

# Vibe Coding

---

## 1. التعريف :

Vibe Coding هو عملية استخدام اللغة الطبيعية (Natural Language) لبرمجة الحاسوب عبر الذكاء الاصطناعي.

بدل كتابة:

if "فسخ" in النص:

print("تحذير")

نكتب:

"اكتب برنامج Python يحلل العقد ويكتشف كلمة فسخ."

والذكاء الاصطناعي يكتب الكود بدلاً منك.

---

## النموذج النظري الكامل:

يمكن تمثيل العملية كنظام:

Intent (النية): ما تريد بناءه.

↓

Prompt (التعليمات): الوصف النصي لما تريد.

↓

AI Model (النموذج)

↓

Generated Code (الكود الناتج)

↓

Evaluation (التقييم): هل الكود صحيح؟

↓

Refinement (التحسين): تحسين الـ Prompt للحصول على كود أفضل.

## لماذا Prompt Engineering مهم جدًا؟

لأن النموذج لا يفهم "النية" مباشرة، بل يفهم فقط "النص".

مثال:

**Prompt بدائي:**

"اكتب برنامج يحلل عقد."

**Prompt احترافي:**

"أنت مهندس برمجيات متخصص في Legal Tech. اكتب برنامج Python بسيط يساعد المحامي على تحليل نص عقد قانوني واكتشاف الكلمات الخطيرة مثل "فسخ" و "غرامة". استخدم دالة function واجعل الكود واضح."

الـ Prompt الثاني يحتوي على معلومات أكثر، لذلك الكود سيكون أفضل.

---

## كيف يعالج الذكاء الاصطناعي الـ Prompt؟

النموذج يقوم بثلاث عمليات:

1. تحليل الكلمات (Token Analysis)
2. تحديد السياق (Context Detection)
3. توليد الكود (Code Generation)

---

## البنية الكاملة لأي Prompt احترافي:

أي Prompt احترافي يحتوي على 6 عناصر:

1. الدور (Role)
2. المهمة (Task)
3. السياق (Context)
4. القيود (Constraints)
5. شكل المخرجات (Output format)
6. أمثلة (Examples)

فيما يلي بعض الأمثلة تظهر أهمية إدراج العناصر السابقة في الـ Prompt:

---

## 1. Role - الدور: يحدد "من هو الذكاء الاصطناعي".

مثال:

"أنت مهندس برمجيات متخصص في Python وتحليل النصوص القانونية."

هذا يجعل النموذج يكتب كود أفضل.

مقارنة عملية:

**Prompt** بدائي:

"اكتب برنامج Python يحلل عقد."

الكود الناتج:

نص = input("ادخل نص: ")

if "فسخ" in نص:

print("يوجد فسخ")

هذا الكود:

● بسيط جدًا

● غير منظم

**Prompt** احترافي:

"أنت مهندس برمجيات متخصص في Legal Tech."

اكتب برنامج Python بسيط يساعد المحامي على تحليل نص العقد واكتشاف الكلمات الخطيرة

مثل "فسخ" و "غرامة".

الكود الناتج:

```
نص_العقد = input("يتم فسخ عقد الزواج ما لم يتسنى الصلح بين الطرفين")
كلمات_خطيرة = ["فسخ", "غرامة"]
for فسخ in كلمات_خطيرة:
    if فسخ in نص_العقد:
        print("تحذير: تم العثور على", فسخ)
```

## لماذا الكود الثاني أفضل؟

لأن Role جعل النموذج:

- يفهم السياق القانوني
- يستخدم قائمة كلمات
- يستخدم حلقة

---

## Task 2 - المهمة: هو ما الذي يجب أن يفعله البرنامج بدقة.

مثال:

**Prompt** بدائي:

"اكتب برنامج Python."

الكود الناتج:

```
print("مرحبا")
```

**Prompt** احترافي:

"اكتب برنامج Python يقوم بتحليل نص العقد واكتشاف كلمة "غرامة"."

الكود الناتج:

```
نص = input("ادخل نص العقد: ")
```

```
if "غرامة" in نص:
```

```
    print("تم العثور على غرامة")
```

```
else:
```

```
    print("لا توجد غرامة")
```

---

### 3. Context - السياق: يشرح لماذا نحتاج البرنامج.

مثال:

**Prompt** بدائي:

"اكتب برنامج يبحث عن كلمة."

الكود الناتج:

نص = input(ادخل نص العقد)

if "فسخ" in نص:

print("فسخ")

**Prompt** احترافي:

"اكتب برنامج Python يستخدمه محامي لتحليل العقد واكتشاف المخاطر القانونية."

الكود الناتج:

نص\_العقد = input("ادخل نص العقد: ")

مخاطر = ["فسخ", "غرامة", "محكمة"]

نتائج = []

for عنصر in مخاطر:

if عنصر in نص\_العقد:

نتائج.append(عنصر)

print("المخاطر الموجودة:", نتائج)

---

### 4. Constraints - القيود: تحدد كيف يجب كتابة الكود.

مثال:

**Prompt** بدائي:

"اكتب برنامج تحليل عقد."

الكود الناتج:

```
print("تم التحليل")
```

**Prompt** احترافي:

"اكتب برنامج Python لتحليل العقد باستخدام دالة function."

الكود الناتج:

```
def تحليل(نص):
```

```
    كلمات = ["فسخ", "غرامة"]
```

```
    for غرامة in كلمات:
```

```
        if غرامة in نص:
```

```
            print("تم العثور على: " + غرامة)
```

---

## 5. Examples - الأمثلة: تساعد النموذج على فهم المطلوب.

مثال:

**Prompt** بدائي:

"اكتب برنامج يصنف النص."

الكود الناتج:

```
print("تصنيف")
```

**Prompt** احترافي:

"اكتب برنامج Python يصنف النص القانوني."

مثال:

ادخال: لم يتم دفع الايجار.

اخراج: قضية مدنية."

الكود الناتج:

```
نص = input("ادخل النص: ")
if "يجار" in نص:
    print("قضية مدنية")
else:
    print("غير معروف")
```

---

أهم برامج الذكاء الاصطناعي للمحامين: فيما يلي ملخص لأهم البرامج المفيدة للباحثين القانونيين حول العالم:

### Lexis+ AI .1

من أشهر أنظمة البحث القانوني في العالم.

أهم الوظائف:

- البحث الذكي في القوانين والسوابق القضائية
- تحليل المذكرات القانونية
- اقتراح السوابق القضائية
- تحليل توجهات القضاة

---

### Harvey AI .2

أهم الوظائف:

- تحليل العقود
- due diligence
- البحث القانوني
- إعداد المستندات

---

### Darrow .3

أهم الوظائف:

- اكتشاف القضايا الجماعية
  - تحليل البيانات
  - دعم التقاضي
- 

## AI-Lawyer .4

أهم الوظائف:

- إنشاء العقود
  - تحليل القضايا
  - تبسيط النصوص
  - توقع النتائج
- 

## HAQQ AI .5

أهم الوظائف:

- إدارة القضايا
  - كتابة المذكرات
  - إدارة العملاء
  - إدارة الوثائق
- 

## LegalSure .6

أهم الوظائف:

- كتابة المذكرات
  - تحليل الوثائق
  - البحث القانوني
  - تنظيم المعرفة
-

## أهم استخدامات الذكاء الاصطناعي في مهنة المحاماة:

- البحث في القوانين
- تحليل العقود
- كتابة المذكرات
- إدارة القضايا
- توقع النتائج
- مراقبة الامتثال

كما يمكن للذكاء الاصطناعي مراجعة العقود واستخراج المخاطر بسرعة كبيرة.

---

### المصادر

1. Russell, Stuart & Norvig, Peter. Artificial Intelligence: A Modern Approach.
2. Liu, Pengfei et al. Prompting Methods in NLP.
3. Brown, Tom B. et al. Language Models are Few-Shot Learners.
4. Surden, Harry. Artificial Intelligence and Law.