

# PYTHON COURSE

Python course , from scratch  
to Professionalism ...

Made by : Taha Khaldoun Amro 

# دورة تعليم لغة بايثون

## تقنيات النصوص

في لغة بايثون , يوجد لكل نوع من أنواع البيانات مجموعة من التقنيات المدمجة باللغة , التي تسهل مجموعة من العمليات التي قد تواجهها في مسيرة عملنا.  
تمتلك النصوص حصة الأسد من هذه التقنيات المدمجة , سنتعرف الان إلى أهم هذه التقنيات .

### 1. len()

تعمل تقنية len على إيجاد طول العنصر , و هي لا تقتصر على النصوص فقط , بل يمكن استخدامها في أنواع بيانات أخرى :

```
x = 'we love python'  
print(len(x))
```

### 2. strip()

تعمل تقنية strip على إزالة القيمة المعطاة من جوانب النص , حيث تكون علامة المسافة هي القيمة الاساسية .

```
x = '    we love python    ' # replace the spaces with anything and try to strip it  
print(x.strip()) # try to put another string value in the brackets
```

## تقنيات النصوص

### 3. title()

تعمل تقنية title على جعل أول حرف من كل كلمة في النص capital .

```
x = 'we love python'  
print(x.title())
```

### 4. capitalize()

تعمل تقنية capitalize على جعل أول حرف في النص capital و جعل باقي الكلمات تبدأ بأحرف small .

```
x = 'we Love Python'  
print(x.capitalize())
```

## تقنيات النصوص

### 5. zfill()

تعمل تقنية zfill على ملء الخانات الفارغة في الأرقام بأصفار لتصل الى عدد منازل معين :

001

005

010

050

100

200

و هكذا ...

```
x , c , z , v = "1" , "11" , "111" , "1111"  
print(x.zfill(4))  
print(z.zfill(4))  
print(c.zfill(4))  
print(v.zfill(4))
```

## تقنيات النصوص

### 6. upper() and lower()

تعمل هاتان التقنيتان على إعادة النص مع جعل كل أحرفه إما capital أو small .

```
x = 'we love python' # change the string to capital  
  
print(x.upper())  
  
print(x.lower())
```

### 7. split()

تعمل هذه التقنية على إعادة قائمة تتضمن كل كلمات الجملة .

```
x = 'we love python and programming'  
print(x.split()) # takes a second value of maxsplit times  
  
x = 'we-love-python-and-programming' # changing the chareacter between each word  
print(x.split('-')) # change the value to any string you want
```

## تقنيات النصوص

### 8. center()

تعمل هذه التقنية على توسيط الكلام باستخدام نص معين .  
يقبل الأمر قيمتين , قيمة رقمية تعبر عن طول النص الجديد بعد إضافة النصوص من حوله , و قيمة النصوص التي سيتم اضافتها.

```
x = 'Jomana'

print(x.center(11 , '-')) # 11 is the length of the new string
```

### 9. count()

تعمل هذه التقنية على إعادة عدد مرات تكرار عنصر ما داخل النص .

```
x = 'k;dajkls fdjvnd kfjvd ifnireljfvldfji n biuf hwopid svmk sfjbd kuhbs lkldf'

print(x.count('a'))
print(x.count('a' , 2 , 27 )) # takes also a starting and finishing positions : count('char' , start , end)
```

## تقنيات النصوص

### 10. swapcase()

تعمل هذه التقنية على قلب الأحرف (capital يصبح small و small يصبح capital).

```
x = 'we love python'  
print(x.swapcase())
```

### 11. startswith() and endswith()

تعمل هاتان التقنيتان على إرجاع قيمة منطقية تعبر عن إجابة للسؤال : هل يبدأ النص بحرف "س" أو هل ينتهي به ؟

```
myname = "jehad abo sondos"  
  
print(myname.endswith("s"))  
  
print(myname.startswith("w"))  
  
print(myname.startswith('d' , 4 , 10)) # both functions takes a starting and ending positions
```

## تقنيات النصوص

### 12. index()

تعمل هذه التقنية على اعادة رقم و موقع العنصر في النص.

```
x = 'we love python'  
print(x.index('p')) # takes a starting and ending positions
```

### 13. find()

تقوم هذه التقنية بنفس مهمة السابقة ولكن عند عدم إيجاد قيمة معينة , يقوم بطباعة 1- بدل من error .

تختلف تقنية index و تقنية find عن بعضهما البعض في أن تقنية index عندما لا تجد ما تبحث عنه داخل النص , فإنها تقوم برفع خطأ للمستخدم , مما يوقف الكود , قد تضطر أنت إلى استخدام الأمر find لأنه لا يقوم بإيقاف الكود , بل يعطي القيمة 1- عند عدم إيجاده لكلمة معينة .



## تقنيات النصوص

### 14. replace()

تعمل هذه التقنية على استبدال قيمة داخل النص بقيمة أخرى يحددها المستخدم .

```
x = 'we love python'
```

```
s = 'www.blabla.com'
```

```
print(x.replace('o' , 's')) # replace(old value,new value)
```

```
print(s.replace('w' , 'f', 1)) # replace(old value,new value , count) takes how many times we want to replace
```

## تقنيات النصوص

### 15. join()

تعمل هذه التقنية على ضم أعضاء و عناصر قائمة مع بعضهم بواسطة قيمة نصية.

```
x = ['first' , 'second' , 'third']  
  
print('-'.join(x)) # 'string'.join(list)
```

ما يتم هنا هو أن عناصر القائمة الموجودة في الكود , تجمع في نص واحد , بحيث يفصل بينها نص معين .

في القسم التالي من تقنيات النصوص , سنتحدث عن مجموعة من التقنيات التي تختص بإعادة القيم المنطقية , فمثلاً لو كنت تريد أن تعرف ما اذا كان نص ما يتكون من أحرف فقط , او من أرقام و أحرف , لا تقلق فقد وفرت لنا لغة بايثون كل ما نحتاج في مجموعة من التقنيات المدمجة .

# دورة تعليم لغة بايثون

## تقنيات النصوص

في هذه الجزئية من تقنيات النصوص سنتحدث عن بعض التقنيات التي تعيد لنا قيماً منطقية (نعم أو لا).

1. `istitle()`

نحن بطبيعة الحال نعرف ماذا تفعل تقنية `title` , تقوم هذه التقنية بفحص ما إذا كان النص يوافق نتاج العملية .

2. `isspace()`

تعمل هذه التقنية على فحص ما إذا كان النص عبارة عن `space` أو “مسافة”

3. `islower()`

تعمل هذه التقنية على فحص ما إذا كان النص يتكون من حروف `small` فقط

4. `isupper()`

تعمل هذه التقنية على فحص ما إذا كان النص يتكون من حروف `capital` فقط

5. `isidentifier()`

تعمل هذه التقنية على فحص ما إذا كان النص يصلح ان يكون اسم متغير

5. `isalpha()`

تعمل هذه التقنية على فحص ما إذا كان النص يتكون من أحرف فقط

### 6. isalnum()

تعمل هذه التقنية على فحص ما إذا كان النص يتكون من أحرف و أرقام.

مرفق لكم شرح لما جاء في الدرس :

[https://youtu.be/HmDLsnLgt0M?si=r6WHA\\_j1n9NxW3lv](https://youtu.be/HmDLsnLgt0M?si=r6WHA_j1n9NxW3lv)

<https://youtu.be/doDJDkUOEJQ?si=W1umyMYqe2zvu4f7>

<https://youtu.be/kgb96E9ogUw?si=x-j2XYXDpAbLOrYR>

<https://youtu.be/jbV9d9H-udY?si=SFP7NwyRsifZtxPI>

# PYTHON COURSE

Python course , from scratch  
to professionalism ...

Made by : Taha Khaldoun Amro 