

دليل البدء السريع

المتطلبات الأولية

قبل البدء، تأكد من توفر:

- ✓ **Python 3.8+** مثبت على نظامك
- ✓ **API مفتاح** (OpenAI أو Google Gemini) واحد على الأقل
- ✓ **ملف نصي** يحتوي على قصتك/سيناريوك

التثبيت (5 دقائق)

الخطوة 1: استنساخ المشروع



```
bash
git clone https://github.com/your-repo/script-analyzer.git
cd script-analyzer
```

الخطوة 2: إنشاء بيئة افتراضية



```
bash
# Linux/Mac
python3 -m venv venv
source venv/bin/activate

# Windows
python -m venv venv
venv\Scripts\activate
```

الخطوة 3: تثبيت التبعيات



```
bash
```

```
pip install --upgrade pip
pip install -r requirements.txt
```

ملاحظة: قد يستغرق التنصيب 5-10 دقائق حسب سرعة الإنترنت

API الخطوة 4: إعداد مفاتيح

الطريقة الأولى: متغيرات البيئة (موصى به)



bash

```
# Linux/Mac
export OPENAI_API_KEY="sk-your-openai-key-here"
export GOOGLE_API_KEY="your-google-api-key-here"

# Windows (PowerShell)
$env:OPENAI_API_KEY="sk-your-openai-key-here"
$env:GOOGLE_API_KEY="your-google-api-key-here"

# Windows (CMD)
set OPENAI_API_KEY=sk-your-openai-key-here
set GOOGLE_API_KEY=your-google-api-key-here
```

env . الطريقة الثانية: ملف

في المجلد الرئيسي env . إنشاء ملف



bash

```
# .env
OPENAI_API_KEY=sk-your-openai-key-here
GOOGLE_API_KEY=your-google-api-key-here
ANTHROPIC_API_KEY=your-anthropic-key # اختياري
TOGETHER_API_KEY=your-together-key # اختياري
```

أول تحليل (دقيقتان)

1. تحضير ملف النص

احفظ قصتك/سيناريوك في ملف نصي، مثال:



my_story.txt

متطلبات الملف:

- .md أو .txt :تنسيق
- UTF-8 :الترميز
- 1,000 - 50,000 كلمة :الطول الموصى به:

2. تشغيل التحليل البسيط.



bash

```
# الانتقال لمجلد backend
cd backend

# إعداد المتغيرات
export PROSE_FILE_PATH="./my_story.txt"
export PROJECT_PATH="./output"
export PROJECT_NAME="MyFirstStory"

# تشغيل المحطة الأولى فقط (اختبار سريع)
python stations/station1.py
```

الوقت المتوقع: 30-60 ثانية

3. التحليل الكامل (جميع المحطات)



bash

```
# تشغيل جميع المحطات السبع
python run_all_stations.py
```

الوقت المتوقع: 5-10 دقائق

4. عرض النتائج

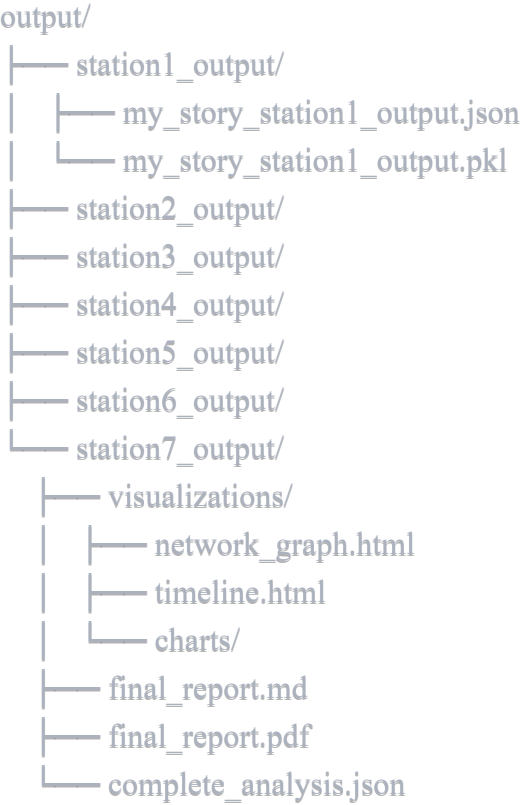


bash

#النتائج ستكون في:

```
cd ../output
```

#هيكل المخرجات:



فهم النتائج

المحطة 1: تحليل النص الأساسي

الملف: station1_output/my_story_station1_output.json



json

```
{
  "major_characters": ["أحمد", "فاطمة", "علي"],
  "character_analysis": {
    "علي": {
      "personality_traits": "شجاع، متهور، مخلص",
      "motivations_goals": "البحث عن العدالة",
      "narrative_function": "البطل المأساوي"
    }
  },
  "narrative_style": {
    "overall_tone": "مظلم ومشوّق",
    "pacing": "سريع"
  }
}
```

المحطة 6: التشخيص والتوصيات

الملف: station6_output/my_story_station6_output.json



json

```
{
  "overall_health_score": 72.5,
  "criticality_level": "moderate_issues",
  "treatment_recommendations": {
    "prioritized_actions": [
      {
        "priority": "high",
        "description": "دمج الشخصية المعزولة اسارة",
        "estimated_time": "ساعات 2-4"
      }
    ],
    "quick_fixes": [
      {
        "description": "تقوية العلاقة بين علي وفاطمة",
        "estimated_time": "دقيقة 30-60"
      }
    ]
  }
}
```

المحطة 7: التصورات

افتح الملفات التفاعلية في المتصفح



bash

```
# فتح الرسم البياني للشبكة
open output/station7_output/visualizations/network_graph.html

# أو
firefox output/station7_output/visualizations/network_graph.html
```

تخصيص التحليل

تعديل ملف التكوين



bash

```
# نسخ ملف التكوين النموذجي
cp backend/config.yaml.example backend/config.yaml

# تعديل الإعدادات
nano backend/config.yaml
```

مثال على التعديلات المفيدة



yaml

```
station_settings:
  station1_text_processing:
    max_characters_analyzed: 10 # الافتراضي: 7
    analysis_depth: "comprehensive" # أو "standard"
```

```
station3_network_builder:
  enable_ai_inference: true
  min_relationship_strength: 3 # الافتراضي: 5
```

```
station6_diagnostics:
  strict_mode: false # أقل صرامة في التشخيص
```

تشغيل محطة واحدة فقط



bash

مثال: تشغيل المحطة الثالثة فقط #

cd backend/stations

python station3_network_builder.py

💡 نصائح للحصول على أفضل النتائج

1. جودة النص

✅ افعل:

- استخدم نصاً كاملاً ومنظماً
- تأكد من وضوح أسماء الشخصيات
- اجعل الصراعات واضحة

❌ لا تفعل:

- تحليل نصوص غير مكتملة
- استخدام أسماء مشفرة أو رموز
- إرسال نصوص قصيرة جداً (>1000 كلمة)

2. إعدادات النماذج

للحصول على دقة عالية



yaml

```
api_providers:
  openai:
    default_model: "gpt-4" # بدلاً من gpt-4-mini
    temperature: 0.3
```

للحصول على سرعة عالية



yaml

```
api_providers:
  openai:
    default_model: "gpt-4o-mini"
    temperature: 0.7
```

إدارة التكاليف 3.

تقدير التكلفة (لنص 20,000 كلمة):

- OpenAI GPT-4-mini: ~\$0.50 - \$1.00
- Google Gemini 1.5 Flash: مجاني (ضمن الحدود)
- OpenAI GPT-4: ~\$2.00 - \$4.00

للتحليل النهائي GPT-4 للاختبار، ثم استخدم Gemini أو GPT-4-mini نصيحة: ابدأ بـ

🦊 حل المشاكل الشائعة

API المشكلة 1: خطأ في مفاتيح



Error: Invalid API key

الحل:



bash

تحقق من المتغيرات

```
echo $OPENAI_API_KEY
```

```
echo $GOOGLE_API_KEY
```

تأكد من صحة المفتاح

OpenAI keys تبدأ بـ *sk-...*

Google API keys طويلة وعشوائية

المشكلة 2: خطأ في الاستيراد



ModuleNotFoundError: No module named 'openai'

الحل:



bash

تأكد من تفعيل البيئة الافتراضية

```
source venv/bin/activate
```

إعادة تثبيت التبعيات

```
pip install -r requirements.txt --upgrade
```

المشكلة 3: ملف النص غير موجود



Error: Prose file not found

الحل:



bash

تحقق من المسار

```
export PROSE_FILE_PATH="$(pwd)/my_story.txt"
```

أو استخدم المسار الكامل

```
export PROSE_FILE_PATH="/home/user/documents/my_story.txt"
```

المشكلة 4: بطء التحليل

الأسباب المحتملة:

- نص طويل جداً (>50,000 كلمة)
- اتصال إنترنت بطيء
- استخدام نماذج ثقيلة (GPT-4)

الحلول:



yaml

في config.yaml

station_settings:

enable_caching: true # تفعيل التخزين المؤقت

api_providers:

openai:

timeout: 120 # زيادة المهلة الزمنية

الخطوات التالية

بعد التحليل الأول الناجح:

1. اقرأ الوثائق الكاملة: [docs/README.md](#)
2. جرب أمثلة متقدمة: [examples/](#)
3. راجع: [docs/configuration.md](#) خصص التحليل: راجع
4. [docs/advanced-features.md](#): اختبر ميزات متقدمة
5. [CONTRIBUTING.md](#): ساهم في المشروع

الأسئلة الشائعة

س: هل يمكنني تحليل نصوص بلغات أخرى؟

ج: نعم! النظام يدعم العربية والإنجليزية بشكل أساسي. للغات أخرى

- استخدم نماذج متعددة اللغات
- Prompts في config.yaml قد تحتاج لتعديل بعض

س: كم مرة يمكنني تشغيل التحليل؟

ج: بدون حدود! لكن انتبه لـ

- (حسب الاشتراك \$: OpenAI) حدود استخدام
- التخزين المؤقت سيجعل التشغيلات اللاحقة أسرع

س: هل يمكن دمج النظام مع أدوات أخرى؟

ج: نعم! النظام يُصدّر

- JSON (للبرمجة)
- REST API (قريباً)
- Python API (استخدم الوحدات مباشرة)

الحصول على المساعدة SOS

إذا واجهت مشكلة:

1. راجع هذا الدليل والوثائق الكاملة 📖
2. [GitHub Issues](#) ابحث في 🔍
3. [Discord Community](#) انضم لـ 💬
4. support@script-analyzer.com راسلنا 📧

جاهز للبدء؟ 🚀 `python run_all_stations.py`

! ✨ التحليل الأول قد يستغرق 5-10 دقائق، لكن النتائج ستستحق الانتظار