



Claude Code

كتيب تعليم Claude Code من الصفر للاحتراف

دليلك الشامل بالعربي لتعلم أقوى أداة AI للبرمجة – من التثبيت لبناء
مشاريع حقيقية

من صفر لاحتراف 🛠️

أمثلة حقيقية <>

دليل عملي 📖

جمال محفل

@jamal_mohafil



محتويات الكتيب

3	< >	1 شو هو Claude Code؟
4	< >	2 التثبيت والإعداد
5	> _	3 أول أوامرك مع Claude Code
6		4 AGENTS.md و CLAUDE.md – ملفات التعليمات
7		5 Memory و Agents – الذاكرة والكلاء
8		6 Skills و Hooks – المهارات والأتمتة
9		7 كتابة Prompts ذكية
10		8 MCP – توسيع القدرات
11	< >	9 مشروع عملي: بناء موقع كامل
12		10 نصائح وأسرار المحترفين

هالكتيب مبني على تجربة حقيقية – مو كلام نظري. كل شي فيه مجرّب وطبّقته بنفسه.

شو هو Claude Code؟

Claude Code هو أداة من Anthropic بتشتغل من ال Terminal مباشرة. بتخليك تبني مشاريع برمجية كاملة باستخدام الذكاء الاصطناعي – من كتابة الكود لحل المشاكل لعمل Git commits.

ليش مميز؟

- بيتشتغل من ال Terminal
- يفهم المشروع كامل
- يعدل ملفات متعددة
- بيتعلم من تعليماتك
- أسرع 10x من الكتابة اليدوية

شو بيقدر يعمل؟

- يكتب كود كامل من الصفر
- يعدل ملفات موجودة
- يحل bugs ويصلح أخطاء
- يسوي Git commits و PRs
- يقرأ ويفهم أي codebase

Claude Code مش بس أداة – هو زي مبرمج شريك بيتشتغل معك. كل ما تعلمته أكثر عن مشروعك، كل ما صار أقوى. 

التثبيت والإعداد

1 المتطلبات

- Node.js نسخة 18 أو أحدث
- حساب على Anthropic (للا API key)
- Terminal (Mac/Linux/WSL)

2 التثبيت

```
# عالمياً تثبيت Claude Code
npm install -g @anthropic-ai/claude-code
```

3 أول تشغيل

```
# ادخل على مجلد مشروعك
cd my-project

# شغل Claude Code
```

بعد التسجيل، Claude Code رح يبدأ يقرأ مشروعك ويفهم بنيته. خليه يشتغل – هالخطوة مهمة!

⚠ تأكد إنك داخل مجلد المشروع قبل ما تشغل claude. هو بيقرأ الملفات من المجلد الحالي.

أول أوامرك مع Claude Code

بعد ما تشغل Claude Code، تقدر تكلمه بالعربي أو الإنجليزي. هي أهم الأوامر والاستخدامات:

1 أسأله عن المشروع

"اشرح لي بنية هالمشروع" – رح يقرأ الملفات ويعطيك ملخص كامل

2 اطلب منه يبني ميزة

"سويلي صفحة تسجيل دخول بـ React" – رح يكتب الكود ويعدل الملفات

3 صلّح bug

"الزر مش شغال، صلّحه" – رح يلاقي المشكلة ويصلّحها

4 سوّي commit

"commit/" – رح يسوي commit مع رسالة واضحة

5 راجع كود

"راجع التعديلات وقّلي إذا في مشاكل" – code review كامل

كل ما كنت واضح ومحدد بطلبك، كل ما النتيجة أحسن. بدل "سويلي موقع" قل "سويلي landing page بـ Next.js فيها hero section و features و CTA".



ملفات التعليمات – CLAUDE.md و AGENTS.md

Claude Code عنده نظام ملفات تعليمات بمستويات مختلفة. كل مستوى له دور محدد:

Global CLAUDE.md – ~/.claude/CLAUDE.md

تعليمات شخصية تنطبق على كل المشاريع. مثل: "دائماً استخدم pnpm" أو "أنا بفضل TypeScript".

Project CLAUDE.md – ./CLAUDE.md

تعليمات خاصة بالمشروع الحالي. مثل: "هالمشروع Next.js 15" أو "استخدم Tailwind". بيتشارك مع الفريق عبر Git.

Sub-agents – AGENTS.md

لما Claude يشغل agents ثانيين، بيقرأوا AGENTS.md بدل CLAUDE.md. في نسخة global وproject – نفس المنطق.

الأولوية: Project CLAUDE.md ← Global CLAUDE.md ← تعليماتك بالمحادثة. الأقرب يهزم!

```
# مثال Project CLAUDE.md
- Tech: Next.js 15, Tailwind, pnpm
- Always use server components
- Run pnpm build before commit
```

Agents – الوكلاء الذكيين

Claude Code يقدر يشغل sub-agents – نسخ ثانية من نفسه – تشتغل بالتوازي على مهام مختلفة. هي القوة الحقيقية!

كيف بيشتغلوا؟

- Claude الرئيسي بيّفوض مهام لا sub-agents
- كل agent بيشتغل بشكل مستقل على مهمته
- يقرأوا AGENTS.md بدل CLAUDE.md
- بيرجعوا النتائج لا agent الرئيسي

Plan Agent

يخطط الحل قبل ما يبدأ – يحلل المشروع ويقترح خطوات.

Explore Agent

متخصص بالبحث السريع بال codebase – يلاقي ملفات وكود بسرعة.

Memory Agents

Claude Code عنده نظام ذاكرة – بيحفظ معلومات عنك وعن مشاريعك بملفات بمجلد `~/claude/projects/`. بيتذكر تفضيلاتك، feedback سابق، ومعلومات المشروع عبر المحادثات.

قله "تذكر إني بفضل..." ورح يحفظها بال memory. بالمحادثة الجاية رح يتذكر!

Skills و Hooks

1 Skills – المهارات

Skills هي أوامر جاهزة تقدر تنادي عليها بـ /اسم_المهارة. كل skill عنده تعليمات محددة يبتبعها Claude.

أمثلة على Skills

- `commit/` – يسوي commit بأفضل رسالة تلقائياً
- `review-pr/` – يراجع Pull Request
- `simplify/` – يراجع الكود ويحسنه
- `claude-api/` – يساعدك تبني apps بال Claude API

2 Hooks – الأتمتة

Hooks هي أوامر shell تنفذ تلقائياً عند أحداث معينة – مثل قبل/بعد تعديل ملف أو قبل `commit`. تحطهم بـ `settings.json`.

```
// في settings.json
{
  "hooks": {
    "afterEdit": [{
```

Hooks مفيدة لتشغيل linter بعد كل تعديل، أو تشغيل tests تلقائياً. بتوفر وقت كبير!



كتابة Prompts ذكية

الفرق بين مبتدئ ومحترف بـ Claude Code هو طريقة كتابة الـ Prompts. هي أهم القواعد:

Prompt قوي

- "سوّيلي landing page بـ Next.js فيها hero, features, CTA"
- "الزر بالـ header مش شغال لما أضغط عليه بالموبايل"
- "غيّر الألوان لـ dark theme مع الحفاظ على الـ contrast"

Prompt ضعيف

- "سوّيلي موقع"
- "صلّح الكود"
- "غيّر التصميم"

قواعد ذهبية

1 كن محدد – شو بدك بالضبط؟

حدد التقنية، الشكل، السلوك المطلوب

2 وّصف السياق – وين المشكلة؟

اذكر الملف، الكومبوننت، الخطوة الي فيها المشكلة

3 حدد النتيجة المتوقعة

"لما أضغط الزر لازم ينتقل لصفحة /dashboard"

MCP – توسيع القدرات

MCP (Model Context Protocol) بيخلي Claude Code يتصل بأدوات خارجية – مثل قواعد بيانات، APIs، أدوات تصميم، وغيرها. هي القوة الحقيقية!

أمثلة على MCP Servers

- **Screenshot MCP** – يأخذ screenshots لمواقع
- **Nano Banana** – يولّد صور بال AI
- **Database MCP** – يتصل بقاعدة بياناتك
- **Gmail MCP** – يقرأ ويرسل إيميلات

كيف تضيف MCP Server

```
// في ~/.claude.json
{
  "mcpServers": {
    "screenshot": {
      "type": "stdio",
      "command": "npx",
      "args": ["-y", "mcp-screenshot"]
    }
  }
}
```

MCP بيخلي Claude Code أقوى بكثير – بدل ما يكون بس أداة كود، بيصير أداة أتمتة كاملة!

مشروع عملي: بناء موقع كامل

خلينا نبني landing page كاملة من الصفر باستخدام Claude Code. اتبع الخطوات:

< >

1 أنشئ المشروع

1

افتح Terminal واكتب: `npx create-next-app my-site && cd my-site && claude`

2 اكتب CLAUDE.md

2

"سوّلي ملف CLAUDE.md فيه إن المشروع Next.js بـ Tailwind CSS واللغة عربي"

3 ابني ال Hero Section

3

"سوّلي hero section فيها عنوان كبير وزر CTA وخلفية gradient"

4 أضف الأقسام

4

"ضيف قسم features بـ 3 كاردات وقسم testimonials وقسم pricing"

5 الموبايل والتفاصيل

5

"خلي الموقع responsive وأضف animations خفيفة"

6 نشر الموقع

6

"commit/ ثم انشره على Vercel"

هالمشروع بياخذ أقل من ساعة مع Claude Code – بدونك كان يأخذ يوم كامل أو أكثر!

نصائح وأسرار المحترفين

أسرار السرعة

- استخدم `compact/` لتصغير المحادثة لما تطول
- استخدم `clear/` لبداية محادثة جديدة
- اكتب `Shift+Tab` للتبديل بين الوضع العادي وال Plan mode
- استخدم `Escape` لإيقاف Claude أثناء العمل

أسرار الجودة

- حدّث `CLAUDE.md` كل ما تعلمت شي جديد عن مشروعك
- استخدم Plan mode للمشاريع الكبيرة قبل ما تبدأ
- راجع الكود قبل ما تقبله – Claude مش مثالي
- قسّم المهام الكبيرة لخطوات صغيرة

Slash Commands مهمة

- `commit/` – سوّي commit بأفضل رسالة
- `review/` – راجع التعديلات
- `help/` – كل الأوامر المتاحة
- `init/` – سوّي `CLAUDE.md` تلقائياً



شكراً إنك قرأت الكتيب للآخر!

إذا استفدت، شاركه مع صديق يهتم بالبرمجة والذكاء الاصطناعي

@jamal_mohafil

تابعني على Instagram

محتوى يومي عن Claude Code و AI

جمال محفل

@jamal_mohafil

