# البرمجة و الذكاء الإصطناعي 2

السنة 1 ماستر - علم الإجتماع

جامعة سطيف 2

الأستاذة: د.حياة دحمري

جوان 2026

# قائمة المحتويات


**I-تحليل النصوص في Python** 5

آ. مقدمة.....5

ب. تحليل النصوص باستخدام Python.....5

1. بعض دوال بايثون المستخدمة في تحليل النصوص.....6

پ. مكتبات تحليل النصوص في Python.....6

1. استخدام المكتبات.....7

ت. مراحل تحليل النصوص.....7

قاموس 9

قائمة المراجع 11

مراجع الأنترنيت 13

# I تحليل النصوص في Python


مقدمة 5
تحليل النصوص باستخدام Python 5
مكتبات تحليل النصوص في Python 6
مراحل تحليل النصوص 7


## آ. مقدمة

يمثل بايثون (Python) لغة برمجة عالية المستوى، مجانية، مفتوحة المصدر، و مفسرة. تعتبر الخيار الأول للمبتدئين نظرا لسهولة تعلمها واستخدامها وتوفرها على عدد كبير من المكتبات الجاهزة في مختلف المجالات.
تُستخدم على نطاق واسع في تطوير الويب، الذكاء الاصطناعي، وغيرها. كما أنها من أشهر لغات البرمجة المستعملة في تحليل النصوص و البيانات حيث تحتوي على مكتبات قوية لتحليل النصوص.[1]

*أيقونة بايثون*

## ب. تحليل النصوص باستخدام Python

تحليل النصوص (*Text Analysis*) هو عملية معالجة النصوص، استخراج المعلومات منها ، وفهم محتواها باستخدام خوارزميات وبرامج حاسوبية. وتعتبر Python من أشهر لغات البرمجة في تحليل البيانات حيث تحتوي على مكتبات قوية لتحليل النصوص.[2]
يساعد تحليل النصوص الباحثين في:

- تحليل محتوى المقابلات
- تحليل تعليقات مواقع التواصل الاجتماعي
- دراسة الرأي العام
- تحليل الخطاب الإعلامي

- تحليل الاستبيانات المفتوحة

مثال: بدلاً من قراءة 1000 تعليق على فيسبوك يمكن للبرنامج استخراج:

- أكثر الكلمات استعمالًا
- الموضوعات الرئيسية
- المشاعر (إيجابي / سلبي)

## 1. بعض دوال بايثون المستخدمة في تحليل النصوص

من أشهر الدوال التي تستخدمها لغة بايثون في تحليل النصوص نذكر:[3]

1- الدالة ***len***: تقوم بحساب عدد الأحرف.

مثال:

```
text = "I start with data analysis"
print(len(text))

#result: 26
```

2- الدالة ***lower***: تقوم بتحويل النص إلى حروف صغيرة.

مثال:

```
text = "HELLO PYTHON"
print(text.lower())

#result: hello python
```

3- الدالة ***split***: تقوم بتقسيم النص إلى كلمات.

مثال:

```
text = "Data analysis is important"
words = text.split()
print(words)

#result: ['Data', 'analysis', 'is', 'important']
```

4- الدالة ***replace***: تقوم باستبدال جزء من النص بجزء آخر .

مثال:

```
text = "she is intelligent"
text = text.replace("she", "he")
print(text)


#result: he is intelligent
```

5- الدالة ***len*** بعد استخدام ***split***: تقوم بحساب عدد الكلمات.

مثال:

```
text = "Data analysis is important"
words = text.split()
print(len(words))

#result: 4
```

# ب. مكتبات تحليل النصوص في Python

المكتبة (*Library*) هي مجموعة من الأدوات الجاهزة التي تساعد المبرمج على إنجاز مهام معينة

بسرعة. بدل كتابة كل شيء من البداية، يمكن استعمال المكتبات.

ومن أهم مكتبات تحليل النصوص في بايثون:[5]

1. *NTLK*: اسمها الكامل *Natural Language Toolkit* وهي مكتبة متخصصة في:
   - تحليل اللغة الطبيعية
   - تقسيم النصوص
   - تحليل الكلمات
   - استخراج الكلمات المهمة
2. *spaCy*: مكتبة حديثة وسريعة تستخدم في:
   - تحليل الجمل
   - التعرف على الأسماء (أشخاص، أماكن...)
   - تحليل بنية الجملة
3. *Gensim*: مكتبة متخصصة في:
   - اكتشاف المواضيع داخل النصوص
   - تحليل الوثائق الكبيرة
4. *Pandas*: مكتبة تستخدم في:
   - تحليل البيانات
   - تنظيم البيانات في جداول
   - معالجة ملفات Excel و CSV

ويمكن استعمالها لتنظيم النصوص وتحليلها.

### 1. استخدام المكتبات

قبل استعمال أي مكتبة لأول مرة نقوم بتثبيتها في موجه الأوامر(*Invite de commandes/Command Prompt*) ، مثلا لتثبيت مكتبة *ntlk* نكتب : *pip install nltk* ثم قبل كل استعمال نستورد المكتبة في بايثون:

*import nltk*

*nltk.download('punkt')*

في الإصدارات الحديثة لبايثون نضيف أيضا: (*nltk.download('punkt_tab')*

مثال لاستخدام الدالة ***tokenize*** و التي تقوم بنفس دور الدالة *split* و لكن بدقة أكثر (*split* تقسم النص حسب الفراغ فقط. لذلك تبقى علامات الترقيم مرتبطة بالكلمات، بينما *tokenize* تفصل الكلمات، الفاصلة، و النقطة)

```
1  import nltk
2  nltk.download('punkt')
3  nltk.download('punkt_tab')
4
5  text = "You know! Data analysis is important."
6  words = nltk.word_tokenize(text)
7  print(words)
8
9
10 #result: ['You', 'know', '!', 'Data', 'analysis', 'is', 'important',
   '.']
```

## ت. مراحل تحليل النصوص

تمر عملية تحليل النصوص بعدة مراحل:

*مراحل تحليل النصوص*

1.جمع النصوص: يمكن الحصول على النصوص من:
   - المقالات

-الكتب
-مواقع الإنترنت
-وسائل التواصل الاجتماعي
-الاستبيانات
مثال: تحليل تعليقات الطلبة حول التعليم الإلكتروني.
2.تنظيف النصوص (*Text Cleaning*): النصوص الخام غالبًا تحتوي على:
-علامات ترقيم
-رموز
-روابط
-كلمات غير مفيدة
لذلك يجب تنظيف النص.
مثال:
النص الأصلي: ... *Hello !!! This is a text*
بعد التنظيف: *hello this is a text*
3.تقسيم النصوص (*Tokenization*): تعني تقسيم النص إلى كلمات أو جمل.
مثال:
النص: « *Data analysis is important* »
بعد التقسيم: *importanat - is - analysis - Data*
4.إزالة الكلمات الشائعة: هناك كلمات تتكرر كثيرًا لكنها لا تحمل معنى مهمًا مثل:
-the
-is
-and
-of
و تسمى *Stop Words* ويتم حذفها لتحسين التحليل.
5.استخراج الكلمات المهمة: بعد تنظيف النص يمكن استخراج:
-الكلمات الأكثر تكرارًا
-الكلمات المفتاحية
-الموضوعات الرئيسية
مثال: النص: « *Students use technology in education and learning* ».
يقوم البرنامج ب:
1. تنظيف النص
2. تقسيم النص إلى كلمات
3. حذف الكلمات الشائعة
4. استخراج الكلمات المهمة
الكلمات المهمة: *students, technology, education, learning*

# قاموس

عالية المستوى

قريبة من لغة الانسان و اسهل في الفهم.

لغة مفسرة

يتم تنفيذ الكود سطرًا بسطر، مما يسهل اكتشاف الأخطاء.

مجانية

يمكن تحميل البرنامج واستعماله دون دفع أي مبلغ مالي أو شراء ترخيص.

مفتوحة المصدر

يعني أن الكود البرمجي الداخلي للبرنامج متاح للجميع.

# قائمة المراجع


[1] أحمد منصور. (2020). مدخل إلى البرمجة بلغة بايثون للمبتدئين. دار شعاع، دمشق.

[2] Anthony Elliott. (2021). The Routledge Social Science Handbook of AI

[3] Severance, C. R. (2016). Python for Everybody: Exploring Data Using Python 3. CreateSpace Independent Publishing

# مراجع الأنترنيت

https://docs.python.org/3/library/index.html [5]