الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة المجاهد عبد الحفيظ بوالصوف ميلة
كلية العلوم القانونية والحقوق

ملخص مقياس

الذكـــــاء الإصطناعي

المحور الثاني :لغة البرمجة

الانسان مطالب أن يكون ذكي عند استخدام الذكاء الاصطناعي،
المستقبل يتطلب كوادر قادرة على التحكم في الذكاء الاصطناعي،
ورجال القانون أولى بفهم الذكاء الإصطناعي،
والذكاء الاصطناعي في خدمة القانون

من إعداد الأستاذ :

- عيبش توفيق .

موجه لطلبة :

- ماستر 1 قانون عقاري ؛
- ماستر 1قانون الإداري ؛
- ماستر1 قانون الجنائي والعلوم الجنائية ؛
- ماستر1 قانون الأعمال.

السنة الجامعية 2025 – 2026

المحور الثاني : لغة البرمجة Language Programming

محتوى المحور

- أولا : مدخل إلى البرمجة
- ثانيا : البرمجة بلغة Python
- ثالثا : التعرف على بيئة العمل ، تثبيث بايتون وكتابة أول برنامج بلغة Python
- رابعا : المتغيرات والعمليات
- خامسا : التعامل مع الملفات وقواعد البيانات
- سادسا : هندسة الأوامر وتطبيقاتها في البرمجة

الموقع الرسمي لبايثون : www.python.org

مقدمة

شهد العالم في العقود الأخيرة تحولاً رقمياً متسارعاً أعاد رسم ملامح جميع التخصصات، ولم يكن المجال القانوني بمنأى عن هذا التحول الجذري ،فمع تصاعد حجم البيانات القانونية وتراكم الأحكام القضائية والنصوص التشريعية، باتت الأدوات التقنية الحديثة ضرورة لا ترفاً أكاديمياً[1].

في هذا السياق، تبرز لغة البرمجة Python بوصفها الأداة الأكثر ملاءمةً لطالب الحقوق والباحث القانوني على حدٍّ سواء، نظراً لبساطة تركيبها وقربها من اللغة الطبيعية ووفرة مكتباتها المتخصصة في تحليل البيانات ومعالجة النصوص والذكاء الاصطناعي[2].

يتناول هذا المحور الثاني من مقياس الذكاء الاصطناعي جملةً من المفاهيم الأساسية، تبدأ بمدخل نظري إلى عالم البرمجة ولغاتها وأنماطها، مروراً بالتعريف بلغة Python وخصائصها وتطورها التاريخي، وصولاً إلى التطبيقات العملية في إدارة الملفات وقواعد البيانات القانونية وهندسة الأوامر.

تم توظيف أمثلة مستوحاة من الواقع القانوني الجزائري لتقريب المفاهيم التقنية من ذهن طالب الحقوق.

وإبراز الارتباط الوثيق بين القانون والتكنولوجيا

أولا : مدخل إلى البرمجة

في عصر التحول الرقمي، لم تعد البرمجة حكراً على المهندسين والتقنيين. أصبح المحامي والقاضي والباحث القانوني اليوم بحاجة ماسة إلى فهم أدوات التكنولوجيا الحديثة للتعامل مع الكميات الهائلة من الوثائق والنصوص القانونية والأحكام القضائية (3) .

1 – تعريف البرمجة :Programming :

هي كتابة التعليمات والأوامر وتخزينها في الذاكرة ، ليستدعيها المعالج لتوجيه الحاسوب بكيفية التعامل مع البيانات ، متبعة لقواعد محددة باللغة التي اختارها المبرمج لتنفيذها ولحل المشكلة ، كل لغة لها خصائص تميزها [4]..

2 – المبرمج :Programmer :

المبرمج هو الفرد الذي يمتلك المهارات اللازمة التي تؤهله لمعاينة مشكلة وتحليل بياناتها ، وكتابة التعليمات بلغة اصطلاحية لحل المشاكل.

3 – لغات البرمجة : Programming Language

لغات البرمجة مجموعة من القواعد والرموز تستعمل لكتابة التعليمات التي يفهمها الحاسوب بغرض انشاء برنامج حاسوبي (5) .

4 – أنماط لغات البرمجة : تُصنَّف لغات البرمجة تاريخياً إلى عدة أنماط بحسب مستوى تجريدها وقربها من اللغة الطبيعية:

- **البرمجة الإجرائية :** Procedural Programming نمط برمجة يعتمد على تقسيم البرنامج إلى إجراءات (Procedures) أو دوال تنفذ خطوة بخطوة، تركز على تسلسل الأوامر ، مناسبة للمشاكل البسيطة ،**مثال: لغة C.**
- **البرمجة الوظيفية : (Functional Programming)** نمط برمجة يعتمد على الدوال الرياضية النقية، حيث يتم التعامل مع البرنامج كتركيب لدوال نقية (Pure Functions) ، لا تعتمد على تغيير الحالة (Immutable State). ، تقلل الأخطاء وتُستخدم في التحليل والذكاء الاصطناعي ، مثال: Haskell.
- **البرمجة الموجهة للكائنات :** Object-Oriented Programming نمط برمجة يعتمد على الكائنات (Objects) التي تحتوي على بيانات (Attributes) وسلوك (Methods) ، مبادئها: التغليف، الوراثة، تعدد الأشكال. **مثال: Java و Python.**

5 - الخوارزمية(Algorithm)

الخوارزمية هي مجموعة من الخطوات المنطقية المتسلسلة لحل مشكلة معينة. يجب أن تكون الخوارزمية: واضحة ، محدودة و قابلة للتنفيذ (6).

مثال قانوني — خوارزمية مراجعة عقد

- قراءة بنود العقد كاملاً ؛
- تحديد الأطراف والتزاماتهم؛
- البحث عن البنود الغامضة ؛
- مقارنتها بالنصوص القانونية ؛
- إعداد تقرير الملاحظات.

6 - **اللغات القانونية** :تتكون من النصوص التشريعية والقانونية والتنظيمية

الفرق بين لغة البرمجة واللغة القانونية

الخاصية	اللغة القانونية	لغة البرمجة
المتلقي	انسان يفكر ويجتهد	الة تنفذ حرفيا دون اجتهاد
الدقة	دقيقة لكن تقبل التفسير القضائي	صارمة لا تقبل أي غموض أو تأويل
القواعد	مرنة وتتطور بالاجتهاد القضائي	ثابتة ومحددة بصرامة تامة
الغرض	تنظيم العلاقات بين الأشخاص	توجيه الحاسوب لتنفيذ مهام
التطور	يتطور عبر السوابق والتشريع	تُديره هيئات تقنية متخصصةiso IEEE
الخطأ	يعالج بالتفسير القضائي	يوقف تنفيد البرنامج كليا ERRor

المصدر : من اعداد الباحث بالاعتماد على مصادر الانترنت

ثانيا : البرمجة بلغة بايثون : Python :

تُعدّ Python من أكثر لغات البرمجة طلباً في الوقت الحالي، وإحدى أهم اللغات التي يعتمد عليها المهتمون بمجال الذكاء الاصطناعي وأمن المعلومات، إذ أصبحت الجسر الرابط بين عالم القانون وعالم الذكاء الاصطناعي.(7)

1 – نشأة لغة Python وتطورها التاريخي

لغة Python هي لغة برمجة عالية المستوى مفسَّرة وديناميكية الكتابة ، أبتكرها المبرمج الهولندي Guido van Rossumخلال عمله في مركز أبحاث رياضيات علوم الحاسوب (Centrum Wiskunde & Informatica – CWI)بهولندا، وذلك خلال الفترة الممتدة بين عامَي 1985 و1989، وأُعلن عنها رسمياً عام 1991. .(8) ويمكن إستخدامها لبناء برامج سطح المكتب, تطبيقات الويب, الألعاب, إلخ

الجدول رقم () : إهم إصدارات لغة بايثون

الخاصية	الوصف
1991	الإصدار الأول – Python 0.9.0 بداية المشروع
1994	Python 1.0 – إضافة الدوال الوظيفية(lambda, map, filter)
2008	Python 3.0 – إعادة بناء جذرية للغة
2020	إيقاف دعم Python 2.x رسمياً والتوجه الكامل نحو Python 3
2024+	Python 3.12+ – تحسينات الأداء ودعم البرمجة المتزامنة(async)

المصدر : من اعداد الباحث بالاعتماد على الانترنت

2 - خصائص ومميزات لغة بايثون Python caracteristics

تتميز لغة بايثون بعديد من المميزات:

- تعتبر لغة برمجة عالية المستوى، أي صممت لتكون واضحة نسبيا للبشر وللحواسيب ؛
- لغة سهلة التعلم وفعالة ، تعليماتها بسيطة، صياغتها بسيطة وسهلة القراءة ؛
- تحتوي على مجتمع واسع وعدد كبير من المكتبات ؛
- لغة موجهة للكائنات ، مجانية ومفتوحة المصدر ، ادارة تلقائية للذاكرة ؛

- **الجدول رقم ( ) : خصائص ومميزات لغة بايتون**

الخاصية	الوصف
عالية المستوى(High-Level)	تُجرِّد تفاصيل الآلة وتُقرِّب الكتابة من اللغة الإنجليزية الطبيعية.
مفسَّرة(Interpreted)	تُنفَّذ التعليمات سطراً بسطر عبر مُفسِّر، مما يُسهِّل الاختبار السريع.
ديناميكية الكتابة Dynamic Typing	لا يتطلب التصريح عن نوع المتغير مسبقاً.
كائنية التوجه(Object-Oriented)	تدعم المفاهيم الكائنية الكاملة كالوراثة والتغليف.
مفتوحة المصدر (Open Source)	متاحة مجاناً ويُطوِّرها مجتمع عالمي واسع تحت إشراف PSF.
متعددة المنصات Platform Cross	تعمل على Windows و macOS و Linuxوالأنظمة المدمجة ،
مكتبات ضخمة(Rich Libraries)	تضم أكثر من 450,000 حزمة على مستودع PyPI. و أدوات جاهزة لكل مهمة قانونية تخيلها
نحو واضح(Clean Syntax)	يعتمد المسافات البادئة (Indentation) بدلاً من الأقواس المعقوفة.

المصدر : من إلإعداد الباحث بالاعتماد على مواقع الانترنت

3 - مجالات توظيف لغةPython

يتمتع Python بتنوع استخدام استثنائي، مما يجعلها الخيار الأمثل في مجالات متعددة:

- الذكاء الاصطناعي والتعلم الآلي :(AI & Machine Learning) عبر مكتبات TensorFlow و PyTorchو scikit-learn.
- علم البيانات وتحليلها :(Data Science) باستخدام Pandas و NumPy وMatplotlib.
- أمن المعلومات والاختراق الأخلاقي :(Cybersecurity) أدوات الفحص والتحليل الجنائي الرقمي.
- تطوير تطبيقات الويب :(Web Development) إطارات Django و Flask وFastAPI.
- الحوسبة العلمية والنمذجة :(Scientific Computing) في الفيزياء والكيمياء والأبحاث الأكاديمية.
- أتمتة المهام والنصوص البرمجية.(Automation & Scripting)

4 – مجالات توظيف البرمجة في العمل القانوني

الأداة / المكتبة	التطبيق التقني	المجال القانوني
Python + pandas	قراءة وتنظيم الوثائق تلقائياً	إدارة القضايا والملفات
Python + docx	توليد نماذج عقود آلياً	صياغة العقود
Python + NLP	استخراج السوابق القضائية	البحث القانوني
Python + matplotlib	رصد الاتجاهات القضائية	تحليل الأحكام
Python + cryptography	حماية الملفات والوثائق السرية	أمن المعلومات
منصات إلكترونية API +	رفع الدعاوى والتبليغات رقمياً	التقاضي الإلكتروني

5 – أهمية Python في القضاء والقانون

تعد لغة بايثون (Python) أداة ذات اهمية بالغة في المجال القانوني، ويعتبر جسراً يربط بين القانون والتكنولوجيا، مما يتيح لرجال القانون تقديم خدمات أكثر كفاءة ومواكبة التطورات العصرية.

حيث تُستخدم في عدة جوانب ومنها :

- **أتمتة المهام المكتبية القانونية؛ Legal Automation :**

تساعد في أتمتة صياغة العقود، مراجعة المستندات وإدارة الالتزامات الروتينية، مما يوفر الوقت والجهد .

- **تحليل البيانات القانونية Legal Data Analytics :**

تمكن من تحليل كميات ضخمة العقود والوثائق بسرعة ودقة لاستخراج البنود الأساسية، وفهم الأنماط، وتقييم المخاطر بشكل أسرع وأدق من الطرق التقليدية.

- **البحث القانوني الذكي : Legal Research :**

تُستخدم في البحث في قواعد البيانات القانونية الضخمة لاسترجاع السوابق القضائية والتشريعات ذات الصلة بفعالية، بفضل قدراتها في معالجة اللغات الطبيعية.(NLP)

- **التنبؤ بنتائج القضايا** (Predictive Analytics) :

بفضل خوارزميات التعلم الآلي، يمكن لبايثون المساعدة في التنبؤ بقرارات المحاكم بناءً على بيانات القضايا السابقة من خلال تحليل سلوك القضاة والمحامين والخبراء .

- **تحليل سلوك القضاة والمحامين** والموثقين ،وتحليل مشاعرهم والتنبؤ بالاتجاهات ..
- **تقليل الأخطاء البشرية وخفض التكاليف التشغيلية للمكاتب القانونية وتسريع الإجراءات**

ثالثا : التعرف على بيئة العمل ، تثبيث بايتون وكتابة أول برنامج بايتون

1 - بيئة التطوير المتكاملة (IDE)

Integrated Development Environments (IDEs)

تعد بيئة التطوير المتكاملة (IDE) برنامجا يستخدم من قبل المبرمجين لتطوير البرمجيات، حيث تجمع مجموعة من الأدوات الأساسية داخل واجهة مستخدم رسومية (GUI) واحدة، مما يسهل عملية البرمجة وزيادة الإنتاجية [9]

1 – 1 - مكونات بيئة التطوير المتكاملة IDE: تشمل هده المكونات عادة :

المكوّن	وظيفته
محرر الكود source Text Editor	مساحة كتابة التعليمات مع تمييز الألوان للتوضيح
مفسّر Python Interpreter	يترجم الكود إلى لغة الآلة وينفذه فور الطلب
مصحح الأخطاء Debugger	يكشف الأخطاء البرمجية ويساعد في إصلاحها
سطر الأوامر Command Line Interface	يتيح تنفيذ الأوامر مباشرة والتفاعل مع النظام

1 – 2 أمثلة على بيئات التطوير

- **Spyder**

هي بيئة تطوير متكاملة مفتوحة المصدر، موجهة أساسًا لعلوم البيانات والتحليل باستخدام لغةPython توفر واجهة تفاعلية وغالبًا ما تُستخدم ضمن حزمة Anaconda .

- **Jupyter Notebook**

هو تطبيق ويب مفتوح المصدر يتيح إنشاء ومشاركة مستندات يُستخدم بشكل واسع في تنظيف البيانات ، تحليل البيانات وتعلم الآلة .

ملاحظة : الادوات مثل **Spyder** و**Jupyter Notebook** تُعد أساسية في مجال علم البيانات وتعلم الآلة، نظرًا لقدرتها على الدمج بين البرمجة والتحليل الرياضي والإحصائي.

1 - 3 خيارات بيئة العمل لطالب الحقوق

المميزات	الأداة	النوع
مجاني – لا يحتاج تثبيتاً – يعمل على أي جهاز	Google Colab	عبر المتصفح) للمبتدئين(
بسيط وسريع – مثالي للتجربة الأولى	online-python.com	عبر المتصفح) للمبتدئين(
محرر احترافي مجاني – قابل للتخصيص	VS Code + Python	على الجهاز) للمتقدمين(
حزمة متكاملة لعلم البيانات والتحليل	Anaconda (Spyder/Jupyter)	على الجهاز) للمتقدمين(

2 - تثبيت بايثون وتشغيل المحرر (انظر النشاط رقم 01 في منصة Elearning)

https://elearning.univ-mila.dz/a2026/mod/resource/view.php?id=68479

2 - 1 تحميل وتثبيت بايثون : يمكن تثبيت لغة Pythonبسهولة من الموقع الرسمي، حيث تتوفر نسخ متوافقة مع مختلف أنظمة التشغيل

يتم ذلك عبر الموقع الرسمي : **www.python.org**

يمكن تشغيل بايثون بعد التثبيت باستخدام :

- الواجهات الرسومية (Graphical User Interface–GUI) عبر بيئات التطوير (IDE)
- سطر الأوامر (Command Line Interface – CLI) لتنفيذ الأوامر مباشرة

وللتحقق من التثبيت عبر سطر الأوامر بكتابة python –version

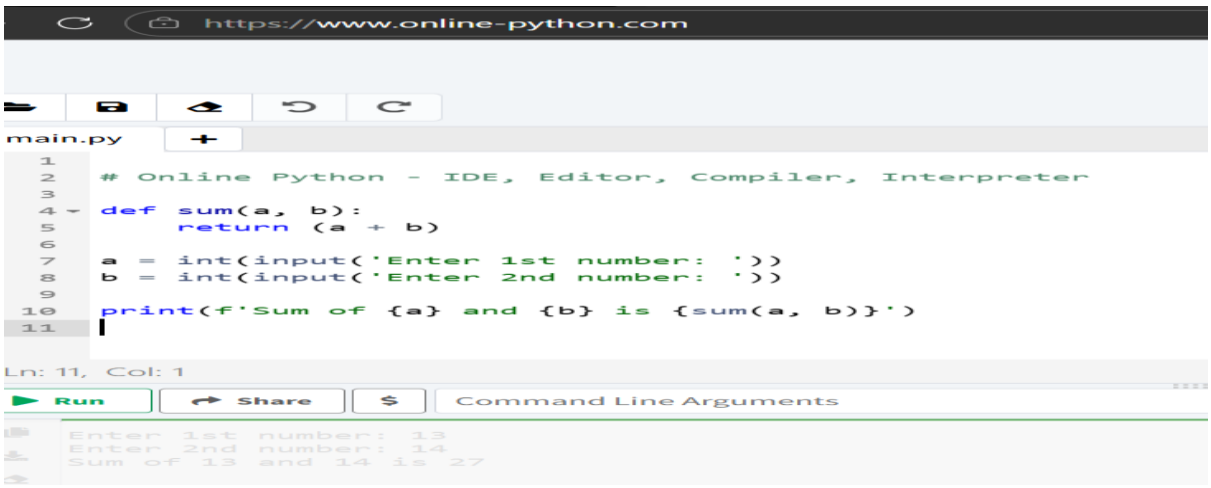
2 – 2 البرمجة عبر الانترنت (Online Python)

يمكن أيضًا كتابة وتشغيل البرامج دون تثبيتPython ، من خلال منصات برمجية عبر الإنترنت، **مثل:** **https://www.online-python.com/**

تتيح هذه الأدوات: كتابة الشيفرة البرمجية مباشرة، تنفيذ البرامج (Run) واختبار النتائج بشكل فوري ،لا تحتاج إلى تثبيت أو إعداد مسبق ، تعمل على أي جهاز (حاسوب أو هاتف) ، مفيدة للتعليم ،يعد استخدام المنصات عبر الإنترنت مناسبا للمبتدئين أو عند عدم توفر بيئة عمل محلية، في المقابل ، يُفضل تثبيت Python محليًا للحصول على تحكم أكبر وإمكانيات أوسع في تطوير المشاريع.

3 - أول برنامج بسيط بلغة Python

https://www.online-python.com/



أول برنامج يحاكي سيناريو قانونيً — حساب مدة التقادم في القانون المدني(10):

```
====== #البرنامج الأول :حساب مدة التقادم======

#تعريف بيانات القضية
اسم_الموكل' = السيد أحمد بن علي'
تاريخ_الواقعة2020 =
مدة_التقادم # 10 = سنوات وفق القانون المدني
السنة_الحالية2024 =

#الحساب
السنوات_المنقضية = السنة_الحالية - تاريخ_الواقعة
متبقي = مدة_التقادم - السنوات_المنقضية

#عرض النتيجة
print('===تقرير مدة التقادم===')
print('الموكل', ':' اسم_الموكل)
print('السنوات المنقضية', ':السنوات_المنقضية' ,سنوات)'

if متبقي:0 >
print('الحالة :لا تزال القضية ضمن مدة التقادم')
print('المتبقي', ':متبقي' ,سنوات)'
else:
print('تنبيه :انتهت مدة التقادم!')
```

ملاحظة

Python 3 تدعم العربية في أسماء المتغيرات مما يجعل الكود أقرب إلى تفكير طالب الحقوق ويُسهِّل الفهم..

رابعا : المتغيرات والعمليات

1 – المتغيرات

المتغير هو مساحة في ذاكرة الحاسوب لها اسم وقيمة ونوع بيانات .يمكن تشبيه المتغير بملف في مكتب محامٍ :له اسم) رقم الملف(، ومحتوى) وقائع القضية(، ونوع) جنائي، مدني، تجاري (11).

```
====== #أنواع المتغيرات بأمثلة قانونية======

#عدد صحيح – (int) للأرقام والتواريخ
عدد_القضايا47 =
سنة_الحكم2023 =
مدة_العقوبة # 5 = بالسنوات

#عدد عشري – (float) للمبالغ المالية
مبلغ_التعويض # 150000.75 = دينار جزائري
نسبة_الغرامة25% # 0.25 =

#نص – (str) للأسماء والعناوين
اسم_المتهم' = محمد بن عمر'
نوع_الجريمة' = احتيال مالي'
رقم_الملف' = ج'2023/0472/

#منطقي – (bool) للحالات الثنائية
القضية_مفتوحةTrue =
تم_الاستئنافFalse =

#عرض المعلومات
print('رقم الملف رقم ',رقم_الملف(
print('اسم_المتهم |' ,الجريمة ,':نوع_الجريمة )
print('التعويض المطلوب ',':مبلغ_التعويض' ,دج')
```

2 – المتغيرات المركبة

تُستخدم المتغيرات المركبة لتخزين مجموعة من البيانات معاً، مما يناسب تنظيم الملفات القانونية المعقدة:

```
#القائمة – (list) قائمة المتهمين في قضية جماعية
المتهمون'] = أحمد سالم' ,'عمر بوعلي' ,'خالد مسعود['
print('عدد المتهمين)len(المتهمون((

#القاموس – (dict) ملف قضية كاملة
ملف_القضية{ =
' الرقم' : 'ج,'2023/0472/
' النوع' : 'جنائي,'
' المتهم' : 'محمد بن عمر,'
' التهمة' : 'احتيال مالي,'
' الحالة' : 'قيد النظر,'
' التعويض150000.75 : '
}
```

```
print('نوع القضية:', ملف_القضية['النوع'])
print('الحالة الراهنة:', ملف_القضية['الحالة'])
```

3 العمليات على المتغيرات

مثال قانوني	الرمز	نوع العملية
مبلغ_التعويض ← 1.25 * إضافة 25% غرامة	+ - * / %	الرياضية
مدة_العقوبة ← 3 > هل تتجاوز 3 سنوات؟	== != > < >= <=	المقارنة
مفتوحة and لم_يُستأنف ← قابلة للطعن؟	and or not	المنطقية
عدد_الجلسات ← 1 =+ إضافة جلسة جديدة	= += -= *=	التخصيص

منطق القرار القانوني

4 - الجمل الشرطية — محاكاة القرار القضائي

تُستخدم الجمل الشرطية لتنفيذ كود مختلف بناءً على شرط معين — تماماً كما يتخذ القاضي قراراً بناءً على وقائع القضية وأحكام القانون.

```
# ====== تصنيف الجرائم وفق العقوبة ======

def تصنيف_الجريمة(مدة_السجن):
    if مدة_السجن == 0:
        return 'مخالفة – غرامة مالية فقط'
    elif مدة_السجن <= 2:
        return 'جنحة بسيطة'
    elif مدة_السجن <= 10:
        return 'جنحة خطيرة'
    elif مدة_السجن <= 20:
        return 'جناية – محكمة الجنايات'
    else:
        return 'جناية كبرى – عقوبة مشددة'

# الدالة اختبار
print(تصنيف_الجريمة(0))     # مخالفة
print(تصنيف_الجريمة(1))     # جنحة بسيطة
print(تصنيف_الجريمة(15))    # جناية
```

5 – الحلقات — معالجة ملفات القضايا

تُمكّن الحلقات من تكرار عملية على مجموعة من البيانات، مما يوفر وقتاً هائلاً في معالجة الملفات القانونية المتعددة:

```
# ====== المعلقة القضايا ملفات مراجعة ======

معلقة_قضايا = [
    {'الرقم': 'م/001', 'المدعي': 'الأمل شركة', 'المبلغ': 50000},
    {'الرقم': 'م/002', 'المدعي': 'علي بن أحمد', 'المبلغ': 120000},
    {'الرقم': 'م/003', 'المدعي': 'سالم مصطفى',  'المبلغ': 35000},
]

print('=== المعلقة القضايا تقرير ===')
المجموع = 0

for قضية in معلقة_قضايا:
    print(f"الملف {قضية['الرقم']} | {قضية['المدعي']} | {قضية['المبلغ']}
دج")
    المجموع += قضية['المبلغ']

print('بها المطالب المبالغ إجمالي:', المجموع, 'دج')
```

خامسا : التعامل مع الملفات وقواعد البيانات القانونية

1 - الملفات في Python — (الوثائق القانونية رقمياً)

الملف في البرمجة هو وحدة تخزين تحتوي على بيانات محفوظة على القرص الصلب

يتكوّن مسار الملف من :اسم المجلد + اسم الملف + الامتداد.

مثل (Acts/Act_001.txt) :

الملفات النصية تدعم ترميز UTF-8 الذي يدعم اللغة العربية[12].

يشير امتداد الملف الى نوع الملف ، ويساعد على تحديد البرنامج المناسب لفتح الملف .

```
# ====== قانونية وثيقة وقراءة إنشاء ======

# عقد ملف وكتابة إنشاء
with open('عقد_إيجار_001.txt', 'w', encoding='utf-8') as ملف:
    ملف.write('إيجار عقد\n')
    ملف.write('علي بن أحمد السيد :المؤجر\n')
    ملف.write('بوعلي فاطمة السيدة :المستأجر\n')
    ملف.write('العقد مدة: 12 شهراً\n')
    ملف.write('الشهري الإيجار: 25000 دج\n')

print('بنجاح الوثيقة إنشاء تم')

# الملف قراءة
with open('عقد_إيجار_001.txt', 'r', encoding='utf-8') as ملف:
    محتوى = ملف.read()
    print('العقد محتوى:')
    print(محتوى)
```

2 – 2 ـ التعامل مع قواعد البيانات (قواعد البيانات القانونية)

تنقسم قواعد البيانات حسب بنيتها إلى :

- **قواعد البيانات المهيكلة(Structured Databases)**

تُخزَّن فيها البيانات ضمن جداول منظمة تحتوي على صفوف وأعمدة، وتعتمد على لغة **SQL**للاستعلام والتحكم في البيانات. تُستخدم عندما تكون البيانات رقمية أو نصية بسيطة ومتجانسة، مثل نتائج الاستبيانات أو سجلات الطلبة.

أمثلةMySQL ،: PostgreSQL ،SQL Server ،Microsoft Access ،SQLite.

مزاياها :سرعة الاستعلام، الدقة، ضمان تكامل البيانات.

- **قواعد البيانات غير المهيكلة(Unstructured / NoSQL Databases)**

تُستخدم لتخزين بيانات غير منتظمة أو متنوعة الأشكال، مثل الصور والفيديوهات والنصوص الطويلة والردود المفتوحة في الاستبيانات النوعية.

أمثلةMongoDB ،: Elasticsearch ،Firebase ،Cassandra.

مزاياها :مرونة عالية، قابلية التوسع، مناسبة للبيانات الضخمة.(Big Data)

- **قواعد البيانات شبه المهيكلة**

نوع وسيط يجمع بين النوعين، مثل ملفات **JSON**و**XML**، وتُستخدم كثيراً في تبادل البيانات بين التطبيقات والبرامج الإحصائية.

2 – 2 : أهم برامج قواعد البيانات

البرنامج	الاستخدام الأمثل	المستوى
Microsoft Access	البحوث الصغيرة والتعليمية	مبتدئ
Google Sheets / Excel	تخزين وتحليل البيانات البسيطة	مبتدئ
SQLite	التطبيقات الأكاديمية والمشاريع البرمجية	متوسط
MySQL / PostgreSQL	المشاريع الأكاديمية الكبيرة	متقدم
MongoDB	البيانات النوعية والضخمة	متقدم

قواعد البيانات ، لكلٍّ منها استخدام محدد في العمل القانوني:

الاستخدام القانوني	الأمثلة	النوع
سجلات المحاكم، قوائم الموكلين، أرشيف الأحكام	MySQL, SQLite, PostgreSQL	مهيكلة(SQL)
الوثائق النصية الطويلة، الاتفاقيات المعقدة	MongoDB, Firebase	غير مهيكلة (NoSQL)
تبادل البيانات بين منصات التقاضي الإلكتروني	JSON, XML	شبه مهيكلة

3. قاعدة بيانات للمحكمة — مثال تطبيقي

```
import sqlite3

# إنشاء قاعدة بيانات المحكمة
conn = sqlite3.connect('المحكمة.db')
cursor = conn.cursor()

# إنشاء جدول القضايا
cursor.execute('''
    CREATE TABLE IF NOT EXISTS القضايا (
        id          INTEGER PRIMARY KEY,
        رقم_الملف  TEXT,
        المدعي      TEXT,
        المدعى_عليه TEXT,
        النوع       TEXT,
        الحالة      TEXT
    )
''')

# إضافة قضية جديدة
cursor.execute('INSERT INTO القضايا VALUES (?,?,?,?,?,?)',
    (1, 'م/2023/001', 'أحمد علي بن', 'شركة البناء', 'مدني', 'قيد النظر'))
conn.commit()

# استعلام عن القضايا المفتوحة
cursor.execute('SELECT * FROM القضايا WHERE الحالة = ?', ('قيد النظر',))
for قضية in cursor.fetchall():
    print(قضية)
conn.close()
```

النشاط 2 : المطلوب من الطالب الولوج الى هده المواقع وإجراء عدة محاولات على العمليات :

https://Blockly-demo.appspot.com

https://www.online-python.com

سادسا : هندسة الأوامر وتطبيقاتها في البرمجة Prompt engineering

1 – **هندسة الأوامر** (Prompt Engineering) تُعدّ هندسة الأوامر فرعًا حديثًا ضمن مجال الذكاء الاصطناعي، وتهدف إلى تصميم وصياغة التعليمات الموجهة إلى النماذج الذكية بطريقة منهجية للحصول على نتائج دقيقة وموثوقة.

وهي مهارة متقدمة تساعدك على كتابة توجيهات (Prompts) ذكية لتحصل على أفضل إجابات وتحليلات من نماذج الذكاء الاصطناعي مثل ChatGPT و Google Bard و Claude AIوغيرها.

وهي عملية تحسين المدخلات النصية (Prompts) لتحقيق أفضل استجابة ممكنة من **نماذج اللغة الكبيرة**.

2 – **أهمية هندسة الأوامر** تكمن أهميتها في :تحسين دقة النتائج ؛ تقليل الأخطاء ؛ التحكم في أسلوب الإجابة ؛ تسريع إنجاز المهام البرمجية .

تشير الدراسات الحديثة إلى أن جودة الـ Prompt تؤثر مباشرة على جودة المخرجات، حتى دون تغيير النموذج نفسه.

2 – **مكونات الامر الجيد** : لصياغة أمر فعّال، يجب أن يتضمن:

الوصف	مثال تطبيقي
تحديد المجال والتخصص	"في مجال القانون الجزائري"...
تحديد هوية النموذج المطلوبة	"كونك محامياً متخصصاً في العقود"...
ما المطلوب بالضبط	"اشرح الفرق بين الجنحة والجناية"...
النص أو البيانات المُراد معالجتها	نص عقد أو نص حكم قضائي
صيغة الرد المطلوبة	"في شكل نقاط مرقمة" أو "جدول مقارن"

تطبيقات هندسة الأوامر في البحث القانوني

- استخراج البنود الجوهرية من العقود الطويلة
- تلخيص الأحكام القضائية والاجتهادات
- المقارنة بين نصوص تشريعية متعددة
- إعداد مذكرات قانونية أولية
- ترجمة وثائق قانونية من وإلى اللغة العربية

بعد الحصول على النتيجة ينبغي تحليل وتقييم استجابات الذكاء الاصطناعي، تحسين دقة الأوامر، واكتشاف الأخطاء لتحصل على أفضل أداء ممكن.

تحسين صياغة التوجيه يمكن أن يزيد دقة إجابة الذكاء الاصطناعي بنسبة 40–60%

المصادر والمراجع

أولاً — المراجع التقنية

[1] Van Rossum, G. & Drake, F.L., Python 3 Reference Manual, Python Software Foundation, 2009. — المرجع الرسمي للغة Python. متاح على: docs.python.org

[2] Matthes, E., Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 3rd ed., No Starch Press, San Francisco, 2023.

[3] Lutz, M., Learning Python, 5th ed., O'Reilly Media, Sebastopol, 2013.

[4] McKinney, W., Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter, 3rd ed., O'Reilly Media, Sebastopol, 2022. [Scopus]

[5] Gaddis, T., Starting Out with Python, 5th ed., Pearson Education, New York, 2021. [Scopus]

[6] Cormen, T.H., Leiserson, C.E., Rivest, R.L. & Stein, C., Introduction to Algorithms, 4th ed., MIT Press, Cambridge, MA, 2022. [Web of Science / Scopus]

[7] Kreibich, J.A., Using SQLite: Small. Fast. Reliable. Choose Any Three, O'Reilly Media, Sebastopol, 2010.

ثانياً — مقالات علمية محكّمة في القانون والذكاء الاصطناعي

[8] Surden, H., «Machine Learning and Law», Washington Law Review, Vol. 89, No. 1, pp. 87–115, 2014. [HeinOnline / Web of Science]

[9] Katz, D.M., Bommarito, M.J. & Blackman, J., «A General Approach for Predicting the Behavior of the Supreme Court of the United States», PLOS ONE, Vol. 12, No. 4, e0174698, 2017. [Scopus / Web of Science

[10] Remus, D. & Levy, F., «Can Robots Be Lawyers? Computers, Lawyers, and the Practice of Law», Georgetown Journal of Legal Ethics, Vol. 30, pp. 501–558, 2017. [HeinOnline]

[11] Liu, P. et al., «Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing», ACM Computing Surveys, Vol. 55, No. 9, Article 195, 2023. [ACM Digital Library / Scopus]

[12] White, J. et al., «A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT», arXiv:2302.11382v1, 2023. [arXiv

[13] Doshi-Velez, F. et al., «Accountability of AI Under the Law: The Role of Explanation», Berkman Klein Center Research Publication, Harvard University, 2017. [SSRN

ثالثاً — مراجع عربية

[14] المطبوعات ديوان السادسة، الطبعة البحوث، إعداد وطرق العلمي البحث مناهج ،.م.م والذنيبات، .ع بوحوش، الجزائر، الجامعية، 2007.

[15] الإسكندرية، الجديدة، الجامعة دار الإلكترونية، والتجارة الرقمي القانون ،.ح.م منصور، 2019.

[16] Susskind, R. (ترجمة)، الغد محامو: مستقبلك إلى مدخل المهني، الطبعة الثانية، أكسفورد، 2017. النسخة الأصلية: Tomorrow's Lawyers, 2nd ed., Oxford University Press, 2017. [Scopus / Web of Science]

رابعاً — النصوص القانونية والمصادر الرسمية

[17] الأمر رقم 58-75 المؤرخ في 26 سبتمبر 1975 المتضمن القانون المدني الجزائري المعدل والمتمم] .الجريدة الرسمية للجمهورية الجزائرية الديمقراطية الشعبية[

[18] الموقع الرسمي للغة Python: https://www.python.org تاريخ الاطلاع: 2025.

[19] توثيق مكتبة pandas: https://pandas.pydata.org/docs تاريخ الاطلاع: 2025.

[20] مستودع حزم Python الرسمي — https://pypi.org (PyPI): تاريخ الاطلاع: 2025.