Mamdouh AI Course · الكورس التاني

البرمجة بالـ AI

من أول تسطيب Claude Code لحد Spec-Kit والـ Agents و MCP و Skills و Hooks،
ومشروع Scroll-Cinematic في الآخر

1 2 3 4 5 6

عينة مجانية: أول مستويين · إعداد: ممدوح

1

خريطة الرحلة

من سؤال بسيط في التيرمنال لحد فريق agents بيشتغل بطريقتك

1

الأساسيات

التسطيب والبرومبت الصح للكود

←

2

Context وخطة

CLAUDE.md و Plan Mode والتنفيذ

←

3

Spec-Kit

مواصفة ← خطة ← مهام ← تنفيذ

←

4

الـ Agents

Subagents و Loops وفرق

5

التوسعة

MCP و Skills و Plugins و Hooks

←

6

شات AI

مساعد جوه تطبيقك بـ OpenRouter

←

★

المشروع

Scroll-Cinematic

المستوى 1

1

Claude Code: التسطيب والطلب

agent بيقرا مشروعك، ويعدّل ملفات، ويشغّل أوامر، وانت اللي بتوجّه

الأدوات | التسطيب | البرومبت للكود | تراجع إمتى

1

الأدوات: تستخدم إيه إمتى

الأداة	بتعمل إيه	إمتى
Claude.ai (الشات)	يشرح ويكتب snippets ويراجع كود تلزقه	سؤال سريع، أو فهم فكرة
Claude Code (CLI)	agent جوه المشروع: يقرا ويعدّل ويشغّل	أي شغل حقيقي على كود
Desktop (تاب Code)	نفس Claude Code بواجهة، وبراوزر للـ preview	لو مش بتحب التيرمنال
VS Code و JetBrains	Claude Code جوه المحرر، والـ diff قدامك	شغلك اليومي في المحرر
claude.ai/code	سيشن على السحابة بتشتغل على الريبو	مهمة طويلة والجهاز مقفول
Codex و Cursor و Copilot	بدائل، ونفس الأفكار بتشتغل عليهم	لو الفريق بيستخدمها

1

التسطيب وأول تشغيل

```
# Windows (PowerShell)
irm https://claude.ai/install.ps1 | iex

# macOS / Linux
curl -fsSL https://claude.ai/install.sh | bash

# run it inside your project
cd path/to/your-project
claude

# one question, no UI (scripts / automation)
claude -p "summarize the last 5 commits"

# continue the last session / pick an old one
claude -c
claude -r
```

- سطر واحد بيسطّب، ومن غير Node
- افتحه جوه فولدر المشروع دايماً
- أول مرة بيطلب login بحساب Claude
- /doctor لو حاجة مش شغالة
- وفيه Desktop وإضافات VS Code و JetBrains

1

الخطوات الثابتة قبل أي مهمة

1

المكان

افتحه جوه فولدر المشروع الصح

←

2

المشكلة

اوصفها بالعناصر الأربعة

←

3

الخطة

plan/ لأي حاجة أكبر من تعديل صغير

←

4

التنفيذ

خطوة خطوة، والتستات شغالة

←

5

المراجعة

diff/ و code-review/ قبل الـ commit

1

العناصر الأربعة في طلب كود

نفس قاعدة أي برومبت، بس بتفاصيل المشروع

العنصر	في الكود يعني	مثال
Context	الـ stack والملفات المعنية	Laravel 12 API. الـ auth في `AuthController` و `LoginRequest`
Task	المطلوب بالظبط	ضيف rate limit على الـ login: 5 محاولات في الدقيقة لكل IP
Constraints	الممنوع والقواعد	من غير packages جديدة، ومتغيّرش شكل الـ response
Output	شكل التسليم والتحقق	خطة الأول، وبعدين الكود + feature test، وشغّل التستات

1

طلب ضعيف وطلب قوي

❌ ضعيف

```
fix the login
```

✅ قوي

```
Laravel 12 API. Users report they can brute-force
the login. Read AuthController and LoginRequest.

Add rate limiting: 5 attempts per minute per IP,
return 429 with a JSON message in Arabic.
No new packages. Don't change the success response.

First explain your plan in 5 lines and wait.
Then implement + a feature test for the 6th attempt,
and run php artisan test.
```

الطلب القوي فيه: الملفات، والمطلوب برقم، والممنوع، وإزاي يتأكد، و"استنى موافقتي"

2

المستوى 2

الـ Context والخطة والتنفيذ

يفتكر إيه، وتخليه يخطط قبل ما يكتب، وتتأكد من اللي عمله

Context window

CLAUDE.md

Plan Mode

التنفيذ والمراجعة

2

الـ Context: ذاكرة الشغل



كل ملف بيقراه، وكل output، وكل رسالة، بياخد مساحة من نفس الذاكرة

لما تتملي، بيبدأ ينسى التفاصيل الأولانية ويبقى أبطأ وأغلى.

المهارة: تديله اللي محتاجه بس، وتنضّف لما المهمة تتغير، وتحط الثابت في CLAUDE.md.

2

تتحكم في الـ Context إزاي

الأداة	بتعمل إيه	إمتى
`/clear`	محادثة جديدة بـ context فاضي	مهمة جديدة مالهاش علاقة باللي فات
`/compact focus on X`	يلخّص المحادثة ويكمّل	نفس المهمة بس طولت
`/context`	يوريك الـ context مليان بإيه	حاسس إنه بطيء أو بينسى
`@path/file.php`	يشاور على ملف بعينه	بدل "اقرا المشروع كله"
Subagent	مهمة بحث بـ context منفصل	بحث كبير أو output طويل (المستوى 4)
`CLAUDE.md`	حقايق بتتقري كل سيشن	معلومة بتكررها كل مرة

2

CLAUDE.md: يتكتب فيه إيه

ملف بيتقري أوتوماتيك أول كل سيشن، فالنتايج تبقى ثابتة

✅ اكتب

- الـ stack والإصدارات
- أوامر التشغيل والتستات
- Conventions خاصة بمشروعك
- الممنوعات (packages، و API)
- أماكن الحاجات المهمة

❌ متكتبش

- شرح عام للـ framework
- كود كامل (مكانه Skill)
- توكنات أو باسوردات
- تعليمات بتتغير كل يوم
- ملف 1000 سطر

بتستخدم أكتر من أداة؟ اكتب القواعد في AGENTS.md، وخلي CLAUDE.md فيه سطر @AGENTS.md

2

الدورة: Explore ← Plan ← Implement ← Verify

1

Explore

يقرا الكود المعني ويفهم، من غير تعديل

←

2

Plan

يكتب خطة: الملفات والخطوات والتستات

←

3

Implement

ينفّذ خطوة خطوة على الخطة

←

4

Verify

تستات و diff/ و code-/ review

أغلب المشاكل بتيجي من إنه قفز على Implement من غير Explore و Plan

2

Plan Mode: يخطط من غير ما يلمس حاجة

- Shift+Tab لحد ما تشوف plan mode، أو /plan
- بيقرا ويسأل ويكتب الخطة، ومبيعدّلش ولا ملف
- راجع الخطة، وعدّل عليها بالكلام، ووافق
- استخدمه لأي حاجة أكبر من تعديل سطرين
- الخطة الكويسة: ملفات، وخطوات، وتستات، ومخاطر

```
/plan Add order cancellation for customers.
Read OrderController, Order model and the
order tests first.

Rules:
- only orders in "pending" can be cancelled
- the owner only (403 otherwise)
- restore the product stock in a transaction

In the plan list: files to change, migrations,
tests, and anything you are unsure about.
```

2

⭐ Git: شبكة الأمان قبل أي agent

```
git checkout -b feature/order-cancel
git add -A && git commit -m "wip: before agent"

claude
# ... agent works ...

git diff                # read everything
php artisan test

# bad result? throw it away
git checkout main
git branch -D feature/order-cancel
```

- commit قبل ما تديله أي مهمة كبيرة
- branch لكل feature: لو باظت، امسح الـ branch
- راجع git diff بنفسك قبل الـ commit
- /rewind بيرجّع، بس Git هو الضمان الحقيقي
- متخليهوش يعمل push أو force من غير موافقتك