 'bob', 'ada', 'bea']  
first_two = finishers[:2]
```

### Copying a list

```
copy_of_bikes = bikes[:]
```

## Tuples

*Tuples are similar to lists, but the items in a tuple can't be modified.*

### Making a tuple

```
dimensions = (1920, 1080)
```

## If statements

*If statements are used to test for particular conditions and respond appropriately.*

### Conditional tests

equals	x == 42
not equal	x != 42
greater than	x > 42
or equal to	x >= 42
less than	x < 42
or equal to	x <= 42

### Conditional test with lists

```
'trek' in bikes  
'surly' not in bikes
```

### Assigning boolean values

```
game_active = True  
can_edit = False
```

### A simple if test

```
if age >= 18:  
    print("You can vote!")
```

### If-elif-else statements

```
if age < 4:  
    ticket_price = 0  
elif age < 18:  
    ticket_price = 10  
else:  
    ticket_price = 15
```

## Dictionaries

*Dictionaries store connections between pieces of information. Each item in a dictionary is a key-value pair.*

### A simple dictionary

```
alien = {'color': 'green', 'points': 5}
```

### Accessing a value

```
print("The alien's color is " + alien['color'])
```

### Adding a new key-value pair

```
alien['x_position'] = 0
```

### Looping through all key-value pairs

```
fav_numbers = {'eric': 17, 'ever': 4}  
for name, number in fav_numbers.items():  
    print(name + ' loves ' + str(number))
```

### Looping through all keys

```
fav_numbers = {'eric': 17, 'ever': 4}  
for name in fav_numbers.keys():  
    print(name + ' loves a number')
```

### Looping through all the values

```
fav_numbers = {'eric': 17, 'ever': 4}  
for number in fav_numbers.values():  
    print(str(number) + ' is a favorite')
```

## User input

*Your programs can prompt the user for input. All input is stored as a string.*

### Prompting for a value

```
name = input("What's your name? ")  
print("Hello, " + name + "!")
```

### Prompting for numerical input

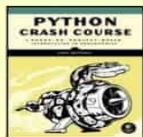
```
age = input("How old are you? ")  
age = int(age)
```

```
pi = input("What's the value of pi? ")  
pi = float(pi)
```

## Python Crash Course

Covers Python 3 and Python 2

[nostarchpress.com/pythoncrashcourse](http://nostarchpress.com/pythoncrashcourse)





# مراجع قد تحتاجها





# المثال الأول : تمهيد



attaa-hello-world.herokuapp.com

Hello World!

<https://attaa-hello-world.herokuapp.com>

[https://github.com/Ruqyai/Attaa\\_workshop/tree/master/hello\\_world](https://github.com/Ruqyai/Attaa_workshop/tree/master/hello_world)



تأكد من تحميلك للبايثون أو حملها من الرابط

Make sure that you have python or download it [here](#)

## Step-1: Create Python Virtual Environment

قم بكتابة الأمر التالي في Terminal برنامج VSCode

```
python -m venv attaa  
source attaa/bin/activate
```



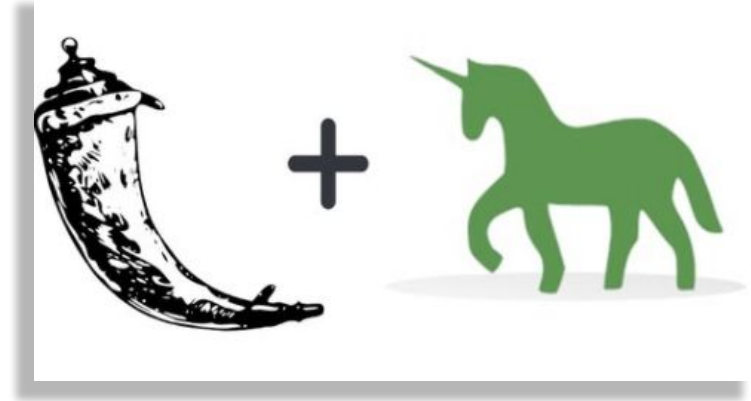




## Step-2: Install Flask & Gunicorn

قم بكتابة الأمر التالي في Terminal برنامج VSCode

```
pip install flask gunicorn
```





## Step-3: Create an app folder and simple python app



hello\_world

إما أن تقوم بإنشاء مجلد يدويا وتدخل بداخله

```
mkdir hello_world  
cd hello_world
```

أو قم بكتابة الأمر التالي في Terminal برنامج VSCode بدلا عن ذلك



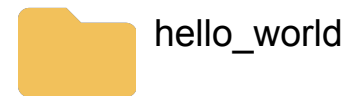
## Step-4: Create the application, app.py

قم بإنشاء مجلد بداخله ملف app.py وضع فيه الكود التالي

```
# app.py
from flask import Flask # import flask
app = Flask(__name__) # create an app instance

@app.route("/") # at the end point /
def hello(): # call method hello
    return "Hello World!" # which returns "hello world"

if __name__ == "__main__": # on running python app.py
    app.run() # run the flask app
    app.run(debug=True) # debugger will be active
```



hello\_world



app.py



## Step-5: Run the application in your local system

```
python app.py
```

قم بكتابة الأمر التالي في Terminal برنامج VSCode

```
(tensorflow_env) Ruqiya:hello_world RS$ python app.py
* Serving Flask app "app" (lazy loading)
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [16/Aug/2020 09:52:28] "GET / HTTP/1.1" 200 -
█
```

سيظهر لك مثل هذا وبه الرابط

قم بالدخول على الرابط

← → ↻ 🏠 ⓘ 127.0.0.1:5000

Hello World!



## Step-6: Install Heroku CLI

If not have account please (Sing up)

### For Mac

سجل بالموقع و قم بكتابة الأمر التالي في Terminal لتحميل الأداة اذا جهازك ماك

```
brew tap heroku/brew && brew install heroku
```

Other systems [here](#)

للأجهزة الأخرى ادخل الرابط







## Step-7: Create requirements.txt and Procfile file

قم بكتابة الأمر التالي في Terminal برنامج VSCode

```
pip freeze > requirements.txt
```

Procfile



قم بإنشاء ملف جديد اسمه Procfile وضع بداخله التالي

```
web: gunicorn app:app
```



## Step-8: Create an app in Heroku

قم بإنشاء تطبيق جديد بالموقع


### Create New App


App name

✓

**app-attaa** is available

Choose a region

 United States

 Add to pipeline...

Create app



## Step-9: Deploy your app to heroku

انشر موقعك بكتابة الأوامر التالية تباعا

```
heroku login
```

الأمر الأول للمصادقة و سينقلك إلى المتصفح

```
git init
```

```
heroku git:remote -a app-attaa
```

انسخ الاوامر التالية لربط عملك ب heroku  
ملاحظة بعد حرف ال -a هو اسم تطبيقك أو موقعك

```
git add .
```

```
git commit -am "make it better"
```

```
git push heroku master
```

انسخ الاوامر التالية لرفعها ورفع التحديثات التي تقوم بها

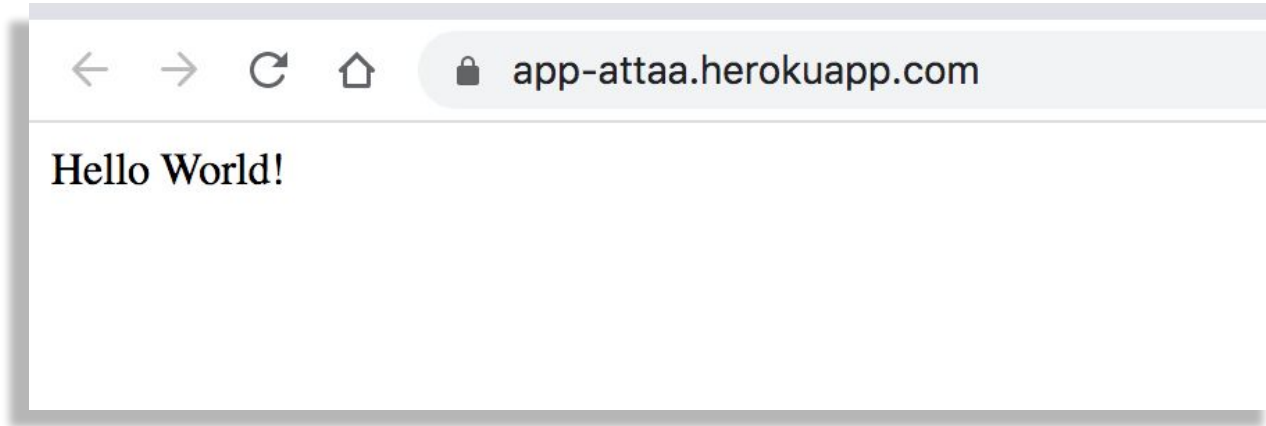
بعد الانتهاء سيظهر لك رابط موقعك في ال Terminal



## Step-10: Open your application on browser

<https://app-attaa.herokuapp.com>

ادخل على الرابط وسيظهر موقعك

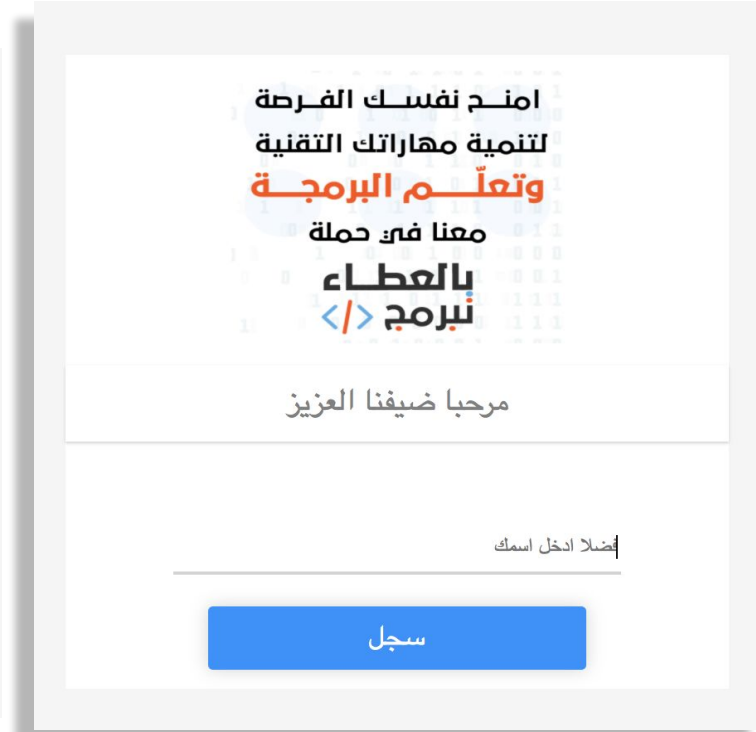




# المثال الثاني : موقع بسيط



<https://attaa.herokuapp.com>



[https://github.com/Ruqyai/Attaa\\_workshop/tree/master/hello\\_web](https://github.com/Ruqyai/Attaa_workshop/tree/master/hello_web)





## Step-1: Create Python Virtual Environment

قم بكتابة الأمر التالي في Terminal برنامج VSCode

```
python -m venv attaa-venv  
source attaa-venv/bin/activate
```

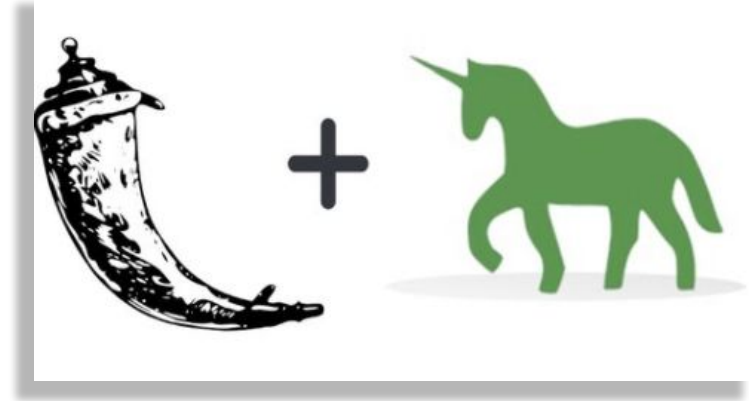




## Step-2: Install Flask & Gunicorn

قم بكتابة الأمر التالي في Terminal برنامج VSCode

```
pip install flask gunicorn
```





## Step-3: Create an app folder and simple python app



hello\_web

إما أن تقوم بإنشاء مجلد يدويا وتدخل بداخله

```
mkdir hello_web  
cd hello_web
```

أو قم بكتابة الأمر التالي في Terminal برنامج VSCode بدلا عن ذلك



## Step-4: Create the application

هذه المجلدات والملفات التي يحتويها موقعك افتح الرابط  
بالأسفل لتشاهد الأكواد الموجودة فيها والصور والملفات



hello\_web



static

img



hello.css



index.css



template

hello.html



index.html



app.py

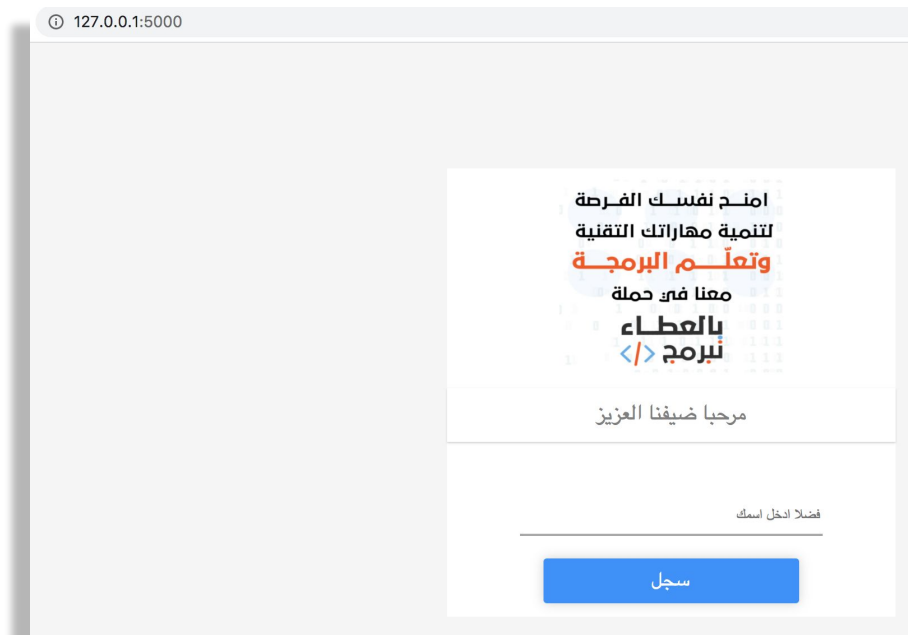
[https://github.com/Ruqyai/Attaa\\_workshop/tree/master/hello\\_web](https://github.com/Ruqyai/Attaa_workshop/tree/master/hello_web)



## Step-5: Run the application in your local system

```
python app.py
```

قم بكتابة الأمر التالي في Terminal برنامج VSCode





## Step-6: Create requirements.txt and Procfile file

```
pip freeze > requirements.txt
```

قم بكتابة الأمر التالي في Terminal برنامج VSCode

Procfile



قم بإنشاء ملف جديد اسمه Procfile وضع بداخله التالي

```
web: gunicorn app:app
```




## Step-7: Create an app in Heroku

قم بإنشاء تطبيق جديد بالموقع



### Create New App


App name

**app-attaa** is available

Choose a region

 United States 

 Add to pipeline...

[Create app](#)





## Step-8: Deploy your app to heroku

انشر موقعك بكتابة الأوامر التالية تباعا

```
heroku login
```

الأمر الأول للمصادقة و سينقلك إلى المتصفح

```
git init
```

```
heroku git:remote -a attaa-flask
```

ملاحظة بعد حرف ال a- هو اسم تطبيقك او موقعك

```
git add .
```

```
git commit -am "make it better"
```

```
git push heroku master
```

انسخ الاوامر التالية لرفعها ورفع التحديثات التي تقوم بها

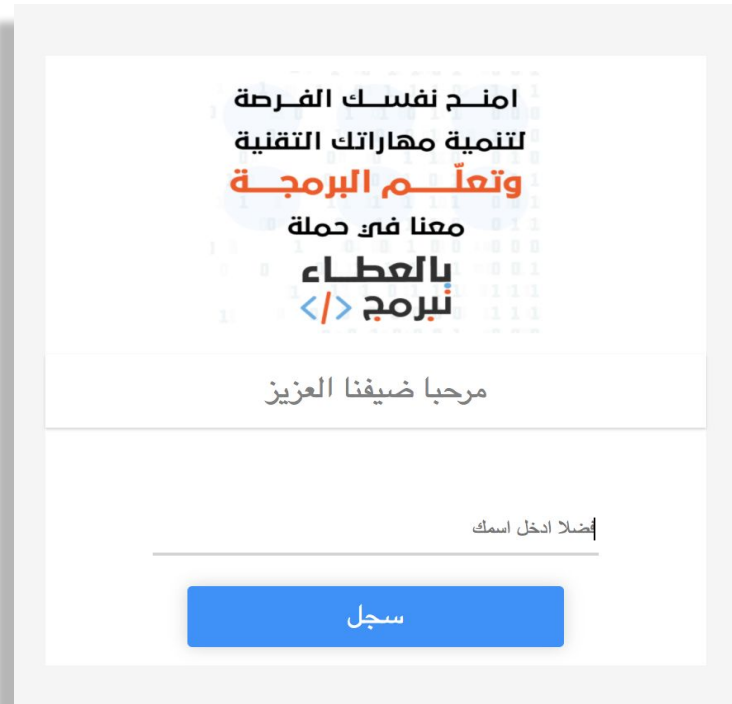
بعد الانتهاء سيظهر لك رابط موقعك في ال Terminal



## Step-9: Open your application on browser

<https://attaa-flask.herokuapp.com>

ادخل على الرابط وسيظهر موقعك





## توقع سعر البيت

عدد الغرف

عدد الحمامات

سنة البناء

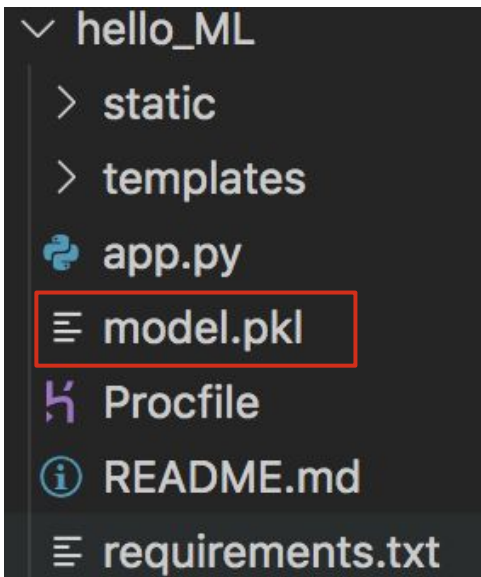
توقع



## Install libraries

هناك مكتبات جديدة

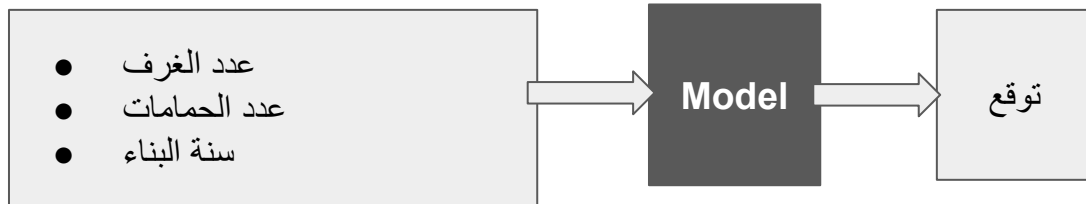
```
pip install flask gunicorn numpy joblib scikit-learn
```



نموذج  
تعلم آلة  
مدرب  
ومحفوظ

## Create UI that take the input of model

قم بإنشاء الواجهة



# مسابقة إنشاء موقع إلكتروني



## المتطلبات:

- إنشاء موقع بأحد لغة برمجية
- "يمكن إنشاء موقع من صفحة ويب واحدة"
- أن تحتوي على فكرة بسيطة
- أن تكون الواجهة جميلة تجذب الزوار
- إرفاق شرح عن عمل المقدم
- الاعتناء بالتوثيق ووضوح الكود

## معايير التقييم:

- الفكرة
- المظهر و التنسيق
- الكود الواضح المفهوم

آخر موعد لاستقبال المشاركات  
الثلاثاء 25 - 08 - 2020

\* سيتم الاختيار والفرز عن طريق محكمين

<https://attaa.sa/forms/view/8>

للمشاركة



# تم بحمد الله الجزء الثاني



**Ruqiya Bin Safi**

WTM Ambassador



@Ru0Sa