

## Assignment 2

**Deadline: 8/8**

تحية طيبة لكم جميعاً،

يتكون هذا الواجب البيتي من سؤالين، يتعلق السؤال الأول بالنصوص Strings والسؤال الثاني في Lists القوائم. تتوفر المادة المساندة أدناه حيث يمكن قراءتها والاستفادة منها في الإجابة على السؤالين. ويوجد في آخر الواجب البيتي مجموعة من الروابط المفيدة من موقع W3School التي يمكن أن تساعدكم أيضاً في فهم المسائل وتجربتها.

---

### Question 1: write the code for the following challenges related to Strings:

كتابة كود بايثون لحل المسائل التالية (يمكن حل السؤال عن طريق خاصية slice في النصوص Strings)

```
my_string = "Hello, World, I like Python Code, Python is so amazing  
language"
```

- أ- طباعة الأحرف من الرابع حتى الحرف السادس
  - ب- طباعة الأحرف من الأول حتى الحرف الثامن
  - ت- طباعة الأحرف من الحرف السادس حتى الأخير
  - ث- تقسيم النص إلى نصوص أصغر عن طريق استخدام الفاصلة (,) comma
  - ج- استبدال كلمة Python بكلمة Amideast
  - ح- فحص هل النص يحتوي على أرقام أم لا؟
  - خ- فحص عدد مرات تكرار حرف e
  - د- تحويل النص كامل إلى أحرف كبيرة Capital letters
  - ذ- تحويل كامل النص إلى أحرف صغيرة small letters
- 

### Question 2: write Python code for the following challenges related to Lists:

كتابة كود بايثون لحل المسائل التالية المتعلقة ب Lists حسب المثال التالي ( يمكن طباعة القائمة في كل مرة للتأكد من أن حلنا صحيح، يمكنكم استخدام append, remove, insert, pop لتسهيل الحل لديكم).

```
my_list = ["apple", "banana", "cherry"]
```

- أ- طباعة القائمة List كاملة
- ب- طباعة العنصر الثاني في القائمة
- ت- تعديل العنصر الأول في القائمة ليصبح Cucumber بدلاً من apple
- ث- إضافة Orange إلى نهاية القائمة
- ج- إضافة Tomato إلى بداية القائمة
- ح- إضافة Apricot إلى الموقع الثالث في القائمة
- خ- طباعة عناصر القائمة من الثاني حتى الرابع
- د- حذف العنصر الثاني في القائمة
- ذ- حذف Cherry من القائمة

## النصوص والقوائم:

### أولاً: النصوص:

يتم تعريفها بالطريقة العادية مع علامة التنصيص المنفردة single quotation او المزدوجة double quotation الكوتيشن المنفرد أو المزدوج

```
a = "sweet home alabama"
```

```
a = 'sweet home alabama'
```

- للوصول إلى حرف معين نستخدم الأقواس المربعة (رمز القائمة) ونحدد رقم الحرف ال1ي نرغب في الوصول إليه حيث تبدأ الأرقام من 0 ، 1 ، 2 وهكذا بحيث يكون أول حرف رقمه index يساوي صفر

```
a[0] , a[3] , a[7]
```

- لتكرار النص عدد مرات معينة نضربه في ذلك العدد، مثال 3 مرات

```
a*3
```

- للوصول إلى جزء من النص ولنفرض من الحرف الخامس للثاني عشر

```
a[4:12]
```

- للوصول إلى جزء من النص ولنفرض من البداية للثاني عشر

```
a[:12]
```

- للوصول إلى جزء من النص ولنفرض من الحرف الخامس للنهاية

```
a[4:]
```

- للوصول إلى كل النص

```
a[:]
```

- لتحويل النص إلى قائمة list فيها كل الحروف لوحدها , بما فيها المسافة  
`list(a)`
- لتحويل النص إلى قائمة list فيها كل الحروف لوحدها , ولكن مع ترتيب  
`sorted(list(a))`
- لتحويل النص إلى قائمة list فيها كل الحروف لوحدها لكن بدون تكرار , يعني a لو متكررة 5 مرات يتم إحضارها مرة واحدة  
`set(x)`
- لتقسيم النص على كلمات وليس حروف  
`a.split()`
- لتقسيم النص حسب رقم أو عدد معين بحيث يكون ما داخل الأقواس هو الفاصل:  
`x = "12354785669854412503665"`  
`x.split("5")`
- لتقسيم النص إلى أسطر بحيث يكون كل سطر عبارة عن جملة  
`a.splitlines()`
- لتبديل حرف بحرف آخر:  
`a.replace('f','i')`
- لتبديل جزء من النص بجزء آخر : لاحظ ان النص الأصلي لا يتغير, هو يظهر النتيجة و انت احفظها  
`a.replace('sweet','ugly')`
- لمعرفة حرف معين مثال m تكرر كم مرة في الجملة أو لمعرفة تكرار نص معين  
`a.count('m')`
- يجعل اول حرف في الجملة كابيتال والباقي سمول

`a.capitalize()`

- يجعل اول حرف في كل كلمة كابيتال

`a.title()`

- يجعل كل الحروف كابيتال

`a.upper()`

- يعكس , اي حرف كابيتال يجعلها سمول و العكس

`a.swapcase()`

- يعمل مسافات يمين و شمال الكلام , بحيث يكون مجموع السطر بالمسافات بالكلام 30 خطوة

`a.center(30)`

- هل كل الحروف المستخدمة الفابيتيك أي أحرف أبجدية, لو فيها رموز زي (%@+) يعطي false غير ذلك يعطي true

`a.isalpha()`

- معناها حذف اي مسافات قبل او بعد الكلام

`a.strip()`

- هل كل السترنج (النص) أرقام (هذه ال function تعطينا True أو False ) وكذلك جميع ال Functions التي تبدأ بكلمة is

`a.isdigit()`

- هل كل السترنج (النص) حروف كابيتال

`a.isupper()`

- هل كله سمول

`a.islower()`

- هل اول حرف بس كابيتال

`a.istitle()`

- هل اخر كلمة في a هي كلمة كذا (true or false)

`a.endswith('alabama')`

- هل اول كلمة هي كلمة كذا (true or false)

`a.startswith('sweet')`

- يقوم امر (جوين) باضافة ما قبله لما بعده

`a = ', '.join( ('one', 'two', 'three') )`

- عمل مسافات بين كل حرف و الثاني

`' '.join('hello')`

## ثانياً: القوائم

- ما هي :

- مجموعة من البيانات التي يتم تخزينها بترتيب محدد
- قد يكون بها انواع مختلفة من البيانات مثل :

- نصوص String
- أرقام Integer , Float , Complex
- بوليان Boolean
- لا يستلزم تحديد نوعها بشكل مسبق
- يمكن دمج اكثر من نوع معا في القائمة
- يمكن للقائمة ان يتم تعديلها لاحقا , ويتم إضافة بيانات

- الأنواع المختلفة :

■ القائمة List

- يتم التعامل معها بالأقواس [ ]

- تشمل أنواع مختلفة من البيانات , ويمكن تعديلها لاحقا
- tuple الصفوف
  - يتم التعامل معها بالأقواس ( )
  - تشمل بيانات غير قابلة للتغيير
- المجموعة Set
  - يتم التعامل معها بالأقواس { }
  - لا تقبل التكرار
- القاموس Dictionary
  - يكون شكلها {'a':1, 'b':2, 'c':3}
  - تشمل علي دليل و مفاتيح و قيم

### تفاصيل List:

- يتم تعريفها عن طريق كتابة اسمها و عمل اقواس مربعة فارغة
- `L = [ ]`
- يمكن أن نضيف عليها ارقام بالطريقة التالية
- `s = [1,2,3,6,9,8,7]`
- ممكن القائمة تحتوي على عناصر من انواع مختلفة
- `s = [ "Majd" , 5 , 3.2 , 5+2j ]`
- ممكن أن نعمل قائمة في داخل القائمة nested lists نسميها كما في المثال:
- `s = [ 2 , 3, [ 4 , 5 ] ]`
- يمكن تحويل النص String إلى قائمة List بحيث تكون عناصر القائمة هي كل حرف من حروف النص بما فيها المسافة أو الفراغ

`x = list("love apple")`

مراجع إضافية Extra Resources

[Python Strings \(w3schools.com\)](https://www.w3schools.com/python/strings/)

[Python - Slicing Strings \(w3schools.com\)](https://www.w3schools.com/python/python_strings_slicing.php)

[Python - Modify Strings \(w3schools.com\)](#)

[Python - String Methods \(w3schools.com\)](#)

[Python Lists \(w3schools.com\)](#)

[Python - Change List Items \(w3schools.com\)](#)

[Python - Add List Items \(w3schools.com\)](#)

[Python - Remove List Items \(w3schools.com\)](#)