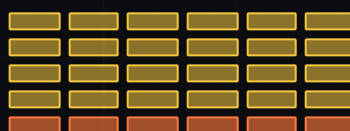


30 TACTICS · CLAUDE CODE

30



تكتيك ونصيحة لاستخدام **Claude Code**
أفضل من **99%** من الناس



LEVEL 01

LEVEL 02

LEVEL 03

المستوى الأول – الأساسيات

01 ابدأ بملف CLAUDE.md

عند تنفيذ الأمر `/init` داخل مشروع قائم، يفحص Claude Code المجلد بالكامل – ملفاته وبنيته ومحتوياته – ثم ينشئ ملف `CLAUDE.md`. يعمل هذا الملف بوصفه دليلًا للمشروع: تعريفًا به، وبياناتًا لمواضع العناصر المهمة فيه، وللقواعد المتبعة في العمل عليه. وفي غيابه تنطلق كل محادثة جديدة من نقطة الصفر، ويتكرر عليك شرح المشروع في كل مرة.

02 أمل صوتًا بدلًا من الكتابة

الإملاء الصوتي أسرع من الكتابة بفارق كبير، ويتيح تمرير قدر أوفر من التفاصيل والسياق في زمن أقل. والتطبيق المستخدم للكتابة الصوتية مبني باستخدام Claude، ورابط تحميله متاح في وصف المقطع.

03 وظيفتان لزر Escape داخل ال terminal

الضغط الواحدة أثناء تنفيذ الوكيل لمهمة تُلغي العملية الجارية وتوقفها فورًا. أما الضغطان المتتاليان فتمسحان النص المكتوب في حقل الإدخال دفعة واحدة، بدلًا من حذفه تدريجيًا.

04 خصص ال status line

يمكنك أن تطلب من Claude Code تخصيص ال status line ليظهر أسفل المحادثة شريط يعرض المؤشرات التي تختارها وفق حاجتك: اسم النموذج المستخدم، ونسبة ال context المستهلك، والتكلفة، وال branch الذي تعمل عليه، وسواها. تمنحك هذه الإضافة متابعة دائمة لحالة الجلسة بدلًا من تقديرها.

05 ضع المحتوى الطويل في ملف بدلًا من لصقه

لا تلصق المحتوى الطويل داخل المحادثة مباشرة؛ احفظه في ملف Markdown ثم اطلب من Claude Code قراءته. يتيح ذلك له قراءة النص على دفعات، أو إنشاء وكلاء لقراءته كاملاً عند الحاجة. والأهم أن النص يصل كاملاً دون فقدان أي جزء منه، خلافاً لما يقع عند اللصق المباشر.

06 راجع /context باستمرار

يعرض الأمر `/context` تفصيلاً لما يستهلك ذاكرة الجلسة في لحظتها. اجعل مراجعته عادة ثابتة؛ فهو يمنحك صورة واضحة عن مسار الجلسة وعن الجهة التي تستنزف مساحتك.

07 استخدام /compact و /clear

تنخفض جودة استجابة النموذج بعد استهلاك ما يقارب 50 إلى 60% من ال context. عند بلوغ هذه النسبة استخدم **/compact** إذا أردت متابعة المحادثة نفسها؛ إذ يختصر الوكيل المحادثة كاملة في ملخص مركّز، فتستأنف العمل بذاكرة أنظف وأداء أدق. أما عند انتهاء الجلسة والانتقال إلى موضوع مختلف كليًا فاستخدم **/clear** لبدء جلسة جديدة من الصفر.

08 ابدأ بال plan mode لا بالتنفيذ

ادخل وضع التخطيط بالضغط على Shift + Tab قبل أي تنفيذ. لا يكتب Claude Code في هذا الوضع شيئًا، بل ينتقل إلى البحث والاستجواب ليعمل معك على تحرير المطلوب وتوضيحه. وثلاثة أرباع النجاح في العمل مع الذكاء الاصطناعي تتحقق في مرحلة التخطيط.

09 اجعل Claude Code يسألك

أهم نقلة في إتقان هذه الأداة أن تحوّلها إلى آلة لطرح الأسئلة. اطلب منه قبل أي تنفيذ، وداخل ال plan mode خصوصًا، أن يوجّه إليك أسئلة كثيرة عن مشروعك حتى يبلغ فهمًا كاملًا لما تريد. وكل سؤال يطرحه في هذه المرحلة يوفّر عليك جولة تعديلات كاملة لاحقًا.

10 لا خطة بلا اختبارات تحقق

اطلب من Claude Code أن يضمّن الخطة اختبارات ووسائل تحقق أثناء بنائها. ودكّر به بأن يدرج ضمن بنودها اختبارًا لفحص ما أنجز والتأكد من عمله قبل الانتقال إلى المرحلة التالية. ويزداد أثر ذلك حين تزوّده بأدوات اختبار مثل Playwright و Chrome DevTools، وتفصيلها في التكتيك رقم 24.

11 استخدم ال artifacts

أصبح Claude Code قادرًا على إنشاء صفحات ويب سريعة ورفعها مباشرة إلى رابط قابل للمشاركة. يمكنك أن تطلب منه بناء داشبورد بصيغة HTML ونشره ك artifact، فيوفّر لك رابطًا تشاركه مع من تشاء دون حاجة إلى أي بنية تحتية.

المستوى الثاني – توسيع القدرات

12 تابع العمل من هاتفك

يربط الأمر `/remote control` محادثتك الجارية بتطبيق Claude على الهاتف مباشرة. تفتح التطبيق فتستأنف العمل في المحادثة نفسها وبالقدرة نفسها، شريطة بقاء الحاسوب قيد التشغيل.

13 امنح Claude Code بريدًا إلكترونيًا خاصًا

يتحول الوكيل إلى مساعد شخصي بالمعنى الحرفي حين يمتلك بريدًا إلكترونيًا خاصًا به. أنشئ حسابًا في خدمة مثل Resend، واحصل على المفتاح من الموقع، واطلب من Claude Code كتابة كود الربط، ثم أضف مفتاح الـ API إلى ملف `.env` لديك. وتبقى الخطوة الأهم: أن تدوّن في ملف `CLAUDE.md` أن الوكيل يملك بريدًا إلكترونيًا، حتى يدرك ذلك في كل المحادثات اللاحقة.

14 استخدم أمر `rewind`

يعيدك أمر `rewind` إلى نقطة سابقة في المحادثة مباشرة، ويزيل معها كل كود كتبه Claude Code بعد تلك النقطة.

15 استخدم أمر `fork`

ينشئ أمر `fork` نسخة من المحادثة الجارية. تصبح لديك محادثتان تحملان السياق نفسه، فتتابع في إحداها هدفًا مختلفًا دون العودة إلى نقطة الصفر.

16 اجعل `CLAUDE.md` خريطة لا مستودعًا

لا تتعامل مع `CLAUDE.md` بوصفه مستودعًا يستوعب كل شيء، بل بوصفه خريطة تدل على موضع كل شيء. ضع المراجع والقوالب والقواعد والملفات المهمة داخل مجلداتها، واجعل `CLAUDE.md` يشير إليها: هذا العنصر موضعه هنا، وذاك موضعه هناك. والسبب أن هذا الملف يُحمّل تلقائيًا مع كل رسالة في كل محادثة، فيقتطع كل سطر فيه جزءًا من الـ context في كل جلسة. لذلك اجتهد ألا يتجاوز حجمه 200 إلى 300 سطر.

17 قاعدة المرأة – ملف agents.md

أنشئ ملفًا باسم agents.md، وأدرج في CLAUDE.md تذكيرًا للوكيل بمطابقة الملفين باستمرار. يؤدي agents.md الدور نفسه تمامًا، غير أنه الملف المعتمد داخل Codex وتطبيق ChatGPT الأخير. وتضمن لك هذه القاعدة أن تفتح المشروع من Codex دون خسارة أي جزء من تنظيمه، فيبقى كل شيء متطابقًا، ولا يرتهن عملك بأداة واحدة.

18 استثمار في بناء ال skills

خصّص جهدًا حقيقيًا لبناء ال skills الخاصة بك: العمليات المتكررة، وال prompts التي تعيد كتابتها باستمرار، وخطوات إنجاز أي عملية من عملياتك. اشرح هذه التفاصيل ل Claude Code واطلب تحويلها إلى skills تُحفظ داخل مجلد **claude**. في مشروعك. وإتقان بناء ال skills واستخدامها ينقل طريقة عملك إلى مستوى مختلف تمامًا.

19 وظيف الوكلاء الفرعيين – sub-agents

لا يكفي وكيل واحد في كثير من الحالات؛ إذ تحتاج إلى وكلاء يعملون تحت إدارته لتسريع التنفيذ واختصار الوقت. فإذا أردت البحث عن معلومة أثناء المحادثة، اطلب من الوكيل الرئيسي أن يوكل البحث في الإنترنت إلى sub-agents؛ فيقرؤون كمًّا كبيرًا من المصادر ويسلّمونه الخلاصة جاهزة. وبذلك يتجنب الوكيل الرئيسي إغراق ذاكرته بالمواد الخام. وهذا مثال واحد من استخدامات كثيرة.

20 افهم ال cache – أكبر مستنزف لحصتك من ال tokens

يأتي أكبر استنزاف لرصيدك من ال tokens من الجهل بآلية ال cache. تبقى المحادثة مخزنة وجاهزة ومنخفضة التكلفة طوال بقائك فيها، لكن ال cache يسقط إذا تركتها وعدت إليها بعد يوم، فتُعاد معالجة تاريخها كاملاً عند أول رسالة ترسلها – وهو ما يستهلك وقتًا ورصيدًا معًا. والحل عادة بسيطة: عند إنهاء محادثة طويلة تنوي استئنافها لاحقًا، اطلب من Claude Code تلخيصها في ملف داخل المشروع يوثق ما أنجز، والقرارات المتخذة، وما تبقى من عمل. وفي اليوم التالي تفتح محادثة جديدة وتطلب من الوكيل قراءة الملف، فيستعيد السياق ويستأنف العمل معك طبيعيًا. ويمكن أتمتة ذلك عبر skill مخصصة – مثل skill «لملمها» المستخدمة في المقطع – تنشئ الملف مباشرة وتحفظه في المجلد المحدد لها.

21 استخدم ultrathink

إذا واجهت مشكلة صعبة وأردت من Claude Code أن يفكر بأقصى طاقته، فأدرج كلمة ultrathink في رسالتك. سيخصص عندها أقصى قدر من التفكير قبل الرد، غير أن استخدامها مكلف، فوظفها بحكمة.

22 استخدام أمر goal/

من أهم الأوامر المضافة حديثًا. اكتب ل Claude Code هدفًا نهائيًا تريد بلوغه وأدرج هذا الأمر في الرسالة، فيتبناه هدفًا شخصيًا ويواصل العمل عليه بإصرار حتى يتحقق. ويشترط أن يكون الهدف قابلاً للقياس والتحقق: «ابن هذا المشروع واستمر حتى يكتمل بناؤه دون أي أخطاء» هدف قابل للقياس، أما «اجعل التصميم جميلًا» فمعيار نسبي لا يقبل القياس، وسيستهلك قدرًا هائلًا من ال tokens دون نتيجة مُرضية.

LEVEL 03

المستوى المتقدم

24 زود Claude Code بأدوات تتحقق من نتائج عمله

الوكيل الذي لا يرى نتيجة عمله يلجأ إلى التخمين. لذلك زوده بأدوات تمكّنه من التحقق بنفسه، كأن تمنحه صلاحية استخدام Playwright أو Chrome DevTools MCP. يصبح عندها قادرًا على فتح الموقع الذي بناه، والضغط على أزراره، وتعبئة ال forms، وقراءة الأخطاء الظاهرة في ال console. أي أنه يستخدم التطبيق بالطريقة نفسها التي يستخدمه بها مستخدم حقيقي.

25 اعتمد نمط التنسيق الوكيل – agent orchestration

من أقوى أساليب استخدام Claude Code أن تخطط مع النموذج الذكي ثم تتنقذ من خلاله بوصفه منسقًا. أي أن يتولى النموذج الذكي دور المنسق بين وكلاء أقل كلفة لبناء المشاريع الكبيرة، فيوزّع عليهم المهام وينظّم أدوارهم. وبعد إعداد خطة مفصلة، اطلب منه تنفيذها بالاعتماد على وكلاء أقل كلفة تحت إشرافه.

26 استخدم ال hooks

ال hooks أوامر تُنفَّذ تلقائيًا عند وقوع حدث محدد؛ أي أنك تعرّف قاعدة من نوع: إذا وقع هذا الحدث فننقذ هذا الإجراء. تُضبط مرة واحدة ثم تعمل آليًا في جميع الجلسات. ويفرّقها عن التعليمات العادية أن ما يُكتب في CLAUDE.md عرضة للنسيان، بينما ال hook قاعدة مضمونة التنفيذ في كل مرة دون استثناء. ومن أمثلتها hook يصدر تنبيهًا صوتيًا فور انتهاء أي مهمة، فتشغّل عدة جلسات وتنصرف إلى عمل آخر، ويعلمك الصوت باكتمالها. أو hook يمنع الوكيل من المساس بمجلد بعينه. وأيسر طريقة لإنشائها أن تصف المطلوب ل Claude Code بلغة عادية، فيتولى ضبط الإعدادات بنفسه.

27 استخدام ال work trees

ستحتاج في مرحلة ما إلى تشغيل جلستين في وقت واحد: واحدة تبني ميزة، وأخرى تعالج خللاً. وإذا عملت الجلستان داخل المجلد نفسه فسيجري تعديل الملفات ذاتها من الطرفين، فيكتب وكيل ما يحذفه الآخر، ويقع التداخل. والحل هو ال work trees: اطلب من Claude Code إنشاء work tree باسم محدد، فينشئ نسخة كاملة معزولة من المشروع في مجلد جانبي تعمل عليه تلك الجلسة باستقلال، وتُفعل المثل في المحادثة الأخرى لهدف مختلف. تحصل بذلك على مسارين منفصلين تمامًا، لكل منهما نسخته، ثم تدمج النسخ في المشروع الأساسي عند الانتهاء.

28 استخدام أمر loop

إذا أردت من Claude Code تنفيذ مهمة متكررة كل مدة زمنية محددة فأمر loop مخصص لذلك: «افحص هذا الموقع كل خمس دقائق، وأرسل إليّ بريدًا فور عمله»، أو «راجع أداء التطبيق كل ساعة وأرسل إليّ ملخصًا». ويستمر ال loop في العمل حاليًا حتى أسبوع كامل.

29 استخدام أمر schedule لما يجب أن يستمر في غيابك

أما ما يجب أن يستمر حتى مع إطفاء الحاسوب — كبريد أسبوعي بآخر تطورات الذكاء الاصطناعي، أو ملخص صباحي يومي لجدولك — فمكانه أمر schedule: تحدد ما تريد إنجازه وتوقيت إنجازه، ويتولى الوكيل التنفيذ. ويجدر التنبيه إلى أن schedule يعمل في السحابة، وأن Claude في السحابة لا يملك صلاحية الوصول إلى ملف `.env`؛ فإذا احتاجت المهمة إلى مفاتيح وصول وجب إنشاء environment في موقع Claude وإضافة المفاتيح هناك.

30 احم ملف `.env`

يُعد ملف `.env` أخطر ملف في مشروعك، لاحتوائه على المفاتيح والأسرار وما يماثلها من بيانات حساسة. لذلك يجب منع الوكيل من قراءته أو تعديله منقًا قاطعًا. توجد في ملف `settings.json` داخل مجلد `.claude` الرئيسي قوائم المنع أو ال deny rules؛ أدرج فيها منع أي وكيل من قراءة `.env` أو التعديل عليه، مع السماح له فقط بإضافة مفتاح جديد على هيئة placeholder فارغ تملؤه بنفسك، دون أن يُطلع على أي قيمة. وتحميك هذه القاعدة من ال prompt injection: فقد تتضمن skill تُحمّلها أو صفحة ويب تقرأها prompt مخفيًا يطلب من وكيلك إرسال الأسرار إلى جهة خارجية، واحتمال أن ينقذ الوكيل هذه التعليمات ظنًا أنها صادرة عنك احتمال قائم وكبير.

هذه هي النقاط كاملة كما وردت في المقطع — ابدأ بتطبيق واحدة منها اليوم، لا بتطبيقها جميعًا.