Université Constantine 2
جامعة قسنطينة 2

Web Analytics and Natural Language Processing (WANLP)

TP1

Prise en main des outils, d'IDE et des librairies à utiliser dans le module WANLP

Professeur BOURAMOUL Abdelkrim

Département IFA, Faculté NTIC

abdelkrim.bouramoul@univ-constantine2.dz

www.bouramoul.com

Etudiants concernés

Faculté/Institut	Département	Niveau	Spécialité
NTIC	IFA	Master 1	SDIA

Objectifs du TP1

Ce 1er TP vise à permettre aux étudiants de prendre en main les outils, les environnement et les librairies à utiliser dans le cadre du module WANLP

Présentation des outils et des environnements de développement nécessaires pour le module (Python, Jupyter Notebook et PyCharm).

Installation et configuration des bibliothèques Python essentielles pour le TALN et l'analyse web (NLTK, SpaCy).

Plan

Le langage Python

- Eléments clés et syntaxe

IDE pour Python

- Jupyter notebook
- PyCharm

Librairies Python pour le NLP

- TextBlob
- NLTK,
- spaCy
- Gensim
- Core NLP

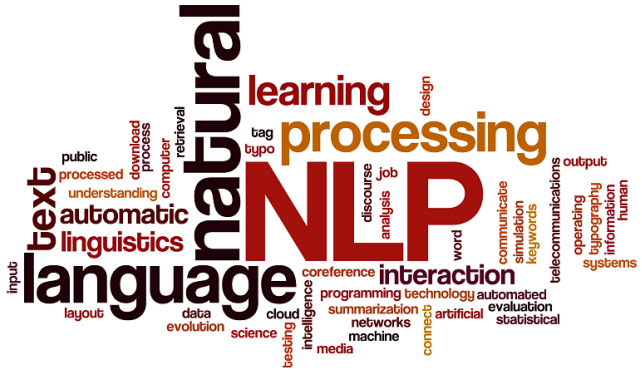
Exercices Pratiques

- Exercice 01
- Exercice 02
- Exercice 03

Le langage Python

Eléments clés du langage Python

- Python est un langage de programmation open source très répondu.
- Il s'est propulsé en tête de langages pour l'analyse de données et le domaine d'IA et d'NLP
- Il est très riche en terme de librairie, y compris celles pour l'NLP
- Il permet aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font.
- Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec d'autres langage. Ce qui a permis un développement plus rapide



Déclaration des variables et affichage sur console

```
x = 4
y = "Salam"
print(x)
print(y)
```

Les conditions

```
number = 4
if number == 0:
  print('Zero')
elif number < 0:
  print('Inférieur à Zéro')
else:
  print('Supérieur à Zéro')
```

Les boucles

```
words = ['Natural', 'Language', 'Processing']
for w in words:
    print(w, end=" ")
```

Importation des modules

```
import nltk
from textblob import TextBlob
```

IDE pour Python

(Integrated Development Environment)

Type d'IDE pour Python

Il y deux types d'IDE pour utiliser Python

IDE **interactif** pour un but éducatif comme **Jupyter Notebook**

- **Exploration et visualisation interactives**
- **Documentation et collaboration**
- **Prise en main facile**

IDE **standard** comme **PyCharm**

- **Editeur de code très riche**
- **Mode de débogage très avancés**
- **Gestion raffinée des projets Python (organisation des dossiers et fichiers)**

IDE - Jupyter

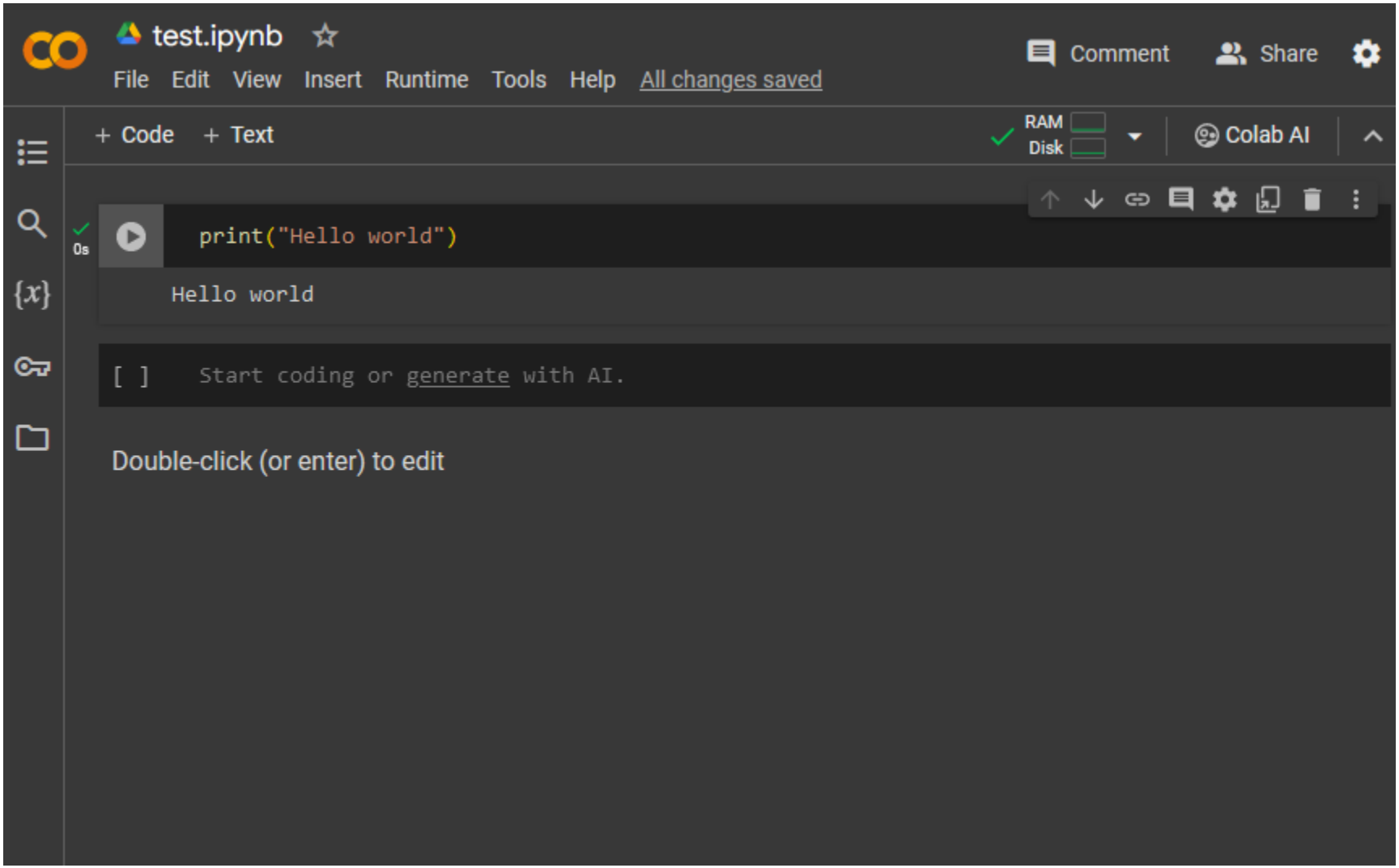
Principe de l'IDE Jupyter

- L'un des plus célèbre IDE pour développer avec la langage python
- C'est un IDE interactif basé sur le Web

Différents manière d'utiliser Jupyter

- Utiliser le site de Jupyter : https://jupyter.org/try
- Utiliser Google Colab : https://colab.research.google.com
- L'installer sur votre machine https://jupyter.org/install

Interface de l'IDE Jupyter sur Google Colab



Autres IDE

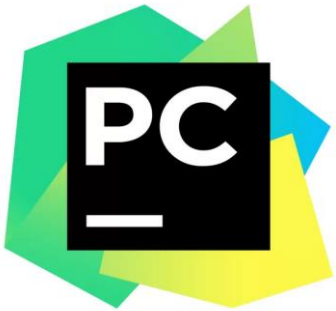
Il existe plusieurs autres IDE standard pour développer des projets avec Python, les plus utilisés sont :

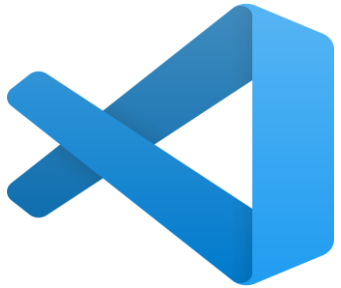
PyCharm

Version payante

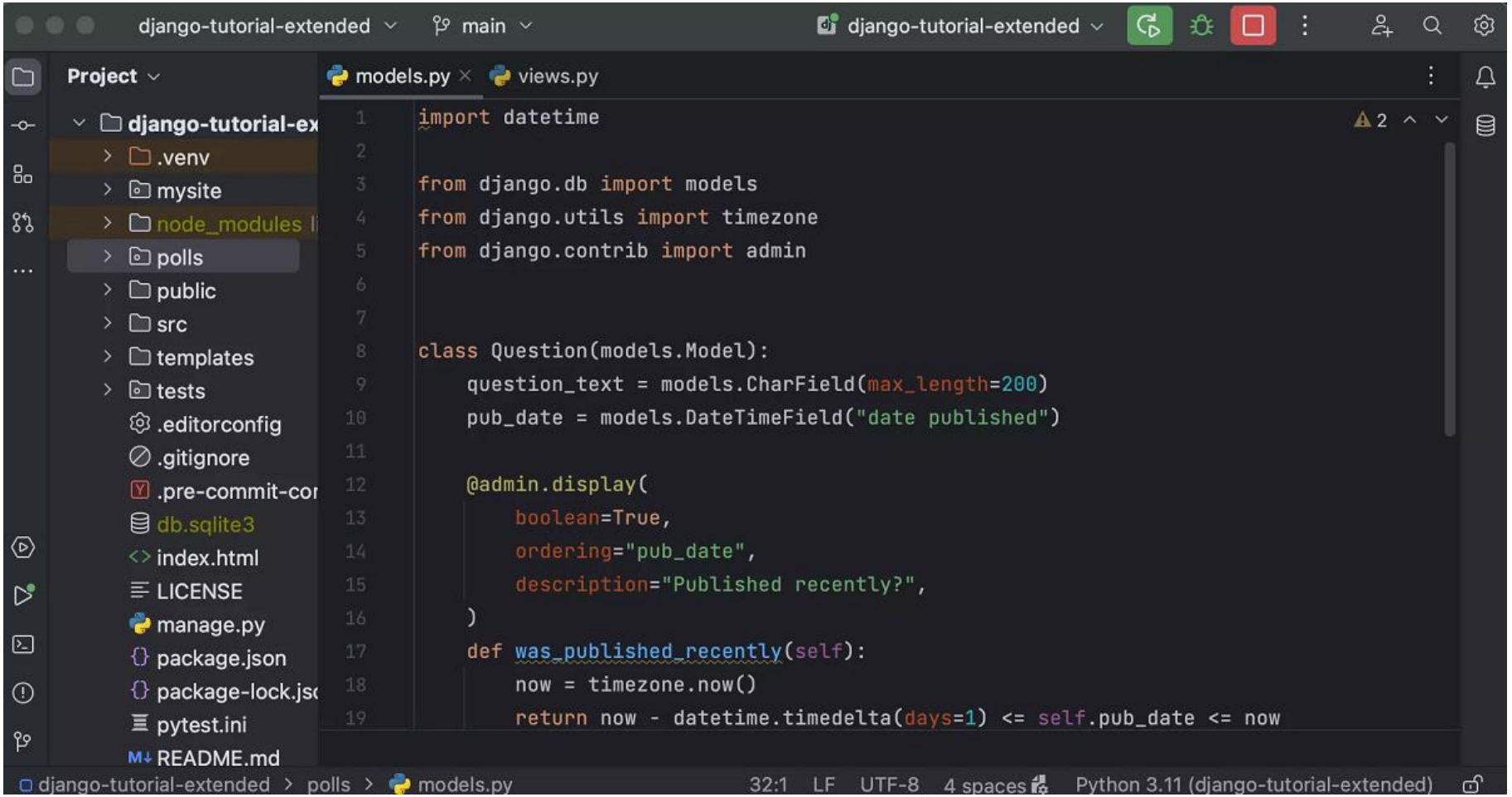
Version communauté (Gratuite) qui suffira largement

VSCode (avec installation des plugins)





Interface de l'IDE PyCharm

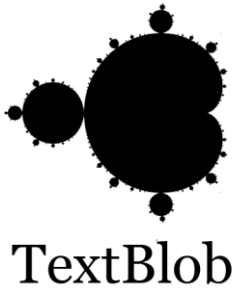


Librairies Python pour le NLP

Librairies Python pour le NLP

Python est riche en terme de librairies, il existe plusieurs librairies pour travailler avec du NLP :

- TextBlob
- NLTK
- spaCy
- Gensim
- Core NLP



Librairies pour le NLP

TextBlob

- Est une librairie Python qui propose une API simple pour le traitement automatique des langues (NLP).
- TextBlob propose plusieurs taches comme : Part of speech tags, sentiment analysis, correction d'orthographe etc …
- Lien de la librairie : https://textblob.readthedocs.io/en/dev/

NLTK

- Est une API très répandue pour le traitement des langues
- Elle propose une série de tâches (tokenization, stemming, tagging) ainsi qu'une collection de datasets (Punkt Tokenizer Models,Evaluation data from WMT15 et d'autres sur : https://www.nltk.org/nltk_data/)
- Lien de la librairie :

spaCy

- Est une librairie Python très puissante utilisée dans le domaine de recherche et le domaine de l'industrie.
- Propose plusieurs tâches à savoir : NER (Named Entity Recognition), Part Of Speech tagging, dependency parsing, word vectors

Librairies pour le NLP

Gensim

- Est une librairie python spécialisée dans le topic modeling mais propose d'autres tâches comme les words embedding.
- Lien de la librairie : https://radimrehurek.com/gensim/

Core NLP

- Est une librairie Java pour le traitement des langues de l'université Stanford. Propose des taches de parts of speech, named entities, numeric and time values.
- Compatible jusqu'à maintenant avec 8 langues: Arabe, chinois, anglais, français, allemand, hongrois, italien et espagnol

Exemple de l'utilisation de TextBlob avec Jupyter

TextBlob

```
[7] import nltk
    nltk.download('averaged_perceptron_tagger') # téléchargement des ressources nécessaires (tagger)

    from textblob import TextBlob
    wiki = TextBlob("Python is a high-level, general-purpose programming language.")
    wiki.tags
```

Sentiment Analysis

```
[8] testimonial = TextBlob("Textblob is amazingly simple to use. What great fun!")
    testimonial.sentiment
    testimonial.sentiment.polarity
```

```
0.39166666666666666
```

Spelling Correction

```
[20] b = TextBlob("I havv goood speling!") # to correct "havv"
     print(b.correct())
```

```
I have good spelling!
```

n-grams

```
[16] blob = TextBlob("Now is better than never.")
     print("ngrams=2")
     print(blob.ngrams(n=2))
     print("\nngrams=3")
     print(blob.ngrams(n=3))
```

```
ngrams=2
[WordList(['Now', 'is']), WordList(['is', 'better']), WordList(['better', 'than']), WordList(['than', 'never'])]

ngrams=3
[WordList(['Now', 'is', 'better']), WordList(['is', 'better', 'than']), WordList(['better', 'than', 'never'])]
```

Exemple de l'utilisation de NLTK avec Jupyter

NLP Libraries

NLTK

```
[19] import nltk # import
     nltk.download('punkt') # download necessary ressources
     sentence = """At eight o'clock on Thursday morning Arthur didn't feel very good.""" # sentence
     tokens = nltk.word_tokenize(sentence) # get tokens
     print(tokens) # print tokens
```

```
['At', 'eight', "o'clock", 'on', 'Thursday', 'morning', 'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Exercices Pratiques

Exercice 1

- Écrire un script Python qui permet de corriger l'orthographe d'un mot. (Utilisation de la librairie TextBlob pour les mots anglais)

Exercice 2

- Ecrire un script Python qui donne si le texte est positif ou négatif (affichage d'un texte positif/négatif et non pas un score).
- Utilisation de TextBlob, pour un texte en anglais

Exercice 3

- Écrire un programme Python qui vérifie d'abord mot par mot si le texte est correct et les corriger puis donner la polarité du texte après sa correction. il faut aussi afficher le nombre de correction effectué.

Fin du TP1