

دليل التعلم المتكامل

# 100 خطوة

لإتقان

# Claude

كتاب مرجعي شامل — من الصفر إلى الاحتراف

نظرية متينة وتطبيق عملي

10 فصول ▪ 100 خطوة ▪ منهجية متدرجة

الإصدار العربي ▪ 2026

# 100 خطوة لإتقان Claude

الدليل العربي المتكامل — من الصفر إلى الاحتراف

---

منهجية مدروسة ▪ أساس نظري + تطبيق عملي  
10 فصول ▪ 100 خطوة

الإصدار العربي ▪ 2026

# مقدمة

في الزمن الذي صار فيه الذكاء الاصطناعي جزءاً من حياتنا اليومية، لم تعد المسألة هل نستخدمه — بل كيف نستخدمه بإتقان.

هذا الكتاب ليس دليلاً سريعاً، ولا مجموعة حيل عابرة، هو منهج مدروس مكون من 100 خطوة موزعة على 10 فصول، بُنيت بترتيب متدرج يأخذك من اللحظة الأولى التي تفتح فيها Claude، حتى اللحظة التي تُصبح فيها قادراً على أتمتة سير عمل كامل، وبناء أنظمة متكاملة، والتفكير كمستخدم محترف من النخبة.

كل خطوة تحتوي على ركيزتين: أساس نظري يُعمق فهمك، وتطبيق عملي يُترجم الفهم إلى مهارة. كثير من الخطوات يصاحبها مثال جاهز على prompt، ونصيحة من خبرة الميدان، وتحذير من خطأ شائع وقع فيه كثير غيرك.

لا تستعجل. اقرأ خطوة، طبق، ثم انتقل. إذا طبقت 100 خطوة بصدق، ستجد نفسك في عالم آخر تماماً.

## كيف تستخدم هذا الكتاب؟

- اقرأ الفصل بالترتيب أول مرة — كل فصل يبني على ما قبله.
- خصص لكل خطوة 15-30 دقيقة على الأقل: قراءة + تطبيق فوري.
- افتح Claude بجانبك أثناء القراءة. التطبيق وقت القراءة أعمق من التطبيق بعدها.
- احتفظ بدفتر ملاحظات (رقمي أو ورقي) سجل فيه ما تعلمته من كل خطوة.
  - بعد إنهاء فصل، ارجع لخطواته وضع علامة على ما أتقنته فعلاً.
  - لا تنتقل لفصل جديد قبل تطبيق 80% من خطوات الفصل السابق.
  - ارجع للكتاب كمرجع — ليس بالضرورة قراءة خطية في كل مرة.

«الأداة لا تصنع المحترف، لكن المحترف يصنع بأي أداة معجزة»



# جدول المحتويات

رحلتك في 10 فصول و 100 خطوة

الخطوات 1-10	<b>اللبات الأولى لفهم Claude — الأساسيات</b>	<b>01</b>
الخطوات 11-20	<b>يفصل بين النتائج العادية والمذهلة — صياغة الـ Prompts الفن الذي</b>	<b>02</b>
الخطوات 21-30	<b>تحويل النصوص إلى منتجات تفاعلية حقيقية — الـ Artifacts</b>	<b>03</b>
الخطوات 31-40	<b>بناء قدرات قابلة لإعادة الاستخدام — الـ Skills</b>	<b>04</b>
الخطوات 41-50	<b>تحويل Claude من أداة إلى شريك — Claude كزميل عمل</b>	<b>05</b>
الخطوات 51-60	<b>ربط Claude بعالمك الرقمي — الموصلات (Connectors)</b>	<b>06</b>
الخطوات 61-70	<b>اجعل Claude يعمل بينما أنت نائم — الأتمتة (Automation)</b>	<b>07</b>
الخطوات 71-80	<b>الأداة التي تكتب الكود بدلاً عنك — Claude Code</b>	<b>08</b>
الخطوات 81-90	<b>كل مجالات الحياة والعمل — حالات الاستخدام المتقدمة في Claude</b>	<b>09</b>
الخطوات 91-100	<b>أن تصير نخبة النخبة — المستخدم الخارق</b>	<b>10</b>



الفصل

# الأول

## الأساسيات

اللبنة الأولى لفهم Claude

الهدف من هذا الفصل

استخدام Claude بفعالية

# الفصل 1: الأساسيات

قبل أن نتقن أي أداة، يجب أن نعرف ماهيتها وكيف تعمل وأين تقع نقاط قوتها وحدودها. هذا الفصل هو حجر الأساس — لا تتجاوز به بسرعة، فهم هذه المفاهيم العشر سيضاعف فاعليتك في كل ما يلي.

## الهدف من هذا الفصل

### استخدام Claude بفعالية

## خطوات هذا الفصل:

- 01 ما هو Claude؟
- 02 نماذج Claude المختلفة
- 03 كيف تعمل المحادثات؟
- 04 نافذة السياق (Context Window)
- 05 أفضل حالات الاستخدام
- 06 خطط الأسعار
- 07 جولة في الواجهة
- 08 الإعدادات التي يجب أن تعرفها
- 09 اختصارات لوحة المفاتيح
- 10 Claude في الجوال والتطبيقات

# 01 ما هو Claude؟

## تعريف وفهم طبيعة المساعد

### • الأساس النظري

Claude هو نموذج لغوي كبير (Large Language Model) طورته شركة Anthropic، وهو يعمل كمساعد ذكي قائم على المحادثة. لكنه ليس مجرد روبوت يردّ على الأسئلة — بل هو شريك تفكير قادر على التحليل والكتابة والبرمجة والتلخيص والإبداع.

ما يميز Claude عن غيره من النماذج هو تركيز Anthropic على ثلاث ركائز: المساعدة الفعالة (Helpful)، الصدق (Honest)، وعدم الضرر (Harmless). هذا المنهج المُسمّى Constitutional AI يجعل Claude يميل لإخبارك عندما لا يعرف، ويقاوم إعطاء معلومات مضلّة.

تذكر دائماً: Claude ليس محرك بحث، وليس قاعدة بيانات. هو نموذج تنبؤي يولّد النص كلمة تلو الأخرى بناءً على ما تعلمه. كلما فهمت طبيعته، أحسنت توجيهه.

### • التطبيق العملي

1. افتح Claude.ai أو تطبيق Claude.
2. اطرح سؤالاً مفتوحاً: «ما الذي يمكنك مساعدتي به؟»
3. لاحظ كيف يستجيب — ليس بقائمة جامدة بل بحوار طبيعي.
4. جرّب طلباً عملياً صغيراً (مثال: «لخص لي مفهوم الذكاء الاصطناعي في 3 جمل»).
5. اقرأ الإجابة بتمعّن وفكر في كيفية إنتاجها.

#### مثال على prompt

اشرح لي بأسلوب بسيط جداً، كأني طفل في العاشرة، كيف يعمل النموذج اللغوي الذي يُشغلك.

#### • نصيحة احترافية

كلما تعاملت مع Claude كزميل ذكي وليس كألة، حصلت على نتائج أفضل بكثير.

## نماذج Claude المختلفة

Opus, Sonnet, Haiku — متى تستخدم كل واحد؟

# 02

### • الأساس النظري

تصدر Anthropic عدة نماذج بأحجام مختلفة، وكل نموذج يُمثل توازناً بين الذكاء، السرعة، والتكلفة.

Opus: الأذكى والأقوى — مخصص للمهام المعقدة (تحليل عميق، كتابة طويلة، برمجة متقدمة، بحوث). أبطأ وأغلى لكنه يفكر بعمق.

Sonnet: التوازن المثالي — قوي وذكي، أسرع من Opus، مناسب لأغلب المهام اليومية: الكتابة، البرمجة، التحليل المتوسط.

Haiku: الأسرع والأخف — مثالي للمهام السريعة والمتكررة، الردود الفورية، والتطبيقات التي تحتاج استجابة لحظية.

قاعدة عملية: ابدأ بـ Sonnet في معظم المهام. ارفع إلى Opus إذا واجهت تعقيداً يحتاج تفكيراً عميقاً. انزل إلى Haiku للسرعة وتقليل التكلفة.

### • التطبيق العملي

1. في Claude.ai، انقر على اسم النموذج أعلى المحادثة.
2. جرب نفس prompt على نموذجين مختلفين (مثلاً Sonnet ثم Opus).
3. قارن العمق والجودة والسرعة.
4. اختر مهمة معقدة (تحليل مستند طويل) — جربها على Opus.
5. اختر مهمة بسيطة (إعادة صياغة جملة) — جربها على Haiku.
6. دون ملاحظتك: متى تشعر بفرق حقيقي؟ متى لا يهم النموذج؟

#### • نصيحة احترافية

لا تستخدم Opus لكل شيء. ستستهلك حصتك بسرعة دون داع. Sonnet يكفي 80% من المهام.

#### • احذر: خطأ شائع

الاعتقاد بأن «الأقوى دائماً أفضل». في المهام البسيطة، Opus قد يُعطيك إجابة أطول وأكثر تعقيداً من اللازم.

## كيف تعمل المحادثات؟

فهم بنية الحوار مع Claude

# 03

### • الأساس النظري

كل محادثة مع Claude تتكون من تبادل رسائل بين «المستخدم» و«المساعد». Claude يقرأ كل ما سبق في المحادثة عند توليد رده.

هذا يعني أن السياق يتراكم: ما قلته في الرسالة الأولى يؤثر على فهم الرسالة العاشرة. هذه ميزة قوية — ولكنها أيضاً عبء على نافذة السياق.

Claude لا يحتفظ بذاكرة بين المحادثات المنفصلة افتراضياً. كل محادثة جديدة تبدأ من الصفر (إلا إذا فعلت ميزة الذاكرة).

المحادثة الواحدة تعالج كتلة واحدة. حتى لو كانت فيها 50 رسالة، Claude يقرأها كلها في كل دور.

### • التطبيق العملي

1. ابدأ محادثة جديدة واطرح سؤالاً.
2. في الرسالة الثانية، استخدم كلمة «هذا» أو «السابق» — لاحظ كيف يفهم Claude المرجعية.
3. اطرح سؤالاً مختلفاً تماماً في الرسالة الثالثة.
4. اطلب منه في الرسالة الرابعة أن يلخص ما ناقشتموه — ستندهش من قدرته على ذلك.
5. أغلق المحادثة، افتح أخرى جديدة، واسأل: «ما كان آخر شيء ناقشناه؟» — سيخبرك أنه لا يعرف.

#### • نصيحة احترافية

للمشاريع الطويلة، استخدم محادثة واحدة طويلة بدل عدة محادثات قصيرة — يحتفظ بالسياق.

## نافذة السياق (Context Window)

ما يستطيع Claude رؤيته ومعالجته في وقت واحد

# 04

### • الأساس النظري

نافذة السياق هي «الذاكرة العاملة» لـ Claude في المحادثة الحالية. تُقاس بالـ tokens (الـ token تقريباً 3-4 حروف عربية أو كلمة إنجليزية قصيرة).

نماذج Claude الحديثة تدعم 200,000 token وأكثر — أي ما يُعادل كتاباً كاملاً تقريباً. هذا يسمح برفع ملفات ضخمة وتحليلها.

لكن هناك تحذير مهم: كلما اقتربت من نهاية النافذة، يبدأ Claude في «نسيان» التفاصيل الدقيقة من بداية المحادثة. هذا ليس عيباً بل طبيعة النموذج.

النافذة تشمل: تعليمات النظام + رسائل المستخدم + رسائل Claude السابقة + الملفات المرفقة. كل شيء يُحسب.

### • التطبيق العملي

1. ارفع ملف PDF أو نص طويل إلى Claude.
2. اسأله عن تفاصيل من بداية الملف، ثم من النهاية — قارن الدقة.
3. في محادثة طويلة، اطلب منه أحياناً: «اذكري بما اتفقنا عليه في البداية» للتحقق من الاحتفاظ.
4. عندما تشعر أن المحادثة بدأت تتلعثم، ابدأ محادثة جديدة وأعطه ملخصاً سريعاً.

#### • نصيحة احترافية

في المهام الكبيرة، ضع المعلومات الأهم في بداية الرسالة، فالنماذج تميل لإعطاء وزن أكبر للبداية والنهاية.

#### • احذر: خطأ شائع

إغراق المحادثة بملفات كثيرة غير لازمة. يُضعف الأداء ويُربك النموذج.

## أفضل حالات الاستخدام

أين يتفوق Claude فعلاً؟

# 05

### • الأساس النظري

Claude متميز بشكل خاص في: التحليل العميق للنصوص الطويلة، الكتابة الإبداعية والتقنية، البرمجة (خاصة في Claude Code)، التلخيص الذكي، الترجمة المتأنية، التفكير المنطقي خطوة بخطوة.

Claude أقل قوة في: العمليات الحسابية الدقيقة جداً (استخدم Python بدلاً من ذلك)، المعلومات اللحظية الحديثة (إلا مع البحث)، توليد الصور (ليس من اختصاصه).

أفضل ما يميز Claude هو «الحوار المعطوق». يمكنك العودة لتنقيح إجابة، طرح أسئلة فرعية، طلب إعادة الصياغة — وهو يحافظ على خيط النقاش.

### • التطبيق العملي

1. اكتب قائمة بـ 5 مهام تكررهما يومياً.
2. صفها: هل هذه مهمة تحليل؟ كتابة؟ برمجة؟ بحث؟
3. اختر واحدة وجرب Claude فيها لمدة أسبوع.
4. قس الفرق: كم وقر من وقتك؟ هل الجودة أفضل أم أسوأ من المعتاد؟

#### • نصيحة احترافية

أفضل استخدام لـ Claude ليس استبدالك، بل تضخيمك. هو الـ amplifier لقدراتك، ليس بديلاً عنها.

## خطط الأسعار

Free, Pro, Max, Team, Enterprise — أيها يناسبك؟

# 06

### • الأساس النظري

Free: مجاني، مع حدود يومية على عدد الرسائل والميزات. ممتاز للتجربة والاستخدام العرّضي.

Pro: اشتراك شهري، يتيح استخدام نماذج Opus، حصة أكبر بكثير، الوصول إلى Projects و Artifacts بحرية أوسع، ميزات أحدث أولاً.

Max: مخصص للمستخدمين الكثيفين، حصة أعلى من Pro بأضعاف، أولوية في الوصول.

Team / Enterprise: للشركات والفرق، يُضيف إدارة المستخدمين، أمان متقدم، Connectors مؤسسية، وخصوصية كاملة للبيانات.

ابدأ بـ Free، ارتق إلى Pro عندما تشعر بحدود الاستخدام، ولا تنتقل إلى Max إلا إذا كنت تستهلك حصة Pro كاملة بانتظام.

### • التطبيق العملي

1. ادخل إلى Claude.ai | Settings | Subscription.
2. اقرأ تفاصيل كل خطة بنفسك (قد تتغير مع الوقت).
3. ابدأ المجاني لأسبوع — قس عدد المرات التي وصلت فيها للحد.
4. إذا تجاوزت الحد 3 مرات في الأسبوع، Pro يستحق الترقية.

#### • نصيحة احترافية

Pro يستحق ثمنه إذا كنت تستخدم Claude للعمل أو الدراسة الجادة. التكلفة أقل بكثير من قيمة الوقت الذي يوفره.

## جولة في الواجهة

تعرف على عناصر Claude.ai

# 07

### • الأساس النظري

واجهة Claude بسيطة بشكل متعمد، لكنها تخفي قوة كبيرة. أهم العناصر: الشريط الجانبي: تاريخ المحادثات، المشاريع، الإعدادات. محرك الكتابة: حقل النص الرئيسي، أزرار الرفع، وزر «+» للأدوات والمرفقات. اختيار النموذج: أعلى المحادثة — اختر بين Sonnet/Opus/Haiku. Artifacts: لوحة جانبية تظهر عند توليد كود أو محتوى منظم. Settings: حيث تفعل/تعطل: البحث، الذاكرة، Connectors، Code Execution.

### • التطبيق العملي

1. افتح Claude.ai وتنقل بين كل عنصر مرة على الأقل.
2. أنشئ مشروعاً جديداً (Project) من الشريط الجانبي.
3. افتح Settings ▢ ألق نظرة على كل تبويب.
4. جرّب اختصارات لوحة المفاتيح (مثل Cmd/Ctrl + K).
5. خصّ واجهتك: الوضع الداكن، اللغة، إلخ.

#### • نصيحة احترافية

10 دقائق من التجول بلا هدف في الإعدادات تكشف لك ميزات لن تكتشفها بالاستخدام العادي.

# الإعدادات التي يجب أن تعرفها

ضبط Claude ليناسب أسلوبك

## 08

### • الأساس النظري

هناك إعدادات تُحدث فرقاً كبيراً ولا يعرفها كثير من المستخدمين:

**User Preferences**: مكان تُخبر فيه Claude عن نفسك وكيف تحب أن يرد. مثال: «أنا مطور برمجيات، أفضّل

الإجابات الموجزة، استخدم العربية في الشرح والإنجليزية في الكود.»

**Writing Style**: يسمح بضبط أسلوب الكتابة — رسمي، ودود، تقني، مختصر. يمكنك إنشاء أسلوبك الخاص.

**Memory**: عند تفعيلها، يحتفظ Claude بمعلومات مهمة من محادثاتك السابقة.

**Web Search**: يمكن Claude من البحث في الإنترنت — أساسي للمعلومات الحديثة.

**Connectors**: تربط Claude بحساباتك (Drive, GitHub, Notion, ...).

### • التطبيق العملي

1. ادخل Settings ▾ Profile ▾ User Preferences.
2. اكتب فقرة قصيرة عن نفسك واهتماماتك ومهنتك.
3. اكتب تفضيلاتك في الردود (الطول، الأسلوب، اللغة).
4. فعل Web Search.
5. اختر نمط كتابة، أو أنشئ نمطاً مخصصاً.
6. ابدأ محادثة جديدة ولاحظ الفرق.

#### • نصيحة احترافية

User Preferences هو سرّ المحترفين — يؤثر عليك كتابة نفس التعليمات في كل محادثة.

## اختصارات لوحة المفاتيح

أسرع طريقة لاستخدام Claude

# 09

### • الأساس النظري

إذا كنت تستخدم Claude بكثرة، اختصارات لوحة المفاتيح ستوفر عليك ساعات أسبوعياً. هذه أهمها:

**Cmd/Ctrl + K**: فتح بحث سريع في محادثتك.

**Cmd/Ctrl + Shift + O**: محادثة جديدة.

**Cmd/Ctrl + /**: عرض كل الاختصارات.

**Cmd/Ctrl + Enter**: إرسال الرسالة.

**Shift + Enter**: سطر جديد داخل الرسالة.

**Up Arrow**: تعديل آخر رسالة.

### • التطبيق العملي

1. اضغط **Cmd/Ctrl + /** في Claude.ai لرؤية كل الاختصارات.

2. ألزم نفسك باستخدام 3 اختصارات جديدة يومياً لأسبوع.

3. ستجدها صارت عادة دون تفكير.

#### • نصيحة احترافية

أكبر مكسب: تعلم **Shift + Enter** لكسر السطر، و **Up Arrow** لتعديل آخر prompt بدل إعادة كتابته.

# Claude في الجوّال والتطبيقات

استخدم Claude في كل مكان

# 10

## • الأساس النظري

Claude متوفر على: المتصفح (claude.ai)، تطبيق iOS، تطبيق Android، تطبيق Mac، تطبيق Windows. تطبيق الجوّال يدعم: الإدخال الصوتي، التقاط الصور وتحليلها، المحادثة الفورية، التزامن مع باقي الأجهزة. تطبيق سطح المكتب يضيف: اختصار سريع للاستدعاء (مثل Spotlight)، تكامل أعمق مع الملفات، أداء أفضل. كل التطبيقات تتزامن: تبدأ محادثة في الجوّال، تكملها على الكمبيوتر.

## • التطبيق العملي

1. ثبت تطبيق Claude على هاتفك.
2. ثبت تطبيق سطح المكتب على جهازك.
3. جرّب الإدخال الصوتي — تحدث بدل الكتابة.
4. التقط صورة لشيء معتد (وصفة، رسم بياني، نص أجنبي) واطلب التحليل.
5. أنشئ اختصاراً سريعاً للاستدعاء Claude بضغط زر.

### • نصيحة احترافية

الإدخال الصوتي يُسرّع التفكير بصوت مرتفع — جرّبه عند العصف الذهني.



الفصل

# الثاني

## صياغة الـ Prompts

الفن الذي يفصل بين النتائج العادية والمذهلة

الهدف من هذا الفصل

جعل Claude متوقعاً ودقيقاً

## الفصل 2: صياغة الـ Prompts

صياغة الـ prompt هي المهارة الأهم في عصر الذكاء الاصطناعي. ليست شيئاً سحرياً، بل مهارة منهجية يمكن تعلمها وإتقانها. في هذا الفصل، ستتعلم صياغة prompts تنتج نتائج احترافية باستمرار.

### الهدف من هذا الفصل

### جعل Claude متوقعاً ودقيقاً

### خطوات هذا الفصل:

- 11 Prompting المبني على الأدوار
- 12 تكديس السياق
- 13 التفكير المتسلسل
- 14 إضافة القيود
- 15 تسلسل الـ Prompts
- 16 نقد المخرجات
- 17 إعادة كتابة سير العمل
- 18 تعليمات النظام
- 19 إعادة الاستخدام: قوالب الـ Prompts
- 20 إجمالي الفصل: صيغة الـ Prompt المثالي

# Prompting المبني على الأدوار

Role Prompting — تعيين شخصية واضحة

# 11

## • الأساس النظري

Role Prompting هو إعطاء Claude دوراً واضحاً يتقّمه قبل الإجابة. مثال: «أنت كاتب إعلانات محترف بـ 15 سنة خبرة في مجال البرمجيات.»

لماذا يعمل؟ النموذج يميل لتقمّص الشخصية في أسلوبه ومستوى عمقه. «أنت طبيب» يُنتج إجابة مختلفة كلياً عن «أنت صديق يشرح بأسلوب بسيط.»

الدور الجيد يحتوي على: المهنة + الخبرة + التخصص + السياق. كلما زادت الدقة، تحسّنت الإجابة.

## • التطبيق العملي

1. اختر مهمة (مثلاً: شرح موضوع تقني).
2. اكتبها مرة بلا دور: «اشرح لي الـ Blockchain.»
3. اكتبها مع دور: «أنت أستاذ علوم حاسب يشرح للطلاب المبتدئين. اشرح لي الـ Blockchain.»
4. قارن الإجابتين — ستجد فرقاً ملموساً.

### مثال على prompt

أنت محرر صحفي متحك في صحيفة اقتصادية كبرى. مهمتك: راجع المقال التالي، حسن صياغته، وأشر إلى الجمل الضعيفة. حافظ على نبرة الكاتب الأصلية.

### نصيحة احترافية •

الدور الأكثر تخصصاً يُنتج إجابة أعمق. «كاتب» عام، «كاتب سيناريو لأفلام الرعب النفسي» محدد ومُلهِم.

## تكديس السياق

Context Stacking — بناء السياق طبقة فوق طبقة

## 12

## • الأساس النظري

Context Stacking هو إعطاء النموذج معلومات السياق بشكل متدرج ومنظم قبل طلب المهمة. كلما زاد السياق المُحكّم، قلّ الغموض في الإجابة.

البنية المثلى: من العام إلى الخاص. اشرح من أنت، ثم اشرح المشروع، ثم اشرح المهمة الحالية، ثم اطلب الفائدة؛ تتجنب أسئلة استيضاحية، تلغي الاحتمالات الخاطئة، تختصر الذهاب والإياب.

## • التطبيق العملي

1. خذ مهمة كنت ستطلبها بسطر واحد.
2. أعد صياغتها بـ 4 طبقات: من أنا | ما المشروع | ما الوضع الحالي | ما المطلوب.
3. ستجد جودة الإجابة تتضاعف.

## مثال على prompt

[المشروع] أعدّ منهجاً لطلاب المستوى [من أنا] مدرس لغة عربية للناطقين بغيرها.  
[المطلوب] اكتب لي خطة درس مدتها 45 [الوضع] أحتاج درساً عن الأفعال الناقصة. المتوسط.  
دقيقة تتضمن: تمهيد، شرح، تمارين تطبيقية، وتقييم.

## • نصيحة احترافية

كلما طال المشروع وزاد تعقيده، تكدس السياق يصير ضرورة لا ترفاً.

## التفكير المتسلسل

Chain-of-Thought — اجعل Claude يفكر خطوة بخطوة

## 13

## • الأساس النظري

Chain-of-Thought (سلسلة التفكير) هو طلب صريح من النموذج بأن يفصل تفكيره قبل الوصول للإجابة النهائية. عبارة بسيطة مثل «فكر خطوة بخطوة» تحسّن الدقة بشكل لافت في المسائل المعقدة.

لماذا يعمل؟ النموذج يُؤدّ النص خطياً. عندما يفكر علناً، كل خطوة تُغذي الخطوة التالية. هذا يُقلّل القفزات المنطقية ويكشف الأخطاء.

أنواع: إما تطلب الخطوات صراحة («اشرح خطواتك»)، أو تعطى مثالاً يُوضّح الأسلوب (Few-shot CoT).

## • التطبيق العملي

1. اطرح مسألة منطقية متوسطة بدون CoT.
2. اطرحها مرة ثانية مع «فكر خطوة بخطوة قبل أن تجيب».
3. في مسائل الرياضيات والمنطق والكود، الفرق دراماتيكي.

## مثال على prompt

لديّ مشروعان محتملان. الأول: تكلفة 5,000 دولار، عائد متوقع 8,000 دولار خلال سنة. الثاني: تكلفة 12,000 دولار، عائد 18,000 خلال سنتين. أيهما أفضل من حيث ROI السنوي؟ فكر خطوة بخطوة قبل الإجابة.

## • نصيحة احترافية

حتى للأسئلة البسيطة، إضافة «اشرح أولاً ثم أجب» تحسّن الجودة 30-40%.

## إضافة القيود

Constraints — حدد ما لا تريد، ليس فقط ما تريد

## 14

## • الأساس النظري

القيود (Constraints) تُوجّه النموذج بحدود واضحة: طول الإجابة، اللغة، النبرة، التنسيق، ما يجب تجنبه. كلما كانت القيود أوضح، كانت الإجابة أقرب لما تتخيل.

أنواع القيود الشائعة: حدّ الطول (في حدود 200 كلمة)، الأسلوب (رسمي، ودود)، التنسيق (نقاط، فقرات، جدول)، ما يجب تجنبه (لا تستخدم مصطلحات تقنية، لا تعط أمثلة).

تذكر: «لا تكتب طويلاً» قيد ضعيف. «اكتب في حدود 100 كلمة» قيد قوي.

## • التطبيق العملي

1. اكتب prompt بدون قيود، انظر الإجابة.
2. أعد صياغته بـ 3 قيود محددة على الأقل.
3. قارن: أيهما أقرب لما كنت تتخيل؟

## مثال على prompt

اكتب لي تغريدة عن قراءة الكتب. القيود: في حدود 280 حرفاً، نبرة دافئة وشخصية، تحتوي على دعوة عملية للقارئ، تتجنب الكلمات المبتذلة كـ «رحلة» و «شغف».

## • نصيحة احترافية

أقوى قيد قد يكون «لا تفعل». ذكر ما لا تريد يلغي مساحة كاملة من الاحتمالات الخاطئة.

## تسلسل الـ Prompts

— Prompt Chaining — تقسيم المهام الكبيرة

# 15

### • الأساس النظري

Prompt Chaining هو تقسيم مهمة كبيرة إلى عدة prompts متتابعة، كل واحد يبني على نتيجة السابق. هذا يُحسن الدقة لأن النموذج يُركز على شيء واحد في كل مرة.

متى تحتاجه؟ في المهام المركبة: «اقرأ، طل، استنتج، اكتب تقريراً». بدل طلب الكل دفعة واحدة، اطلب كل خطوة منفصلة.

الفائدة: تستطيع التحقق من كل خطوة قبل الانتقال، تصحح المسار مبكراً، تتحكم بالمخرج النهائي.

### • التطبيق العملي

1. خذ مهمة معقدة (مثل: «اكتب لي خطة تسويق متكاملة»).
2. قسمها إلى 4-5 خطوات: تحليل الجمهور | تحديد الرسالة | اختيار القنوات | جدول زمني | ميزانية.
3. نفذ كل خطوة في prompt منفصل، ومرر نتيجة كل واحد للتالي.
4. قارن الجودة مع طلب كل شيء دفعة واحدة.

#### مثال على prompt

[الخطوة 1] قبل البدء، طل الجمهور المستهدف لمنتج تطبيق تمارين رياضية للنساء العاملات بين (ثم مرر هذا 40-25 سنة. حدد: احتياجاتهن، تحدياتهن اليومية، قنوات تواجدهن، وما يُحركهن للشراء. التحليل للخطوة التالية: «بناءً على التحليل أعلاه، اقترح 3 رسائل تسويقية...»)

#### • نصيحة احترافية

Chaining ينقذك في المهام التي تحتاج +1000 كلمة من المخرجات. لا تطلبها دفعة واحدة.

## نقد المخرجات

اجعل Claude يُراجع نفسه — Critique Outputs

## 16

## • الأساس النظري

بعد توليد إجابة، اطلب من Claude أن ينتقد عمله. النموذج جيد جداً في رؤية الضعف في نص آخر — حتى لو كان النص نصه هو.

العملية: prompt 1 يُنتج المسوّدة، prompt 2 يقول «راجع المسوّدة أعلاه، حدّد 5 نقاط ضعف، واقترح تحسينات»، prompt 3 يطلب إعادة الكتابة مع التحسينات.

هذه دورة بسيطة لكنها ترفع جودة المخرج بشكل كبير، خصوصاً في الكتابة والكود.

## • التطبيق العملي

1. اكتب نصاً قصيراً (200 كلمة).
2. اطلب نقده بقسوة: «أنت محرر صارم. انتقد هذا النص بصدق.»
3. اطلب إعادة كتابته بناءً على النقد.
4. كرر الدورة مرتين — ستحصل على نسخة محترفة.

## مثال على prompt

هذا هو المقال الذي كتبه أعلاه. اعتبر نفسك محرراً صارماً في مجلة مرموقة. ابحث عن: الجمل الضعيفة، التكرار، الادعاءات بلا دليل، النقاط التي تحتاج توضيحاً. كن صريحاً.

## • نصيحة احترافية

اطلب نقداً «صارماً» أو «قاسياً» — وإلا سيُجاملك Claude بشكل مفرط.

# إعادة كتابة سير العمل

## 17

حسن prompts الموجودة — Rewrite Workflows

### • الأساس النظري

أحد أقوى استخدامات Claude هو تحسين prompts نفسها! اكتب prompt كما تتخيله، ثم اطلب من Claude تحسينه.

النموذج يعرف ما يعمل وما لا يعمل في الـ prompts. سيُضيف هيكلًا، يوضح الغموض، يقترح قيودًا. مع الوقت، تتعلم من تعديلاته وتحسن أسلوبك في الصياغة.

### • التطبيق العملي

1. خذ prompt قديم استخدمته.
2. أعطه لـ Claude مع: «حسن هذا الـ prompt ليُنتج إجابة أفضل. اشرح ما غيرت ولماذا.»
3. احفظ الـ prompts المُحسنة في وثيقة مرجعية.

#### مثال على prompt

هذا prompt استخدمه لطلب ملخصات: «لخص لي هذا النص.» — قيم هذا الـ prompt، اقترح تحسينات تجعله أكثر دقة وفائدة، واكتب نسخة محسنة.

#### • نصيحة احترافية

Meta-prompting (الـ prompt الذي يصنع prompts) من أقوى التقنيات في 2026.

## تعليمات النظام

System Prompts — التعليمات التي تحكم كل المحادثة

# 18

### • الأساس النظري

تعليمات النظام (System Prompt) هي تعليمات «إطار» تُعطى مرة واحدة وتُطبق على كل المحادثة. في Claude.ai، تظهر في صورة User Preferences أو Custom Instructions في Projects.

بدل تكرار «رد بالعربية»، استخدم أسلوباً علمياً، تجنّب المصطلحات الأجنبية» في كل رسالة، اكتبها مرة في الـ System.

هذه التعليمات لها وزن أكبر من تعليمات داخل المحادثة، وتظل فعالة طوال الحديث.

### • التطبيق العملي

1. في Claude.ai، أنشئ Project جديداً.
2. اكتب System Prompt مفصّل للمشروع: من أنت، الأسلوب، اللغة، التنسيق.
3. ابدأ محادثات داخل المشروع — ستجد Claude متسقاً تماماً مع التعليمات.
4. قارن مع محادثة عادية بدون System Prompt.

#### مثال على prompt

أنت مساعد لمدرّس فيزياء ثانوي. أجني دائماً بالعربية الفصحى، (كـ System Prompt في Project):  
استخدم تشبيهات من الحياة اليومية، ابدأ كل شرح بسؤال محتمل، واختم بتمرين تطبيقي. تجنّب المعادلات المعقدة إلا إذا طُلبت.

#### • نصيحة احترافية

Projects مع System Prompts هي أقوى ميزة في Claude للعمل المنظم. لا تتجاهلها.

## إعادة الاستخدام: قوالب الـ Prompts

Reusable Templates — بنك prompts شخصي

# 19

### • الأساس النظري

المحترفون لا يكتبون كل prompt من الصفر. لديهم بنك من القوالب الجاهزة يُعدّلونها للحاجة. أنشئ ملفاً (في Notion أو ملاحظات أو حتى نص بسيط) فيه قوالبك الأكثر استخداماً: قالب تلخيص، قالب نقد، قالب ترجمة، قالب كتابة مقال، إلخ. كل قالب يحتوي على: الدور + السياق + المهمة + القيود + التنسيق + متغيرات قابلة للتعبئة [بين أقواس].

### • التطبيق العملي

1. حدّد 5 مهام تكررّها أسبوعياً.
2. اكتب قالباً لكل واحدة بحقول قابلة للتعبئة.
3. احفظها في مكان سهل الوصول (Notion, Drive, حتى Apple Notes).
4. استخدمها لأسبوع، حسّنّها بناءً على الاستخدام.

#### مثال على prompt

«أنت محلّ [المجال]. لخص [نوع المحتوى] التالي في [عدد] نقاط رئيسية. ركز على قالب تلخيص: [الزاوية الموهبة]. تجنّب [ما يجب تجنّبه]. اختتم بـ [النوع المطلوب من الخاصة].»

#### • نصيحة احترافية

10 قوالب جيّدة محفوظة عندك تساوي 100 محاولة عشوائية في الكفاءة.

# إجمالي الفصل: صيغة الـ Prompt المثالي

البنية الذهبية — Perfect Prompt Formula

## 20

### • الأساس النظري

بعد كل ما سبق، إليك الصيغة الذهبية لـ prompt احترافي يجمع كل ما تعلمته:

1. ROLE (الدور): من تريد Claude أن يكون؟

2. CONTEXT (السياق): الخلفية والجمهور والوضع.

3. TASK (المهمة): ما المطلوب بالضبط؟

4. CONSTRAINTS (القيود): القواعد والحدود.

5. FORMAT (التنسيق): شكل الإخراج المطلوب.

ليست كل prompt تحتاج كل العناصر، لكن كلما كانت المهمة أهم، اقترب من الصيغة الكاملة.

### • التطبيق العملي

1. خذ آخر prompt كتبته اليوم.

2. أعد صياغته باستخدام الصيغة الذهبية كاملة.

3. نفذ النسختين — قارن النتائج.

4. احفظ الصيغة في User Preferences كقالب ذهني.

#### مثال على prompt

[CONTEXT] العميل؛ ستارت أب يُطلق تطبيق صحة نفسية. [ROLE] أنت كاتب إعلانات محترف.  
[TASK] اكتب 5 عناوين إعلانية لحملة الجمهور؛ شباب 22-35 سنة في المدن.  
[CONSTRAINTS] كل عنوان أقل من 12 كلمة، نبرة دافئة لا تخويفية، لا تذكر «اكتئاب». Instagram.  
[FORMAT] قائمة مرقمة، كل عنوان متبوع بسطر مباشرة، استخدم سؤالاً في عناوين على الأقل.  
يشرح لماذا سيعمل.

#### • نصيحة احترافية

احفظ هذه الصيغة في ذهنك كأنها معادلة. مع الوقت، ستصاغ تلقائياً في رأسك قبل أن تكتب.



الفصل

# الثالث

## Artifacts الـ

تحويل النصوص إلى منتجات تفاعلية حقيقية

الهدف من هذا الفصل

**إنشاء مخرجات تفاعلية احترافية**

## الفصل 3: الـ Artifacts

الـ Artifacts هي الميزة التي حولت Claude من «مساعد محادثة» إلى «صانع منتجات». بدل أن تأخذ إجابة نصية وتحوّلها بنفسك إلى شيء قابل للاستخدام، Claude يُنشئ لك المنتج جاهزاً؛ صفحة ويب، تطبيق صغير، حاسبة، تقرير منسق...

### الهدف من هذا الفصل

### إنشاء مخرجات تفاعلية احترافية

### خطوات هذا الفصل:

21 ما هي الـ Artifacts؟

22 متى يُولد Claude Artifact تلقائياً؟

23 صياغة Prompts للـ Artifacts

24 مشاريع بسيطة لتبدأ بها

25 التصميم المتجاوب

26 التعديل المباشر

27 تصحيح الأخطاء (Debugging)

28 أساسيات الـ Frontend

29 React الـ Prompting

30 بناء «منتجات صغيرة»

## ما هي الـ Artifacts؟

تعريف الميزة الثورية

# 21

### • الأساس النظري

Artifact هي محتوى مستقل، قابل للحفظ والمشاركة، يُنتجها Claude في لوحة جانبية منفصلة عن المحادثة. بدلاً من رؤية كود طويل داخل الرد، تراه يُنفذ كصفحة ويب حقيقية.

أنواع Artifacts: صفحات HTML تفاعلية، مكونات React، رسوم SVG، مخططات Mermaid، مستندات Markdown طويلة، أكواد كاملة.

ميزتها: تتطور مع المحادثة. تقول «اجعل الزر أحمر»، فيحدث Claude نفس الـ artifact مباشرة. أنت تعمل مع نسخة حية، لا تنسخ وتلصق من ردود متفرقة.

### • التطبيق العملي

1. اطلب: «أنشئ صفحة ويب بسيطة تعرض اسمي ومهنتي بتصميم جميل.»
2. ستظهر لوحة على اليمين تعرض الصفحة فعلياً.
3. اطلب تعديلاً: «اجعل خلفية الصفحة متدرجة من أزرق إلى بنفسجي.»
4. لاحظ أن الـ artifact تحددت، لم يُكتب رد جديد.

#### • نصيحة احترافية

الـ Artifacts ليست للكود فقط — مستندات طويلة، خطط عمل، حتى قصائد منسقة، كلها يمكن أن تكون artifacts.

## متى يُولد Claude Artifact تلقائياً؟

When They Trigger — فهم منطق التشغيل

# 22

### • الأساس النظري

Claude يُولد artifact تلقائياً عندما يكتشف أن المخرج: (1) طويل (أكثر من ~20 سطرًا)، (2) قابل لإعادة الاستخدام خارج المحادثة، (3) تفاعلي، (4) منظم له بنية واضحة.

لن تظهر artifact إذا كنت تسأل سؤالاً نقاشياً، تطلب تلخيصاً قصيراً، أو تتحاور بشكل طبيعي.

يمكنك إجبار توليد artifact بطلب صريح: «ضع هذا في artifact» أو «أعطني هذا كمستند قابل للحفظ».

### • التطبيق العملي

1. اطلب جملة قصيرة: «اكتب لي شعاراً لشركتي.» — لن يظهر artifact.
2. اطلب صفحة كاملة: «اكتب لي صفحة هبوط لمنتجتي.» — سيظهر artifact.
3. اطلب مراجعة: «ضع رد التلخيص السابق في artifact منسق.»

#### • نصيحة احترافية

إذا أردت محتوى للحفظ والمشاركة، اطلب artifact صراحة — لا تنتظر التشغيل التلقائي.

## صياغة Prompts للـ Artifacts

Artifact Prompting — أسلوب خاص للمنتجات التفاعلية

# 23

### • الأساس النظري

صياغة prompt لـ artifact تختلف عن prompt عادي. أنت لا تطلب نصاً، بل منتجاً. يحتاج: الغرض، الجمهور، الميزات، التصميم، السلوك.

أفضل صياغة تتضمن: ما يفعله المنتج، من يستخدمه، كيف يبدو، ماذا يحدث عند التفاعل، أي قيود تقنية. كلما كان وصفك أقرب لـ «brief مصمم»، كانت النتيجة أقرب للاعتراف.

### • التطبيق العملي

1. اطلب: «أنشئ حاسبة بسيطة.»
2. ثم اطلب: «أنشئ حاسبة لحساب ROI الاستثمار. تأخذ: مبلغ الاستثمار، العائد المتوقع، عدد السنوات. تعرض: ROI سنوي، إجمالي الربح. تصميم مستوحى من Stripe — خلفية بيضاء، أزرار brand color أزرق، typography حديثة.»
3. قارن الـ artifacts الناتجة.

#### مثال على prompt

أنشئ صفحة تعليمية واحدة (single page) تشرح مفهوم «الذكاء الاصطناعي» للمبتدئين. تتضمن: عنوان جذاب، 3 أمثلة بأيقونات، فيديو وهمي placeholder، زر CTA للتسجيل في قائمة بريدية. تصميم: ألوان دافئة، خط Inter، استجابة كاملة للموبايل.

#### • نصيحة احترافية

أعط Claude مرجعاً مرئياً («مستوحى من Apple» / «مثل Stripe» / «أسلوب Notion») — يفهم الإشارات الجمالية.

# مشاريع بسيطة لتبدأ بها

Simple Projects — تمارين عملية

## 24

### • الأساس النظري

أفضل طريقة لتتقن Artifacts: ابن مشاريع صغيرة قابلة للإكمال في دقائق، كل واحد يُعلمك جانباً. 10 مشاريع للمبتدئين: بطاقة عمل، صفحة ملف شخصي، قائمة مهام، مؤقت Pomodoro، مولد كلمات سر، حاسبة BMI، عداد سعرات، عرض شرائح، quiz بسيط، مفكرة يومية. كل مشروع لا يأخذ أكثر من 5 دقائق، لكنه يُكسبك إلفة بالأداة.

### • التطبيق العملي

1. اختر 3 مشاريع من القائمة أعلاه.
2. نفذ كل واحد في محادثة منفصلة.
3. بعد التوليد الأول، اطلب 3 تحسينات على الأقل.
4. احفظ الـ HTML الناتج في ملفاتك.

#### مثال على prompt

أنشئ تطبيق قائمة مهام بسيط بـ HTML/CSS/JS فقط (بلا أطر). الميزات: إضافة مهمة، تعليمها كمنجزة، حذفها، حفظ في localStorage. تصميم: minimalist، ألوان pastel لطيفة عند الإضافة والحذف.

#### • نصيحة احترافية

لا تنتظر مشروعاً كبيراً لتبدأ — المشاريع الصغيرة تبني المهارة الحقيقية.

# التصميم المتجاوب

## 25

Responsive Design — يعمل على كل الشاشات

### الأساس النظري

Artifacts تعمل في الموبايل والتابلت والديسكتوب. لتضمن جودة المخرج، اطلب تصميمًا متجاوبًا صراحة. كلمات سحرية: «touch-friendly»، «works on all screen sizes»، «mobile-first»، «responsive». Claude يستخدم CSS Flexbox و Grid و media queries تلقائياً عندما تطلب الاستجابة.

### التطبيق العملي

1. أنشئ artifact بدون ذكر الاستجابة، افتحه على شاشة صغيرة (Mobile view في DevTools).
2. أعد التوليد مع طلب صريح للاستجابة.
3. قارن سلوك التصميم على الموبايل.

#### مثال على prompt

أنشئ صفحة هبوط احترافية لمنتج SaaS. مطلوب: breakpoints، mobile-first design، عند 768px و 1024px، صور تكيف، نصوص بأحجام نسبية (touch targets)، (rem كبيرة على الموبايل).

#### نصيحة احترافية

اطلب «mobile-first» وليس «responsive» فقط — يعطي نتائج أفضل لأنه يبدأ بالموبايل أساساً.

## التعديل المباشر

Live Editing — حوار مع المنتج

# 26

### • الأساس النظري

أقوى ميزة في Artifacts هي أن كل تعديل يُحدث المنتج بدون فقدان ما سبق. هذا يحول العملية إلى حوار إبداعي.

بدل أن تكتب كل المتطلبات دفعة واحدة، ابن تدريجياً: أنشئ النسخة الأولى `عدل` `عدل` `عدل`. هذا يُحاكي طريقة عمل مصمم حقيقي مع عميل: iteration مستمر.

### • التطبيق العملي

1. أنشئ artifact بسيط.
2. أجر 5 تعديلات متتالية، كل منها صغير: «اجعل اللون أحمر» `أضف ظل للزر` `غير الخط` ...
3. لاحظ كيف يحتفظ Claude بكل ما تم قبل التعديل.

#### • نصيحة احترافية

إذا أردت الرجوع لنسخة قديمة، استخدم أزرار التنقل في الـ artifact panel — كل نسخة محفوظة!

## تصحيح الأخطاء (Debugging)

Debugging — عندما لا يعمل الـ artifact

# 27

### • الأساس النظري

أحياناً يُنشئ Claude artifact به أخطاء: لا يعمل، يبدو سيئاً، يفتقد ميزة، لا تياًس — هذا طبيعي. أفضل طريقة للتصحيح: صف ما تراه بدقة. «الزر لا يعمل» سيء، «عندما أضغط زر الحفظ، لا يحدث شيء، ولا أرى أي خطأ في وحدة التحكم» ممتاز. أعط Claude المعلومات التشخيصية: ما توقعت، ما حدث فعلياً، أي رسائل خطأ، صورة للشاشة إن أمكن.

### • التطبيق العملي

1. أنشئ artifact معقد قليلاً (مثل: تطبيق quiz بـ 5 أسئلة).
2. اختبره كاملاً — ستجد بعض الأخطاء غالباً.
3. صف الأخطاء بدقة واطلب الإصلاح.
4. كرر حتى يعمل بشكل مثالي.

#### مثال على prompt

في الـ quiz الذي أنشأناه: عند الإجابة على السؤال الثالث، يقفز إلى الخامس مباشرة بدل الرابع. أيضاً النتيجة النهائية تظهر صفر دائماً حتى لو أجبت كل شيء صحيحاً. أصلح هذين الخطأين.

#### • نصيحة احترافية

إذا فشل الإصلاح مرتين، ابدأ artifact جديد من الصفر بمواصفات أوضح — أحياناً أسرع من التصحيح.

# أساسيات الـ Frontend

## 28

Frontend Basics — تعلم أكثر تطلب أفضل

### • الأساس النظري

لا تحتاج أن تكون مبرمجاً لاستخدام Artifacts. لكن إن عرفت بعض مفاهيم الويب، ستصوغ طلبات أدق:

HTML: بنية الصفحة (عناوين، فقرات، أزرار).

CSS: التصميم (ألوان، خطوط، تخطيط).

JavaScript: التفاعل (ما يحدث عند الضغط، الحفظ، إلخ).

React: مكتبة لبناء واجهات تفاعلية معقدة (Claude يدعمها كاملاً).

Tailwind CSS: إطار CSS شائع، يُسرّع التصميم.

### • التطبيق العملي

1. اطلب من Claude: «اشرح لي الفرق بين HTML و CSS و JavaScript بمثال عملي.»
2. اطلب: «أنشئ artifact يستخدم Tailwind فقط، واطرح لي كل class مستخدم.»
3. بعد أسبوع، ستفهم نسبة كبيرة من الكود الذي يولده.

#### • نصيحة احترافية

تعلم Tailwind CSS بعمق — ستفهم 80% من تصاميم Artifacts، وتقدر تعديلها بنفسك.

## React AI Prompting

تطبيقات تفاعلية حقيقية — React Prompting

# 29

### • الأساس النظري

React هي مكتبة لبناء واجهات معقدة بـ «مكونات» قابلة لإعادة الاستخدام. Claude ممتاز في React. متى تستخدم React في Artifacts؟ عندما تحتاج: حالة (state) تتغير، تفاعلات متعددة، بنية معقدة، استخدام مكتبات (مثل recharts للرسوم). اذكر صراحة: «أنشئ مكون React» للتأكد من استخدامه.

### • التطبيق العملي

1. اطلب: «أنشئ مكون React لـ dashboard يعرض 4 KPIs مع رسوم بيانية بسيطة. استخدم recharts.»
2. ستحصل على dashboard تفاعلي حقيقي.
3. اطلب إضافة filter (مثلاً: «أضف drop-down لاختيار الفترة الزمنية»).

#### مثال على prompt

أنشئ مكون React تفاعلي: مؤشر تعلم اللغة. يعرض: عدد الكلمات اليومية، streak (سلسلة الأيام)، رسم بياني أسبوعي. الحالة محفوظة في useState. تصميم بـ Tailwind، استخدم lucide-react للأيقونات.

#### • نصيحة احترافية

في Claude، Artifacts لديه وصول لمكتبات: recharts للرسوم، lucide-react للأيقونات، lodash للدوال — استغلها.

# بناء «منتجات صغيرة» 30

## Build Mini Products — مشاريع متكاملة قابلة للاستخدام

### • الأساس النظري

الخطوة المتقدمة في Artifacts: بناء «mini products» — تطبيقات صغيرة لكنها متكاملة وفعلاً مفيدة. أمثلة: مولد سيرة ذاتية، أداة تخطيط ميزانية، حاسبة ضرائب بلدية، تطبيق تتبع عادات، محرر markdown، أداة تصميم ألوان متناسقة. هذه المنتجات يمكن مشاركتها مع الآخرين، نشرها كصفحات مستقلة، أو إدماجها في عملك.

### • التطبيق العملي

1. اختر مشكلة تواجهها شخصياً.
2. ابن artifact يحلها. لا تقلق إذا لم يكن مثالياً — ابدأ.
3. حسنه عبر iterations.
4. احفظ النسخة النهائية وشاركها — حتى لو مع نفسك مستقبلاً.

#### مثال على prompt

أحتاج تطبيق ويب صغير لإدارة قراءاتي. ميزات: إضافة كتاب (عنوان، مؤلف، عدد صفحات)، تتبع التقدم (صفحة حالية)، حالة (أقرأ / منتهي / مؤجل)، إحصائيات (إجمالي كتب السنة، صفحات يومياً). تخزين localStorage. تصميم: قراءة مريحة، typography هادئة، ألوان warm.

#### • نصيحة احترافية

في نهاية كل أسبوع، اسأل نفسك: «أي مشكلة أضيع فيها وقتاً؟» — وابن artifact يحلها.



الفصل

# الرابع

## الـ Skills

بناء قدرات قابلة لإعادة الاستخدام

الهدف من هذا الفصل

**بناء سير عمل قابل لإعادة الاستخدام**

## الفصل 4: الـ Skills

Skills هي «مهارات» مخصصة لتعلم Claude كيف يقوم بمهام معينة بطريقة المفضلة. بدل أن تشرح كيف تكتب التقارير في كل مرة، تنشئ Skill يحفظ الأسلوب ويستدعي بكلمة سحرية.

### الهدف من هذا الفصل

### بناء سير عمل قابل لإعادة الاستخدام

### خطوات هذا الفصل:

- 31 ما هي الـ Skills؟
- 32 بنية الـ Skill
- 33 مهارات الكتابة
- 34 مهارات البحث
- 35 مهارات الاستراتيجية
- 36 مكتبات الـ Skills
- 37 الاتساق
- 38 تقليل الهلوسة
- 39 تخصص سير العمل
- 40 لا تبدأ من الصفر

## ما هي ال Skills؟

مفهوم Skills في Claude

# 31

### • الأساس النظري

Skills هي وحدات معرفة منظمة (مجلدات تحتوي على ملف SKILL.md وملفات مساعدة) يستدعيها Claude تلقائياً عندما تتطابق المهمة مع وصف ال skill.

كل skill يُعلم Claude كيف يتعامل مع نوع معين من المهام: إنشاء مستندات Word، تصميم عروض PPT، فحص ملفات Excel، كتابة بأسلوب علامتك التجارية...

المفهوم أقوى من «قالب»: هو منظومة كاملة فيها تعليمات، أمثلة، قواعد، وأحياناً أكواد جاهزة.

### • التطبيق العملي

1. في Claude.ai، استكشف Skills المتاحة (إذا كانت ميزة Skills مفعلة لحسابك).
2. اطلب مهمة تطابق skill موجودة (مثل: «أنشئ لي تقرير Word احترافي»).
3. لاحظ كيف يستخدم Claude ال skill تلقائياً.

#### • نصيحة احترافية

Skills تُعلم Claude ال «كيف»، بينما ال prompts تُعلمه ال «ماذا». كلاهما معاً = قوة حقيقية.

## بنية ال Skill

Skill Structure — تشرح ال skill

# 32

### • الأساس النظري

كل skill تحتوي على: ملف SKILL.md الرئيسي (فيه: الاسم، الوصف، التعليمات)، وأحياناً ملفات مساعدة (أمثلة، نماذج، scripts).

ملف SKILL.md يبدأ بـ YAML frontmatter يحوي الاسم والوصف. الوصف مهم جداً لأن Claude يستخدمه ليقرر متى يُفعل ال skill.

ثم يأتي محتوى ال skill بأسلوب Markdown: التعليمات، الإجراءات، الأمثلة، حالات الاستخدام، وحالات الاستثناء.

### • التطبيق العملي

1. افحص skill جاهز إذا أتيح لك (في Claude للمؤسسات).
2. لاحظ: كيف يُوصف؟ كيف تُكتب التعليمات؟ هل هناك أمثلة؟
3. ارسم في ورقة هيكل skill خاص بك (مثل: skill لكتابة منشورات LinkedIn).

#### • نصيحة احترافية

Skill description هو أهم سطر — يجب أن يصف بدقة متى تُفعل ال skill، وإلا لن يستدعيها Claude.

## مهارات الكتابة

Writing Skills — skills للمحررين

## 33

## • الأساس النظري

Skills الكتابة تُعلم Claude أسلوبك الخاص، أمثلة: skill لكتابة مقالات بأسلوبك، skill لردود البريد بنبرتك، skill لكتابة سيناريوهات فيديو قصيرة.

محتوى skill الكتابة: قواعد الأسلوب، مفردات مفضلة وممنوعة، بنية النص (مقدمة-جسم-خاتمة)، أمثلة من كتاباتك السابقة، أخطاء شائعة يجب تجنبها.

بمجرد إنشاء الـ skill، يصير Claude «كأنه أنت» في الكتابة — ليس فقط منتجاً للنص، بل بصوتك.

## • التطبيق العملي

1. اختر نوع كتابة تكرر (مثلاً: تغريدات تقنية).
2. اجمع 5-10 من أفضل كتاباتك السابقة.
3. اكتب وصفاً لأسلوبك: ما يميزه، ما تتجنبه.
4. أنشئ skill (أو ملف User Preferences) يحتوي على ذلك.
5. اطلب من Claude الكتابة، قارن مع أسلوبك.

## مثال على prompt

«اكتب نبذة دافئة وشخصية. ابدأ بقصة قصيرة أو سؤال محفز. استخدم جملاً (محتوى skill كتابة):  
قصيرة. تجنب: «بالأكيد»، «من المهم»، «في الواقع». اتم بـ takeaway عملي. أعط القارئ شيئاً  
يُجزبه اليوم.»

## • نصيحة احترافية

أعط Claude 3 نماذج «حقيقية» من كتاباتك — يتعلم النبذة منها أكثر من 100 سطر تعليمات.

# مهارات البحث

## 34

Research Skills — منهجية البحث

### • الأساس النظري

Skills البحث تُعلم Claude منهجيتك في البحث: مصادر مفضلة، طريقة التقييم، شكل الملخص، مستوى التعمق. مثال: skill «بحث استثمائي» يطلب: تحليل الشركة، المنافسين، السوق، المخاطر، التوصية — بهذا الترتيب وبهذا التنسيق.

تجعل البحث منظماً، شاملاً، وقابلًا للمقارنة عبر مواضيع مختلفة.

### • التطبيق العملي

1. اختر مجال بحث تتكرر فيه (تقنية، صحة، أعمال).
2. اكتب «منهجيتك المثالية» في البحث: ما الأسئلة الأساسية؟ ما مصادرك؟ ما شكل المخرج؟
3. أنشئ skill أو قالب يحفظ المنهجية.
4. استخدمها في 3 بحوث مختلفة، حسنّها بناءً على التجربة.

#### • نصيحة احترافية

اطلب من skill البحث أن «يستشهد بالمصادر» — هذا يجعل Claude يبحث فعلياً ولا يخترع.

## مهارات الاستراتيجية

Strategy Skills — التفكير الاستراتيجي

## 35

## • الأساس النظري

Skills الاستراتيجية تحول Claude إلى مستشار استراتيجي. مثلًا: skill «تحليل skill»، «تحليل المنافسين».

كل skill يحتوي على: الإطار النظري، الأسئلة الموجهة، خطوات التطبيق، تنسيق المخرج. الفائدة: تستفيد من قوة Claude التحليلية مع منهجية واضحة، لا تخمين.

## • التطبيق العملي

1. اختر إطاراً استراتيجياً تعرفه (Porter's 5 Forces, SWOT, Lean Canvas).
2. اكتب skill يطبقه على أي شركة/منتج تذكره.
3. جربه على شركة تعرفها — قس عمق التحليل.

## مثال على prompt

«عند طلب تحليل شركة، اتبع: (1 لمحة عامة، 2 نموذج العمل، 3 منافسين (skill استراتيجي مبسط): أساسيين، 4 نقاط قوة فريدة، 5 تحديات، 6 فرص نمو، 7 توصيات استراتيجية. كل قسم في حدود 100 كلمة.»

## • نصيحة احترافية

أفضل skills الاستراتيجية تلهمها كتب كلاسيكية: «The Lean»، «Blue Ocean Strategy»، «Good to Great»، «Startup».

## مكتبات الـ Skills

بناء مكتبتك الخاصة — Skill Libraries

## 36

## • الأساس النظري

مع الوقت، تتراكم لديك skills كثيرة. تنظيمها في «مكتبة» يجعل العمل أسرع وأكثر اتساقاً. هيكل المكتبة المقترح: مجلد رئيسي «Skills»، مجلدات فرعية حسب الوظيفة (كتابة، تحليل، تسويق، إلخ)، كل skill في مجلده الخاص. وثق كل skill: ما يفعل، متى يُستخدم، أمثلة، إصدارات سابقة.

## • التطبيق العملي

1. أنشئ مجلد «My Claude Skills» على Notion أو Drive.
2. ابدأ بـ 3 skills أساسية تستخدمها يومياً.
3. أضف skill جديد كل أسبوعين تقريباً.
4. راجع المكتبة شهرياً وحذف ما لا تستخدمه.

## • نصيحة احترافية

Skills لا تستخدم < حذفها أفضل من احتفاظها — تترك Claude في الاختيار.

## الاتساق

Consistency — اتساق المخرجات

## 37

## • الأساس النظري

أحد أكبر فوائد Skills: الاتساق. كل تقرير، كل مقال، كل تحليل يأتي بنفس البنية والأسلوب. للوصول لاتساق عالٍ، حدد القوالب بوضوح، استخدم headers ثابتة، حدد أطوال الأقسام، وضع نموذج «output template».

هذا يفيد بشكل خاص في العمل الجماعي — كل أعضاء الفريق يُنتجون مخرجات متشابهة.

## • التطبيق العملي

1. أنشئ skill بسيط (مثلاً: «ملخص اجتماع»).
2. ضع template واضح: الحضور، الأجندة، النقاشات، القرارات، المهام، المتابعة.
3. استخدمه على 3 اجتماعات مختلفة، قارن النتائج.
4. هل البنية متطابقة؟ هل المحتوى متسق؟

## • نصيحة احترافية

اطلب صراحة: «اتبع الـ template دون انحراف» — لتقليل التغييرات العشوائية.

## تقليل الهلوسة

Reduce Hallucination — منع الأخطاء

## 38

## • الأساس النظري

«الهلوسة» (Hallucination) تعني توليد معلومات خاطئة بثقة. Skills تساعد في تقليلها عبر: تعليمات صارمة («لا تذكر أي حقيقة بلا مصدر»)، منهجية تحقق، قوالب تطلب الاستشهاد.

اكتب في skill عبارات مثل: «إذا لم تكن متأكدًا، قل لا أعرف.»، «استشهد بالمصدر في كل ادعاء.»، «لا تخرع أرقامًا.»

أيضًا: استخدم Web Search في skill يحتاج معلومات حالية.

## • التطبيق العملي

1. اختر موضوعًا تقنيًا تعرفه جيدًا.
2. اطلب من Claude شرحه دون skill — لاحظ هل هناك أخطاء.
3. أنشئ skill مع تعليمات صارمة ضد الهلوسة، أعد الطلب.
4. قارن الدقة.

## مثال على prompt

- إذا لم تجد المعلومة، اكتب صراحة: - لا تذكر إحصائية بلا مصدر. «قواعد صارمة: (جزء من skill):  
- لا تستشهد بمصادر غير موثوقة.» - ميز بين حقيقة وتفسير. [يحتاج تحقق].

## • نصيحة احترافية

أقوى جملة ضد الهلوسة: «أفضل إجابة قصيرة دقيقة على إجابة طويلة مُخلقة.»

## تخصص سير العمل

Workflow Specialization — skills متخصصة بدقة

# 39

### • الأساس النظري

Skills العامة جيدة، Skills المتخصصة أفضل بكثير. بدل «skill كتابة»، أنشئ: «skill كتابة منشور LinkedIn تقني»، «skill كتابة newsletter أسبوعية»، «skill كتابة عنوان YouTube».

كلما زاد التخصص، زادت الدقة. التخصص يسمح بقواعد دقيقة، أمثلة محددة، أخطاء معروفة.

ابدأ عاماً، ثم انقسم. مع الوقت تكتشف أن «كتابة» تحتاج 5 skills مختلفة.

### • التطبيق العملي

1. خذ skill عام عندك.
2. طل آخر 10 مرات استخدمته فيها — هل كانت كلها لنفس الغرض الدقيق؟
3. غالباً ستجد تنوعاً — قسمه إلى 2-3 skills متخصصة.
4. جرب الـ skills المنقسمة لأسبوع.

#### • نصيحة احترافية

علامة skill ناجح: تقول كلمة-كلمتين فيؤثد مخرباً احترافياً. إن احتاج شرحاً طويلاً، التخصص ضعيف.

# لا تبدأ من الصفر

## 40

Stop from Scratch — استثمر في المكتبة

### • الأساس النظري

الدرس الأهم في فصل Skills: لا تبدأ كل مهمة من الصفر. كل مرة تكتب prompt جديد كاملاً، تُضيع فرصة لبناء أصل (asset).

اسأل نفسك بعد كل مهمة: «هل سأفعل هذا مرة أخرى؟» إن كان الجواب نعم، حوّل الـ prompt إلى skill / قالب.

بعد ستة أشهر من هذا الالتزام، سيكون عندك مكتبة قوية تُسرّع كل عملك.

### • التطبيق العملي

1. في نهاية كل أسبوع، راجع آخر 10 prompts كتبتها.
2. حدّد الـ 3 الأكثر تكراراً في طبيعتها.
3. حوّلها إلى skills / قوالب.
4. في الأسبوع التالي، استخدم القوالب بدل الكتابة من الصفر.
5. تتبّع الوقت الذي توفّره أسبوعياً.

#### • نصيحة احترافية

بعد سنة من هذا الالتزام، ستملك «نظام تشغيل شخصي» يجعلك أسرع بـ 10 أضعاف من غيرك في نفس المهام.



الفصل

# الخامس

## Claude كزميل عمل

تحويل Claude من أداة إلى شريك

الهدف من هذا الفصل

جعل Claude زميلك في العمل

## الفصل 5: Claude كزميل عمل

أكبر تحول في علاقتك مع Claude هو الانتقال من «أداة استخدمها» إلى «زميل أعمل معه». الزميل الجيد له أدوار متعددة: عاصف ذهني، استراتيجي، مراجع، باحث، محرر، معلم. هذا الفصل يُعلمك كيف تستخرج كل دور.

### الهدف من هذا الفصل

### جعل Claude زميلك في العمل

### خطوات هذا الفصل:

- 41 ما معنى أن Claude زميل؟
- 42 التعاون
- 43 Claude كمراجع
- 44 Claude كاستراتيجي
- 45 Claude كباحث
- 46 Claude كمحرر
- 47 Claude كمعلم
- 48 Claude كمخطط
- 49 تعدد الأدوار
- 50 الدعم العملياتي

## ما معنى أن Claude زميل؟

تحول العقلية — Coworker Mindset

# 41

### • الأساس النظري

الفرق بين «أداة» و «زميل»: الأداة تنفذ أمرك، الزميل يحاورك ويعارضك ويقترح. مع Claude، أنت تختار العلاقة. علامات التعامل كزميل: تطلب الرأي قبل الإنجاز، تقبل الاعتراض، تستفيد من «اختلاف وجهة النظر»، تستشير قبل القرار.

هذا التحول يأتي من طريقة كتابتك لـ prompt. «اكتب لي خطة» مقابل «ما رأيك في الخطة التالية؟ نقاط ضعفها؟ هل أنسيت شيئاً؟»

### • التطبيق العملي

1. في 5 محادثات قادمة، استبدل أول prompt من «افعل X» إلى «ما رأيك في X؟».
2. لاحظ كيف يتغير شكل الحوار.
3. اطلب اعتراضات، نقد، وجهات نظر مختلفة.
4. اعتد على ذلك كأسلوب افتراضي.

#### • نصيحة احترافية

زميل العمل الجيد يقول «لا» أحياناً. اسمح لـ Claude بذلك — اطلب صراحة أن يُخبرك إن كنت مخطئاً.

## التعاون

Collaborate — العمل المشترك خطوة بخطوة

## 42

## • الأساس النظري

التعاون الحقيقي ليس «أنا أطلب، هو ينفذ». بل: نقاش، اقتراح، تعديل، اتفاق، تنفيذ. خطوات التعاون: (1) شارك السياق كاملاً، (2) اطلب رأيه قبل العمل، (3) ناقش الخيارات، (4) اتفقا على المسار، (5) نتخذ معاً، (6) راجع النتيجة. هذا يبدو أطول من «اطلب فحسب»، لكنه يُنتج مخرجات أعلى جودة بكثير.

## • التطبيق العملي

1. خذ مشروعاً جديداً (مهما كان صغيراً).
2. بدل البدء بـ «اكتب لي X»، ابدأ بـ «أريد أن أكتب X. ما المعلومات التي تحتاج مني قبل البدء؟»
3. أجب عن أسئلته.
4. اطلب منه «أعطني خطة من 5 خطوات».
5. نفذ الخطوات معاً، خطوة بخطوة.

## • نصيحة احترافية

الـ «brief» الجيد في بداية المشروع يوفر 10 إعادات في النهاية.

## Claude كمراجع

Reviewer — مراجعة عمك بعين خبير

## 43

## • الأساس النظري

أحد أقوى أدوار Claude: المراجع. يمكنه مراجعة كتابتك، كودك، خطتك، قراراتك — بعين موضوعية لا تُجاملك. لتفعيل دور المراجع: عيّن الدور صراحة («أنت محرر صارم في مجلة The Economist»)، اطلب نقداً قاسياً، اطلب اقتراحات محددة. الفائدة: تكتشف نقاط ضعف قبل أن يكتشفها رئيسك أو عميلك.

## • التطبيق العملي

1. خذ نصاً كتبته (تقرير، رسالة، مقال).
2. اطلب: «أنت مراجع صارم بـ 20 سنة خبرة. راجع النص التالي، حدّد نقاط الضعف بصراحة، اقترح تحسينات محددة.»
3. طبّق نصف الاقتراحات على الأقل.
4. قارن النسختين.

## مثال على prompt

أنت محرر سابق في صحيفة كبرى. مراجعتك معروفة بالقسوة العادلة. راجع هذا المقال: حدّد 5 نقاط ضعف بصراحة، رشّح إعادة صياغة لأضعف فقرة، وأخبرني هل عنوان المقال يعمل أم لا.

## • نصيحة احترافية

اطلب «5 نقاط ضعف» لا «نقاط ضعف عامة» — العدد المحدد يُجبر التركيز.

## Claude كاستراتيجي

Strategist — التفكير الكبير

# 44

### • الأساس النظري

Claude يستطيع أن يلعب دور المستشار الاستراتيجي: يحلل، يتنبأ، يرى الصورة الكبيرة، يحذر من المخاطر. متى تستدعي الاستراتيجي؟ قبل قرار كبير: تغيير مهنة، إطلاق منتج، تعيين موظف، استثمار، انتقال جغرافي. كيف؟ شارك الوضع كاملاً، اطلب تحليلاً متعدد الأبعاد، اطلب سيناريوهات (أفضل، متوسط، أسوأ).

### • التطبيق العملي

1. اختر قرارًا تواجهه حالياً.
2. صفه بتفاصيل: الوضع، الخيارات، القيود، المخاوف.
3. اطلب من Claude: «حلل كاستراتيجي. ما الأبعاد التي لم أفكر فيها؟ ما المخاطر؟ ما السيناريوهات الثلاثة المحتملة؟»
4. استخدم تحليله كإحدى الزوايا في قرارك (لا كقرار نهائي).

#### • نصيحة احترافية

في القرارات الكبيرة، اطلب من Claude أن «يُعارضك» — يجد لك حجج ضد قرارك المفضل.

# Claude كباحث

## 45

Researcher — بحث منظم وعميق

### • الأساس النظري

مع تفعيل Web Search و Research mode، يصير Claude باحثاً قوياً. يجمع معلومات، يقارن مصادر، يلخص. للحصول على بحث جيد: حدد السؤال بدقة، اذكر زوايا الاهتمام، اطلب مصادر، حدد العمق. Research mode في Claude.ai يجري 10-30 بحث ألياً ويولد تقريراً منظماً.

### • التطبيق العملي

1. اختر موضوعاً تريد فهمه أعمق.
2. فعل Research mode (إن كان متاحاً).
3. اطرح سؤالاً بحثياً واضحاً.
4. قارن المخرج مع ما تعرفه — هل فيه معلومات جديدة؟ هل المصادر موثوقة؟

#### مثال على prompt

أريد بحثاً عن: «أثر العمل عن بُعد على إنتاجية المطورين في 2024-2025». الزوايا: دراسات أكاديمية، تقارير شركات، آراء قادة الصناعة. أعطني: خلاصة من نصف صفحة، 5 إحصائيات مفتاحية مع مصادرها، 3 وجهات نظر متعارضة.

#### • نصيحة احترافية

اطلب من Claude البحثي «وجهات نظر متعارضة» — يكشف الانحياز ويغني الفهم.

## Claude كمحرر

Editor — صقل اللغة والأسلوب

# 46

### • الأساس النظري

Editor مختلف عن Reviewer: المراجع يقيم، المحرر يُعدّل. تطلب من Claude إعادة كتابة، تشذيب، تحسين. أنواع التحرير: تحرير لغوي (نحو وإملاء)، تحرير أسلوب (الصيغة)، تحرير بنيوي (إعادة تنظيم)، تحرير حاد (تقصير 50% بلا فقد معنى). حدّد نوع التحرير المطلوب — كل نوع يُعطي نتيجة مختلفة.

### • التطبيق العملي

1. خذ نصاً (1000 كلمة).
2. اطلب «تحرير حاد — اختصره إلى 400 كلمة بدون فقد المعاني الجوهرية».
3. ثم اطلب «تحرير أسلوب — حسن صياغة الجمل بدون تغيير الطول».
4. قارن النسخ. لاحظ الفرق بين أنواع التحرير.

#### • نصيحة احترافية

اطلب «احتفظ بصوتي» في كل تحرير — يمنع Claude من تحويل النص إلى «صوته».

## Claude كمعلم

Teacher — التعلّم بطريقةك

## 47

## • الأساس النظري

Claude مدرس استثنائي، يُكيف الشرح لمستواك، يُجيب أسئلتك بلا ملل، يُعطي أمثلة مخصصة لاهتماماتك. لتعليم فعال: حدّد مستواك («أنا مبتدئ تماماً» / «لديّ خلفية في X»)، اطلب أمثلة من مجالك، اطلب اختبار فهمك. تقنية قوية: تقنية فاينمان — اطلب من Claude شرح المفهوم بأبسط طريقة ممكنة، كأنك في الـ 12 من العمر.

## • التطبيق العملي

1. اختر مفهوماً تريد فهمه (مثلاً: «التشفير غير المتماثل»).
2. ابدأ بـ: «اشرح لي كأنني في الـ 12 من العمر، باستخدام تشبيهات من الحياة اليومية».
3. بعد فهم الأساس، ارتق: «الآن اشرح لي كأنني طالب جامعي في علوم الحاسب».
4. اطلب اختبار فهمك: «اطرح عليّ 5 أسئلة، وصحّ إجاباتي».

## مثال على prompt

أريد أن أتعلّم مفهوم «الفائدة المركبة». مستواي: أعرف الرياضيات الأساسية، لم أدرس مالية. علّمني عبر: (1) شرح بسيط بمثال، (2) المعادلة وكيف تعمل، (3) لماذا تُسمّى «الأعجوبة الثامنة»، (4) اختبرني بـ 3 أسئلة تطبيقية.

## • نصيحة احترافية

في نهاية كل جلسة تعلّم: «لخص لي ما تعلّمته اليوم في 5 نقاط» — يُرسّخ المعلومة.

## Claude كمخطط

Planner — تنظيم المشاريع والأهداف

# 48

### • الأساس النظري

تخطيط المشاريع، الأهداف، اليوم، الأسبوع — كلها يجيدها Claude. الفرق بينه وبين تطبيق tasks: يفهم السياق، يقترح أولويات، يحذر من تعارضات.

للحصول على خطة جيدة: شارك الهدف، القيود (وقت، ميزانية، موارد)، الأولويات، التبعيات. اطلب الخطة بصيغة قابلة للتنفيذ: مهام محددة، مواعيد، معايير إنجاز.

### • التطبيق العملي

1. اختر مشروعاً ستبدأه هذا الشهر.
2. صفه لـ Claude: الهدف النهائي، الوقت المتاح، الموارد، أي قيود.
3. اطلب: «خطة من 4 أسابيع، مقسمة لمهام أسبوعية، كل مهمة معها وقت تقريبي».
4. نفذ الأسبوع الأول كما خطط، راجع، عدل الخطة.

#### • نصيحة احترافية

بعد كل أسبوع، أعط Claude تحديثاً («أنجزت X، تأخرت في Y») واطلب تعديل الخطة. الخطة الحية أقوى من خطة جامدة.

## تعدد الأدوار

## 49

Multi-role — Claude يلبس قبعات متعددة في محادثة واحدة

## • الأساس النظري

في مشروع معقد، تحتاج Claude يلبس عدة قبعات: استراتيجي في البداية، باحث في المنتصف، محرر في النهاية.

يمكنك أن تطلب صراحة: «الآن تصرف كـ X»، «الآن غير الدور إلى Y». هذا أفضل من بدء محادثات منفصلة — يحتفظ بالسياق الكامل.

## • التطبيق العملي

1. خذ مشروعاً مركباً (مثل: «إطلاق ورشة تدريبية»).
2. ابدأ المحادثة: «أنت استراتيجي. هل الفكرة قابلة للحياة؟ ما المخاطر؟»
3. بعد قراره: «الآن تحول إلى مخطط. صمّم محتوى الورشة في 6 ساعات.»
4. ثم: «الآن أنت كاتب إعلانات. اكتب صفحة هبوط للتسجيل.»
5. ثم: «الآن أنت محرر. راجع كل ما أنتجناه. أين الضعف؟»

## • نصيحة احترافية

عند تغيير الدور، استخدم عبارات صريحة («غير الدور الآن إلى...») — يُذكر Claude بالانتقال.

# الدعم العملياتي

## 50

Operational Support — العمل اليومي مع Claude

### • الأساس النظري

أعمق درجة من «Claude كزميل»: استخدامه في العمل اليومي العملياتي. ليس للمشاريع الكبرى فقط، بل في كل تفصيلة.

أمثلة: ردّ بريد سريع، صياغة Slack، تلخيص اجتماع، كتابة تعليق على PR، إعداد جدول أعمال، تحضير عرض. هذا التحول من «استخدام أسبوعي» إلى «استخدام بكل ساعة» يُضاعف الأثر.

### • التطبيق العملي

1. خصّ يومين كاملين: استخدم Claude في كل مهمة كتابية، حتى الصغيرة.
2. كل بريد، كل رسالة، كل تعليق — مرّره عبر Claude للمراجعة أو الصياغة.
3. قس: كم وقت وفرت؟ هل تحسّنت الجودة؟
4. بعد اليومين، حدّد المهام التي تستحق الـ Claude والتي لا تستحق.

#### • نصيحة احترافية

ليس كل شيء يستحق Claude. الإيميل من سطرين «نعم، أتفق» لا يحتاج الإيميل المهم لرئيسك يستحق.



الفصل

# السادس

## الموصلات (Connectors)

ربط Claude بعالمك الرقمي

الهدف من هذا الفصل

ربط Claude بعالمك

## الفصل 6: الموصّلات (Connectors)

Connectors هي الخطوة التي تُحوّل Claude من «جزيرة معزولة» إلى عقل متّصل بكل أدواتك. عندما يستطيع قراءة بريدك، تحرير ملفاتك، البحث في مستنداتك — يصير شريكاً حقيقياً.

الهدف من هذا الفصل  
ربط Claude بعالمك

### خطوات هذا الفصل:

- 51 ما هي الـ Connectors؟
- 52 Google Drive
- 53 GitHub
- 54 Notion
- 55 Slack
- 56 تحليل الملفات
- 57 تحليل جداول البيانات
- 58 Workflows المركزية
- 59 إعداد الأذونات
- 60 الوصول لبياناتك

# ما هي الـ Connectors؟

تعريف الموصلات

# 51

## • الأساس النظري

Connector هي «جسر» يربط Claude بخدمة خارجية (Slack, GitHub, Notion, Gmail, Drive...) فيستطيع قراءة المحتوى والعمل عليه مباشرة.

بدل أن تنسخ نص بريد إلكتروني وتلصقه في Claude، يقرأه هو مباشرة. بدل تحميل ملف Excel، يفتحه من Drive. Connectors تعمل بأذونات: أنت تختار ما يستطيع Claude رؤيته، وما لا يستطيع. الخصوصية تحت سيطرتك.

## • التطبيق العملي

1. في Connectors | Settings | Claude.ai، تصفح القائمة.
2. لاحظ التنوع: Jira, Asana, Linear, Slack, Notion, GitHub, Calendar, Gmail, Drive...
3. اقرأ وصف كل واحد لتفهم قدراته.

### • نصيحة احترافية

ابدأ بـ Connector واحد فقط — لا تربط 10 معاً. التركيز يُعمق الفهم.

## Google Drive

أول وأهم Connector

## 52

## • الأساس النظري

Google Drive Connector يتيح لـ Claude: قراءة Docs و Sheets و Slides، البحث في ملفاتك، تحرير المستندات (في بعض الحالات)، إنشاء ملفات جديدة.

هذا يُحوّل Claude إلى محرر مستندات حقيقي: «افتح ملف خطة Q4، أضف قسم المخاطر، احفظ.»  
 أمن: أنت تختار المجلدات المتاحة، يمكن إلغاء الربط في أي وقت، Claude لا يحفظ المحتوى بعد المحادثة.

## • التطبيق العملي

1. اربط Google Drive.
2. اطلب: «ابحث في Drive عن آخر تقرير اجتماع، ولخصه.»
3. اطلب: «أنشئ مستند جديد بعنوان «خطة الأسبوع» يحتوي على X.»
4. افتح Drive وتأكد أن الملف ظهر.

## مثال على prompt

من Drive: ابحث عن كل المستندات التي تحتوي على «استراتيجية 2026»، اعرض لي قائمة بالعناوين والمؤلفين وآخر تعديل، مرتبة من الأحدث.

## • نصيحة احترافية

Projects + Drive Connector = مكتبة بحث شخصية يمكنك سؤالها أي شيء.

## GitHub

لمطوري البرمجيات

## 53

## • الأساس النظري

GitHub Connector يُتيح لـ Claude: تصفح مستودعاتك، قراءة الكود، فحص PRs، قراءة issues، اقتراح تعديلات، للمطور: هذا يعني مساعد كود يفهم سياق مشروعك كاملاً، لا مجرد قطع متفرقة. أمثلة استخدام: «راجع آخر PR وأعطني ملاحظات»، «ابحث عن كل دوال تتعامل مع المستخدمين»، «اشرح بنية المشروع».

## • التطبيق العملي

1. اربط GitHub (لمستودعاتك العامة أو الخاصة).
2. اطلب: «اعرض آخر 5 commits في مستودع X، اشرح ماذا فعل كل واحد.»
3. اطلب: «ابحث في الكود عن استخدامات الدالة authenticate.»
4. اطلب مراجعة PR: «راجع PR رقم 42، حدّد المخاوف.»

## • نصيحة احترافية

GitHub + Claude Code = ثنائي قاتل للمطور. Connector يقرأ Code يكتب ويُصلح.

## Notion

ربط مساحة العمل

# 54

### • الأساس النظري

Notion Connector يجعل Claude يقرأ ويُحرر صفحاتك في Notion: قواعد البيانات، الويكي، الملاحظات، خطط المشاريع.

للشركات والفرق التي تعتمد Notion، هذا تحويل ضخم: Claude يصير مدرك سياق كل المشروع. يمكنك: استرجاع معلومات، إضافة صفحات، تحديث جداول، البحث الذكي في الكنز المعرفي.

### • التطبيق العملي

1. اربط Notion، اختر workspace.
2. اطلب: «ابحث في Notion عن كل صفحة فيها كلمة OKR.»
3. اطلب: «أضف صفحة جديدة في مساحة Marketing بعنوان «حملة يناير» تحوي [محتوى].»
4. اطلب: «من قاعدة بيانات Projects، اعرض المشاريع التي تأخرت عن موعدها.»

#### • نصيحة احترافية

Notion + Claude يحول نظام إدارة المعرفة من «مكتبة جامدة» إلى «مساعد ذكي».

# Slack

## 55

ربط محادثات الفريق

### • الأساس النظري

Slack Connector يُتيح لـ Claude: قراءة الرسائل، البحث في القنوات، تلخيص النقاشات، صياغة ردود. استخدامات: «لخص نقاش قناة #engineering اليوم»، «ابحث عن قرار اتخذ بشأن X»، «طُغ لي رداً على رسالة Y». ملاحظة خصوصية: تحقق من سياسات شركتك قبل ربط Slack الخاص بالعمل.

### • التطبيق العملي

1. اربط Slack شخصياً (مساحة عمل ليست حساسة).
2. اطلب تلخيص قناة لم تتابعها يومين: «لخص لي #general للأيام الثلاثة الماضية في 7 نقاط.»
3. اطلب: «ابحث عن آخر قرار اتخذ بشأن منتج X.»
4. تخيل كيف سيغير هذا حياتك إذا كنت في فريق كبير.

#### • نصيحة احترافية

أعظم استخدام لـ Slack Connector: تلخيص ما فاتك بعد إجازة أو يوم محموم.

## تحليل الملفات

File Analysis — معالجة المستندات بعمق

## 56

## • الأساس النظري

Claude يستطيع تحليل ملفات بأنواع متعددة: PDF, Word, Excel, PowerPoint, CSV, JSON, صور, أكواد. ارفع الملف، اطلب التحليل. Claude يستخرج المحتوى، يفهم البنية، يُحلل، يُلخص، يُجيب أسئلة. في PDF كبير (500 صفحة): اسأل Claude أي سؤال — سيجد الإجابة في الموضوع الصحيح ويستشهد به.

## • التطبيق العملي

1. ارفع PDF طويل (تقرير، كتاب، دراسة).
2. اطرح أسئلة محددة: «ما الفصل الذي يتحدث عن X؟ صفحة كم؟»
3. اطلب تحليلاً: «ما الحجج الرئيسية في الكتاب؟ ما النقاط الضعيفة؟»
4. اطلب جدولاً ملخصاً.

## مثال على prompt

هذا تقرير سنوي لشركة (PDF). استخرج: (1) الأداء المالي الرئيسي مع الأرقام، (2) المخاطر المذكورة، (3) الاستثمارات المستقبلية، (4) أي تغييرات إدارية. ضع كل قسم في جدول.

## • نصيحة احترافية

ارفع ملفات متعددة في محادثة واحدة لـ Claude يقارن بينها — قوة المقارنة هائلة.

## تحليل جداول البيانات

Spreadsheet Analysis — قوة الأرقام

## 57

## • الأساس النظري

Claude مع Code Execution يستطيع تحليل ملفات Excel/CSV بعمق: إحصاءات، تصور، تنظيف، تحويل. للأرقام، Code Execution أدق بكثير من «الحساب الذهني» للنموذج. Claude يكتب Python، ينتخذ، يُعطيك نتائج موثوقة. أمثلة: «اعرض لي توزيع المبيعات حسب المنطقة كرسم بياني»، «احسب متوسط فترة الإقامة لكل فئة عمرية».

## • التطبيق العملي

1. ارفع ملف CSV (بيانات حقيقية أو من الإنترنت).
2. فعل Code Execution.
3. اطلب تحليلاً متعدد الأبعاد.
4. اطلب رسوم بيانية.
5. اطلب تنبؤات أو ترشيحات بناءً على البيانات.

## • نصيحة احترافية

للأرقام: Code Execution مفعّل = ثقة. بدونه = اعتمد على Claude للوصف لا للحساب الدقيق.

## Workflows المركزية

Connectors — تنسيق متعدد الـ Centralized Workflows

## 58

## • الأساس النظري

القوة الحقيقية لـ Connectors تظهر عند تنسيقها معاً: قراءة بريد إلكتروني واستخراج المعلومة وحفظها في Notion لإنشاء مهمة في Asana وإخبار الفريق في Slack.

Workflow مثل هذا كان يتطلب Zapier أو كود مخصص. الآن: prompt واحد.

ابدأ بسير عمل صغير (Connector واحد + اثنين)، توسع تدريجياً.

## • التطبيق العملي

1. صمّم workflow بسيط (مثلاً: «اقرأ بريد العميل، استخرج المتطلبات، أنشئ مستند Notion، أضيف بنود في To-do»).
2. نفذه يدوياً مع Claude مرة.
3. حسّن الـ prompt حتى يصبح reliable.
4. بعدها يمكن أتمتته (الفصل التالي).

## • نصيحة احترافية

لا تبدأ بـ workflow معقد. workflow بسيط يعمل دائماً < workflow ذكي يُخفق نصف المرات.

## إعداد الأذونات

Permissions Setup — الأمان والخصوصية

## 59

## • الأساس النظري

كل Connector له أذونات قابلة للتعديل: ماذا يقرأ، ماذا يُحرر، ما الذي ممنوع رؤيته. أفضل ممارسة: مبدأ «أقل صلاحية ممكنة» (Least Privilege). لا تُعطي Claude صلاحيات لا يحتاجها فعلاً. راجع الأذونات شهرية، خاصة لـ Connectors لم تستخدمها فترة.

## • التطبيق العملي

1. ادخل Connectors Settings | اختر واحداً | Permissions.
2. اقرأ كل صلاحية مفعلة بدقة.
3. ألغ ما لا تحتاجه.
4. في Drive: اربط مجلداً محدداً، ليس كل Drive.
5. في Gmail: اربط labels محددة، ليس كل البريد.

## • نصيحة احترافية

في العمل: راجع سياسات شركتك. بعض Connectors قد تكون ممنوعة على بيانات العمل الحساسة.

## الوصول لبياناتك

Access Your Data — كنزك المعرفي تحت يدك

## 60

## • الأساس النظري

بعد ربط Connectors، تصير كل بياناتك الرقمية «قاعدة معرفة» يمكن سؤالها بلغة طبيعية.

لم تعد تبحث في 5 تطبيقات عن معلومة — تسأل Claude، يبحث في كلها معاً.

هذا تحول في طريقة تنظيم العمل: «بحث ذكي» يحلّ محلّ «أرشفة دقيقة».

## • التطبيق العملي

1. اطرح سؤالاً يحتاج بحثاً في عدة مصادر: «أعطني كل ما يخصّ مشروع X — من Drive, Notion, Slack, Gmail».
2. لاحظ كيف يجمع Claude المعلومات من مصادر متعددة.
3. اطلب ملخصاً موحّداً.

## • نصيحة احترافية

الميزة الحقيقية: لا حاجة لتذكر أين خزنت المعلومة. كل ما في عالمك الرقمي = ذاكرة قابلة للاستعلام.



الفصل

# السابع

## الأتمتة (Automation)

اجعل Claude يعمل بينما أنت نائم

الهدف من هذا الفصل

أتمتة الأعمال مع Claude

## الفصل 7: الأتمتة (Automation)

بعد إتقان الاستخدام الواعي، الخطوة التالية: جعل Claude يعمل من تلقاء نفسه. الأتمتة تحول ساعات أسبوعية إلى دقائق، وتضع روتينك على autopilot.

### الهدف من هذا الفصل

### أتمتة الأعمال مع Claude

### خطوات هذا الفصل:

- 61 ما هو MCP؟
- 62 لماذا MCP مهم؟
- 63 AI Agents
- 64 ربط سلاسل الأتمتة
- 65 أساسيات Claude API
- 66 Make g Zapier
- 67 Make.com — الأتمتة المرئية
- 68 أنظمة الذاكرة
- 69 Guardrails (حواجز الحماية)
- 70 الـ AI المستقل

## ما هو MCP؟

Model Context Protocol — البروتوكول الذي يحكم كل شيء

# 61

### • الأساس النظري

الأدوات الخارجية. MCP (Model Context Protocol) هو معيار مفتوح طورته Anthropic، يحدد كيف تتواصل النماذج اللغوية مع

قبل MCP: كل دمج بين Claude وأداة يحتاج تطوير مخصص. مع MCP: أي أداة تدعم البروتوكول تعمل مع Claude مباشرة.

MCP يعني: أدوات قابلة لإعادة الاستخدام، مجتمع متنام من الـ servers، حرية التبديل بين النماذج.

### • التطبيق العملي

1. ابحث في Google عن «MCP servers list» — ستجد عشرات الـ servers الجاهزة.
2. اقرأ توثيق MCP الرسمي على [modelcontextprotocol.io](https://modelcontextprotocol.io).
3. تصفح بعض MCP servers (مثل: postgres, github, filesystem) — افهم ماذا تفعل.

#### • نصيحة احترافية

MCP هو «USB» للذكاء الاصطناعي — معيار موحد يفتح كل شيء.

## لماذا MCP مهم؟

الثورة الصامتة — Why MCP Matters

# 62

### • الأساس النظري

MCP يحزّر Claude من حدود واجهة الدردشة. مع Claude, MCP يمكنه: الوصول لقاعدة بيانات شركتك، تشغيل أوامر terminal، استدعاء APIs مخصصة، التعامل مع أدوات داخلية لا تدخل ضمن Connectors الرسمية. للمؤسسات: MCP يعني إمكانية بناء «AI داخلية» تتكامل مع كل أنظمتك بأمان. للأفراد المتقدمين: تستطيع كتابة MCP server بسيط (بـ Python أو Node) يربط Claude بأي شيء تتخيله.

### • التطبيق العملي

1. اقرأ قصة شركة استخدمت MCP لربط Claude بأنظمتها الداخلية.
2. تخيل: لو كان عندك MCP يربط Claude بكل بياناتك، ماذا ستطلب منه؟
3. اكتب قائمة بـ 5 تكاملات تتمناها.

#### • نصيحة احترافية

MCP هو ما يجعل «Claude» vendor-agnostic — تستطيع تبديل النماذج دون فقدان التكاملات.

## الـ AI Agents

وكلاء الذكاء الاصطناعي — الجيل القادم

# 63

### • الأساس النظري

AI Agent هو نموذج لا يجاوب فحسب، بل ينقذ خطوات متتالية لتحقيق هدف. يتخذ قرارات، يستخدم أدوات، يتعلم من النتائج.

مثال: agent يبحث عن طيران، يقارن أسعار، يحجز، يضيف للتقويم، يرسل بريد تأكيد — كل ذلك من prompt واحد. Claude يعمل كـ agent عبر: Claude in Chrome (متصفح)، Cowork (سطح المكتب)، Claude Code (مطور برمجيات).

### • التطبيق العملي

1. جرب Claude في Chrome (إذا متاح لحسابك): اطلب مهمة تتطلب عدة خطوات.
2. لاحظ كيف يفتح صفحات، يقرأ، يقرر التالي.
3. هذا مستقبل العمل — تتفرج بينما يعمل النموذج.

#### • نصيحة احترافية

Agents قوية لكنها قد تخفق في مهام دقيقة. ابقَ مراقباً، خاصة في القرارات المالية أو الحساسة.

## ربط سلاسل الأتمتة

Automation Chaining — workflows متتالية

# 64

### • الأساس النظري

Automation Chaining: ربط مهام متعددة بحيث ينتج الأول مدخل الثاني، والثاني للثالث، إلخ. مثال: «كل صباح، اقرأ بريد العمل، استخرج المهام، أضفها لـ Todoist، أرسل لي ملخصاً في Telegram». أدوات تساعد: n8n (مفتوح)، Make، Zapier — كلها تدعم استدعاء Claude API ضمن سلاسل آلية.

### • التطبيق العملي

1. اختر مهمة روتينية متعددة الخطوات (مثلاً: «معالجة فواتير العملاء»).
2. ارسم سلسلة الأتمتة على ورقة: ماذا يحدث أولاً، ثم ثانياً، إلخ.
3. في n8n أو Make، ابن هذه السلسلة باستخدام Claude في الخطوات اللغوية.
4. اختبرها يدوياً، ثم اجعلها تعمل تلقائياً.

#### • نصيحة احترافية

ابدأ بسلسلة من 3 خطوات، كل خطوة تنجح = ثقة لإضافة الرابعة.

# أساسيات Claude API

Claude API Basics — البرمجة مع Claude

## 65

### • الأساس النظري

Claude API يُتيح لك دمج Claude في أي برنامج تكتبه. ترسل طلب HTTP، يرد Claude، تستخدم الرد. أساسي للمطورين: JavaScript، Python، أي لغة تدعم HTTP. مكتبات Anthropic الرسمية تسهل الأمر. تكلفة API: تدفع حسب الاستخدام (per token). أرخص بكثير من اشتراك شهري لاستخدامات محددة.

### • التطبيق العملي

1. أنشئ حساب على [console.anthropic.com](https://console.anthropic.com).
2. احصل على API key (احفظه بأمان!).
3. نزل مكتبة Anthropic (pip install anthropic).
4. اكتب أبسط برنامج: يأخذ سؤالاً من المستخدم، يرسله لـ Claude، يطبع الإجابة.
5. هئي نفسك — كتبت تطبيق AI بـ 10 أسطر.

#### مثال على prompt

```
python
from anthropic import Anthropic
client = Anthropic()
message = client.messages.create(
    model='claude-sonnet-4-6',
    max_tokens=1024,
    messages=[
        {'role': 'user', 'content': 'مرحباً!'}
    ]
)
print(message.content[0].text)
```

#### • نصيحة احترافية

احفظ API key في (environment variable ANTHROPIC\_API\_KEY)، لا تكتبه داخل الكود مباشرة.

## Make و Zapier

أتمتة بلا كود

# 66

### • الأساس النظري

Make و Zapier منصتان للأتمتة بدون كود. تدعمان Claude مباشرة كـ «action» في سلاسلهم.

مثال Zapier: trigger «وصل بريد جديد بعنوان action X» «أرسل المحتوى لـ Claude للتلخيص» action «احفظ الملخص في Notion».

بدون سطر كود واحد، تبني أتمتة قوية.

### • التطبيق العملي

1. أنشئ حساب Make أو Zapier (الخطة المجانية كافية للبداية).
2. ابحث عن أمثلة جاهزة (Zap templates) تستخدم Claude.
3. انسخ زاب جاهز، عدله لاحتياجك.
4. اربط Gmail (مثلاً) بـ Claude بـ Notion في زاب من 3 خطوات.

#### • نصيحة احترافية

ابدأ مع زاب «يخدمك أنت» قبل أتمتة عمل الفريق. تتعلم أسرع عندما المستفيد أنت.

# Make.com — الأتمتة المرئية

Make — سلاسل بصرية معقدة

# 67

## • الأساس النظري

Make.com (سابقاً Integromat) أقوى من Zapier في السيناريوهات المعقدة، أرخص في الاستخدام الكثيف، وله واجهة بصرية ممتازة.

يدعم: شرطيات، طقات، معالجة أخطاء، فلاتر — كلها مفيدة في سلاسل Claude متقدمة. للمتقدمين: استخدم Make لبناء agents كاملة بلا برمجة.

## • التطبيق العملي

1. أنشئ حساب Make.
2. ابن سيناريو: trigger من Claude Webhook يحتل النص شرط (إذا كانت النتيجة «إيجابي»، أرسل لـ X؛ وإلا أرسل لـ Y).
3. اختبر بإرسال أنواع نصوص مختلفة.

### • نصيحة احترافية

Make + Claude API + Webhooks = قدرة بناء «mini-SaaS» شخصية في ساعة.

# أنظمة الذاكرة

Memory Systems — ذاكرة طويلة الأمد

## 68

### • الأساس النظري

في الأتمتة المتقدمة، تحتاج Claude يتذكر معلومات بين الاستدعاءات. الذاكرة الافتراضية للمحادثة لا تكفي. طول: قاعدة بيانات (Postgres, Supabase)، خدمات (Pinecone, Weaviate) للبحث الدلالي، أو Notion كقاعدة معرفة.

نمط شائع: قبل كل استدعاء، اجلب «الذاكرة ذات الصلة»، ضعها في الـ prompt، نتذ.

### • التطبيق العملي

1. اختر مشروعاً يحتاج ذاكرة (مثلاً: chatbot يتذكر تفضيلات المستخدم).
2. صمّم schema بسيط لقاعدة بيانات تخزن المعلومات.
3. في كل استدعاء Claude، اجلب آخر 5 تفاعلات و ضعها في context.
4. اختبر — هل يتذكر؟

#### • نصيحة احترافية

للذاكرة البسيطة: Notion API كافٍ ومجاني. للمتقدمين: Supabase + pgvector.

## Guardrails (حواجز الحماية)

حماية الأنظمة الآلية

# 69

### • الأساس النظري

في الأنظمة الآلية، Claude يعمل بدون رقابتك المستمرة. تحتاج «guardrails» (حواجز حماية) تمنع الأخطاء. أمثلة: قواعد على الميزانية (لا تصرف أكثر من X)، فلاتر للمحتوى (لا تنشر بدون مراجعة)، حدود التكرار (لا ترسل أكثر من 5 رسائل لنفس الشخص يومياً). تطبيق guardrails: في الكود، في ال-prompt، في schema الإخراج، أو في تحقق بشري عند نقاط حرجة.

### • التطبيق العملي

1. لكل أتمتة عندك، اكتب قائمة بـ «أسوأ السيناريوهات».
2. لكل سيناريو، صم guardrail.
3. نفذ في الكود (شرطيات قبل التنفيذ).
4. اختبر بإدخالات سيئة عمداً — هل تتوقف الأتمتة بشكل صحيح؟

#### • نصيحة احترافية

أهم «human in the loop»: «guardrail» للقرارات الحرجة. لا تجعل Claude يُرسل بريدًا مالياً بدون مراجعة.

## الـ AI المستقل

Autonomous AI — حدود الأتمتة الكاملة

## 70

## • الأساس النظري

AI المستقل = نظام يعمل بدون تدخل بشري لفترات طويلة. هدف نهائي لكثير من المهام، لكنه ليس للجميع.

متى يصلح؟ مهام متكررة، محددة بدقة، منخفضة المخاطر، مع آليات مراقبة.

متى لا يصلح؟ قرارات إبداعية، تفاعل عميق مع البشر، مواقف غير متوقعة، مخاطر عالية.

## • التطبيق العملي

1. حدد مهمة واحدة عندك مرشحة للأتمتة الكاملة (مثلاً: «تصنيف الإيميلات الواردة وتلخيصها يومياً»).
2. صمّم نظاماً مع: مراقبة، تنبيهات عند الخطأ، مراجعة أسبوعية للأداء.
3. نفذ، تابعه لشهر.
4. هل أنقذ وقتاً؟ هل أحدث مشاكل؟

## • نصيحة احترافية

الذكاء الاصطناعي «المستقل تماماً» سراب في 2026. «شبه مستقل مع مراجعة دورية» هدف واقعي وأكثر أماناً.



الفصل

# الثامن

## Claude Code

الأداة التي تكتب الكود بدلاً عنك

الهدف من هذا الفصل

استخدام Claude في البرمجة الحقيقية

# الفصل 8: Claude Code

Claude Code هي الأداة التي حولت Claude من «مساعد برمجة» إلى «مهندس برمجيات افتراضي». يعمل من الـ terminal, يفهم مشاريعك كاملة, ينتهز مهام معقدة من دفع كود لإصلاح bugs.

## الهدف من هذا الفصل

## استخدام Claude في البرمجة الحقيقية

### خطوات هذا الفصل:

- 71 نظرة عامة على Claude Code
- 72 اشرح الكود ببساطة
- 73 تصحيح الأخطاء
- 74 إعادة هيكلة الكود
- 75 مشاريع Full Stack
- 76 كتابة التوثيق
- 77 الاختبارات والتحليل
- 78 أفضل الممارسات
- 79 نصائح لسير العمل التطويري
- 80 أرسل كوداً أفضل

## نظرة عامة على Claude Code

Claude Code Overview — ما هي وكيف تختلف

# 71

### • الأساس النظري

Claude Code هي أداة سطر أوامر (CLI) تتيح للمطورين تفويض مهام برمجية إلى Claude مباشرة من terminal. ما يميزها عن واجهة Claude العادية: تعمل في مجلد المشروع، تقرأ كل الملفات بنفسها، تعدل وتنشئ ملفات، تشغل أوامر، تختبر، تدفع لـ Git.

الفرق من ChatGPT: Claude Code Copilot هو agent حقيقي — تقول له «أصلح bug رقم 42»، فيقرأ الكود، يفهم المشكلة، يكتب الإصلاح، يختبره، يعمل commit.

### • التطبيق العملي

1. تأكد من تثبيت Node.js (مطلوب).
2. ثبت Claude Code: `npm install -g @anthropic-ai/claude-code`
3. في مجلد مشروع، شغل: `claude``
4. ستبدأ محادثة في terminal — اطلب أي شيء.

#### • نصيحة احترافية

Claude Code أقوى ميزة لمطور في 2026. إن لم تجربها بعد، توقف الآن وثبتها.

# اشرح الكود ببساطة

— Explain Code Simply — فهم الأكواد غير المألوفة

# 72

## • الأساس النظري

أول استخدام عملي: فهم كود لم تكتبه. مشروع مفتوح المصدر، codebase شركتك القديم، library لا تعرفها. Claude Code يقرأ المشروع كاملاً، يفهم البنية، يشرح بلغتك المفضلة. اسأل: «اشرح لي ماذا تفعل دالة X»، «ما الـ flow عند تسجيل دخول مستخدم؟»، «لماذا هذا الملف موجود؟»

## • التطبيق العملي

1. افتح مشروع مفتوح المصدر (clone أي repo شائع).
2. شغل Claude Code في المجلد.
3. اطرح أسئلة: «اشرح بنية المشروع»، «ما الذي يفعله entry point؟»
4. ستفهم في 30 دقيقة ما كان يحتاج أياماً.

### • نصيحة احترافية

هذا أحد أعظم تطبيقات Claude للمطورين الجدد في فرق كبيرة: «اشرح لي الـ codebase» = شهور من الـ onboarding في ساعة.

## تصحيح الأخطاء

bugs — صيد الـ Debug Errors &amp; Fixes

## 73

## • الأساس النظري

Debugging تقليدياً = ساعات من البحث في الـ stack trace. مع Claude Code: ألصق الخطأ، يقرأ الكود ذي الصلة، يقترح الإصلاح.

أحياناً يكتشف بـ bugs لم تراها — يقرأ الكود بعين «نموذج» لا «إنسان متعب». لإصلاح متقدم: «أصلح الخطأ ثم اشرح ما كان السبب الجذري، حتى لا أكرره».

## • التطبيق العملي

1. اختر bug صعب في مشروع لك.
2. أعط Claude Code: الخطأ، الـ stack trace، الكود ذو الصلة.
3. اطلب الإصلاح + الشرح.
4. تحقق من فهمك للإصلاح قبل تطبيقه.

## • نصيحة احترافية

لا تقبل إصلاحاً لا تفهمه. اطلب شرحاً. الكود الذي لا تفهمه = bug مستقبلي.

# إعادة هيكلة الكود

## 74

Refactor Code — تحسين بلا تغيير الوظائف

### الأساس النظري

Refactoring = إعادة كتابة الكود ليصبح أنظف، أسرع، أسهل صيانة — بدون تغيير ما يفعل.  
 Claude Code ممتاز هنا: «اقرأ هذا الكود الفوضوي، اقترح refactoring يحافظ على نفس السلوك».  
 أمثلة: استخراج functions، تقسيم classes كبيرة، إزالة تكرار، تحديث لـ patterns حديثة.

### التطبيق العملي

1. اختر ملفاً في مشروعك تشعر أنه «فوضوي».
2. اطلب: «اقترح refactoring يحسن قابلية القراءة والصيانة، احتفظ بنفس الوظائف، اشرح كل تغيير».
3. راجع الاقتراحات بعقل ناقد — Claude قد يبالغ.
4. طبق ما يستحق فقط.

#### نصيحة احترافية

Refactor تدريجياً، لا دفعة واحدة. كل refactor يُختبر قبل التالي.

# مشاريع Full Stack 75

تطبيقات كاملة — Full Stack Projects

## الأساس النظري

Claude Code يبني مشاريع كاملة: frontend + backend + قاعدة بيانات + auth + deployment. مثال: «أنشئ تطبيق to-do بـ Next.js و Supabase. الميزات: تسجيل دخول بـ CRUD، Google للمهام، deploy على Vercel».

في ساعة، يكون عندك تطبيق كامل قيد التشغيل. هذا ليس مبالغة.

## التطبيق العملي

1. اختر فكرة تطبيق بسيط لطالما أردته.
2. اطلب من Claude Code بناءه — كن تفصيلياً في المتطلبات.
3. تابع المسيرة: اقرأ ما يفعل، تعلم منه، اطلب تعديلات.
4. deploy على Vercel/Netlify — شارك الرابط مع أصدقاء.

### نصيحة احترافية

حتى لو لست مطوراً، Claude Code يفتح لك بناء تطبيقات، خذها بجدية — إنها قدرة عظيمة.

## كتابة التوثيق

Write Documentation — التوثيق الذي يكرهه الكل

# 76

### • الأساس النظري

أكثر مهمة يكرهها المطورون: كتابة التوثيق. Claude Code يحلّ هذه المشكلة جذرياً. يقرأ الكود، يفهم ما يفعل، يكتب API docs، README، شرح كل function، أمثلة استخدام. النتيجة: توثيق منظم، شامل، أنظف من توثيق كثير من المشاريع الشهيرة.

### • التطبيق العملي

1. اختر مشروعاً عندك بلا توثيق (الأغلب!).
2. اطلب: «أنشئ README متكامل: ما المشروع، التثبيت، الاستخدام، API، أمثلة، contribution guide.»
3. راجع، عدّل، ادفع.
4. كرر للمشاريع الأخرى.

#### • نصيحة احترافية

اطلب التوثيق بعد كل ميزة جديدة، ليس في النهاية. الـ docs المترجمة كابوس.

## الاختبارات والتحليل

Unit Tests & Analysis — كود يختبر نفسه

# 77

### • الأساس النظري

الاختبارات = الفرق بين كود هواة وكود محترفين. Claude Code يكتبها لك.

اطلب: «اكتب unit tests لهذه الـ function، غط: الحالة العادية، الحالات الحدية، الأخطاء المتوقعة، edge cases».

أيضاً: «حلل الكود — أين النقاط الضعيفة؟ ما الـ security vulnerabilities؟ ما الـ performance issues؟»

### • التطبيق العملي

1. اختر module من مشروعك بلا اختبارات.
2. اطلب Claude Code يكتب اختبارات شاملة.
3. شغل الاختبارات — هل تمر؟
4. اطلب «coverage analysis» — كم نسبة الكود المغطى؟

#### • نصيحة احترافية

الاختبارات الجيدة = ثقة. مع Claude Code، لا عذر لعدم كتابتها.

# أفضل الممارسات

## 78

Best Practices — كود نظيف ومعياري

### • الأساس النظري

Claude Code مدرب على أفضل الممارسات الحديثة لكل لغة: JavaScript (ESLint), Python (PEP 8, type hints), modern patterns), إلخ.

اطلب صراحة: «اتبع best practices للغة X»، أو ضع ملف editorconfig و linting في المشروع — يلتزم بها. بهذه الطريقة، تتعلم best practices أثناء البناء.

### • التطبيق العملي

1. أعط Claude Code قطعة كود قديم.
2. اطلب: «أعد كتابتها وفق best practices الحديثة للغة X. اشرح كل تغيير.»
3. كل شرح = درس مجاني في البرمجة الجيدة.
4. كرر مع لغات/أطر مختلفة.

#### • نصيحة احترافية

في كل مرة يشرح لك Claude best practice، احفظها في ملف «دروسي من Claude». بنك معرفي شخصي.

# نصائح لسير العمل التطويري

التكامل مع Git و CI/CD — Dev Workflow Tips

## 79

### • الأساس النظري

Claude Code يدمج مع Git: يكتب commit messages معيارية، يفتح PRs، يكتب وصف الـ PR بناءً على التغييرات. أوامر مفيدة: «اعمل commit للتغييرات الحالية برسالة منظمة»، «أنشئ PR ضد main مع وصف كامل»، «راجع آخر 5 commits — هل هناك مشاكل؟» للـ CI/CD: يكتب GitHub Actions، يصمم workflows، يصلح builds فاشلة.

### • التطبيق العملي

1. في مشروع Git: اطلب «اعمل commit ذكي للتغييرات الحالية».
2. اطلب «أنشئ ملف GitHub Actions لتشغيل tests تلقائياً».
3. اطلب «راجع آخر فشل في CI، اكتشف السبب، اقترح إصلاح».

#### • نصيحة احترافية

أعط Claude Code وصول لـ `git commits log` — يفهم تاريخ المشروع، يكتب رسائل أحدث مألوفة مع الأسلوب.

# أرسل كوداً أفضل

الإنتاجية الحقيقية — Ship Better Code

# 80

## • الأساس النظري

الإنتاجية الحقيقية ليست «أكتب كود أكثر»، بل «أرسل كود أفضل، أسرع». Claude Code يُحقق ذلك: تخطط أقل، تنفذ أكثر. تختبر أكثر، تصلح bugs أقل. توثق بكثرة، تنسى أقل. تتعلم باستمرار، تتطور دائماً. بعد أشهر مع Claude Code، ستلاحظ: شحنك للكود تضاعف، عدد bugs انخفض، استمتاعك بالعمل ازداد.

## • التطبيق العملي

1. تتبع 30 يوماً مقاييسك: كم feature أنجزت؟ كم bug أصلحت؟ كم ساعة فعلت ذلك؟
2. قارن مع 30 يوم قبل Claude Code (إذا تذكرت).
3. غالباً ستجد قفزة 2-5x في الإنتاجية.

### • نصيحة احترافية

أعظم مكسب من Claude Code ليس السرعة، بل الجرأة. أمور كنت تؤجلها لتعقيدها، الآن تبدأها لأنها صارت سهلة.



الفصل

# التاسع

## حالات الاستخدام المتقدمة

Claude في كل مجالات الحياة والعمل

الهدف من هذا الفصل

استخدام Claude في الحياة الحقيقية

# الفصل 9: حالات الاستخدام المتقدمة

بعد إتقان الأدوات، يأتي السؤال: في ماذا أستخدمها؟ هذا الفصل يفتح أمامك أبواباً متعددة — كل خطوة هي عالم كامل من الاستخدامات، اختر ما يهيك، وتعمق فيه.

## الهدف من هذا الفصل

### استخدام Claude في الحياة الحقيقية

#### خطوات هذا الفصل:

- 81 Prompting متقدم
- 82 سير العمل المخصص
- 83 Multi-Agent Systems
- 84 تكاملات متقدمة
- 85 أنظمة قابلة للتوسعة
- 86 ضبط الأداء
- 87 Deep Research Ops
- 88 أتمتة الأعمال
- 89 ابن أنظمتك الخاصة
- 90 Master Claude

# Prompting متقدم 81

## تقنيات النخبة — Advanced Prompting

### • الأساس النظري

بعد إتقان الأساسيات، حان وقت التقنيات المتقدمة: Few-shot learning, Self-consistency, Tree of Thoughts, Reflection.

Few-shot: أعط النموذج 2-3 أمثلة من النوع المطلوب قبل طلبك. الأمثلة تُعلمه أكثر من الشرح.

Self-consistency: اطلب نفس المهمة 3 مرات بطرق مختلفة، اختر الإجابة المشتركة. يقلل الأخطاء العشوائية.

Tree of Thoughts: اطلب من Claude يستكشف عدة مسارات تفكير قبل اختيار الأفضل.

Reflection: بعد إجابة، اطلب «راجع إجابتك، حدد نقاط الضعف، أعد المحاولة».

### • التطبيق العملي

1. خذ مهمة معقدة (مثل: «صنّف هذه التغريدة عاطفياً»).
2. جرّب Few-shot: أعط 3 أمثلة قبل سؤالك.
3. قارن مع طلب بدون أمثلة.
4. الفرق ملحوظ — Few-shot يُعلم النموذج «على الطائر».

#### • نصيحة احترافية

Few-shot هو أسرع طريقة لتعليم Claude مهمة جديدة بدون skill كامل.

# سير العمل المخصص

## 82

Custom Workflows — تصميم لعملك تحديداً

### • الأساس النظري

كل مهنة لها أسلوب عمل مختلف. المحامي يحتل العقود، الطبيب يقرأ الأبحاث، المعلم يصمم الدروس، المصمم يطور الـ moodboards.

صمم workflow خاصاً بمهنتك: ما الخطوات؟ ما المدخلات؟ ما المخرجات؟ في أي خطوة يدخل Claude؟ هذا أعلى مستوى من «استخدام Claude» — يصير جزءاً من نظامك المهني.

### • التطبيق العملي

1. اختر مهمة معقدة في مهنتك تتكرر شهرياً.
2. ارسم خطواتها على ورقة (5-10 خطوات).
3. حدد أي خطوات تتطلب «معالجة لغوية» — هذه فرص Claude.
4. صمم prompts/skills لكل خطوة.
5. نفذ workflow في المرة القادمة — قس التحسين.

#### • نصيحة احترافية

Workflow المخصص = إنتاجيتك x10. هو الفرق بين «أستخدم Claude» و«Claude جزء من نظامي».

## Multi-Agent Systems

أنظمة متعددة الوكلاء

# 83

### • الأساس النظري

في الأنظمة المتقدمة: لا تستخدم Claude واحداً، بل عدة instances كل واحد له دور.

مثال: Researcher Claude (يجمع المعلومات)، Analyst Claude (يحللها)، Writer Claude (يكتب التقرير)، Reviewer Claude (يراجع). كلهم يعملون معاً.

Frameworks: AutoGen, CrewAI, LangGraph تدعم هذا. تكتب «agent» لكل دور، تنسق بينهم.

### • التطبيق العملي

1. اقرأ توثيق CrewAI (مفتوح، يدعم Claude).
2. ابن نظاماً بسيطاً: 2 agents فقط — باحث وكاتب.
3. الباحث يجمع، الكاتب يكتب بناءً على ما جمع.
4. لاحظ كيف تختلف النتيجة عن Claude واحد يحاول الكل.

#### • نصيحة احترافية

Multi-agent ليس دائماً أفضل — أحياناً Claude واحد جيد كافٍ. استخدمه عند مهام تستفيد فعلاً من «أدوار منفصلة».

## تكاملات متقدمة

دمج عميق مع الأنظمة — Advanced Integrations

## 84

## • الأساس النظري

تكاملات متقدمة: ربط Claude بقواعد بيانات داخلية، أنظمة CRM، ERP، أدوات تحليلات، أنظمة دفع.

للشركات: هذا ما يُحوّل Claude من «أداة شخصية» إلى «بنية تحتية ذكية».

يتطلب: فهم MCP servers، APIs مخصصة، أمان عالٍ، حوكمة البيانات.

## • التطبيق العملي

1. للمطورين: حدّد نظاماً داخلياً، اكتب MCP server بسيط له.
2. للمستخدمين: ابحث عن MCP servers جاهزة لأدواتك (HubSpot, Salesforce, ...).
3. ابن تكاملاً واحداً قوياً، توسّع من هناك.

## • نصيحة احترافية

أعمق تكامل = أكبر أثر. سطحياً يفيد، عميقاً يُحدث ثورة.

## أنظمة قابلة للتوسعة

للأعمال الكبيرة — Scalable Systems

# 85

### • الأساس النظري

بناء نظام يخدم 10 مستخدمين سهل. خدمة 10,000 يحتاج تفكيراً مختلفاً. اعتبارات: تكلفة API (تضخم سريعاً)، latency, caching, queueing, fallbacks عند الفشل، تتبع وتحليل الاستخدام. نصيحة: استخدم Anthropic Batches API للمهام غير الفورية — أرخص بكثير.

### • التطبيق العملي

1. إن كنت تبني منتج؛ احسب تكلفة Claude لكل مستخدم.
2. صمّم caching: لا تستدع Claude لطلبات متطابقة.
3. أضف logging شامل لتعرف ماذا يحدث.
4. اختبر تحت ضغط (100+ طلب متزامن).

#### • نصيحة احترافية

Sonnet للأغلب، Haiku للمهام السريعة المتكررة، Opus للمهام الحرجة فقط. التوازن يؤثر آلاف الدولارات.

## ضبط الأداء

Performance Tuning — جودة + سرعة + تكلفة

## 86

## • الأساس النظري

ضبط الأداء = إيجاد التوازن المثالي بين 3 محاور: جودة المخرج، سرعة الاستجابة، التكلفة. تقنيات: prompt caching (يحفظ أجزاء ثابتة)، تقليل الـ tokens غير الضرورية، اختيار النموذج الأنسب لكل مهمة، streaming للتجربة الفورية. قياس: تتبع latency p50/p95/p99، تكلفة لكل request، رضا المستخدم.

## • التطبيق العملي

1. ابدأ بقياس: ما متوسط latency و cost لـ prompt مهم عندك؟
2. جرب تحسين واحد (مثل: تقصير الـ 30% system prompt).
3. قس الأثر — هل تحسنت السرعة؟ هل تأثرت الجودة؟
4. كرر التحسينات الناجحة.

## • نصيحة احترافية

Caching هو أكبر مكسب سهل. استخدم cache\_control في الـ API — قد يخفض تكلفتك 90%.

## Deep Research Ops

البحث العميق المنظم

# 87

### • الأساس النظري

Deep Research = تجاوز البحث السطحي إلى تحقيق صحفي / أكاديمي / استخباري.  
منهجية: تحديد سؤال | فرضيات | مصادر متنوعة | تقاطع المعلومات | بناء قصة | التحقق.  
Claude في Research mode يجري 10-30 بحث، يُولد تقرير منظم بمصادره.

### • التطبيق العملي

1. اختر سؤالاً معتاداً («كيف غيرت OpenAI سوق الذكاء الاصطناعي 2023-2025؟»).
2. فعل Research mode.
3. اطلب تقريراً من 10 صفحات بـ headings, citations, vation.
4. راجع الـ citations — هل المصادر جديّة؟

#### • نصيحة احترافية

للبحوث الجديّة: اطلب من Claude يستشهد بثلاث مصادر لكل ادعاء، لا واحدة. يقلل الهلوسة.

## أتمتة الأعمال

Business Automation — إعادة تصميم العمليات

# 88

### • الأساس النظري

Business Automation مع Claude لا يعني «استبدال موظفين» بل «إعادة تصميم العمليات».

أسئلة لتبدأ: ما المهام المتكررة في فريقك؟ ما التي تحتاج معالجة لغوية؟ ما القرارات التي تتبع قواعد واضحة؟ ابدأ بـ pilot صغير: مهمة واحدة، فريق صغير، شهر تجربة. قس الأثر، توسع.

### • التطبيق العملي

1. في عملك، اختر عملية تأخذ وقت +5 ساعات أسبوعياً.
2. طلها: أين الخطوات اللغوية / المتكررة؟
3. صمّم أتمتة جزئية (50-70% من العملية) مع مراجعة بشرية.
4. اختبر مع فريق صغير لمدة شهر.
5. قس: وقت موثّر، أخطاء، رضا الموظفين.

#### • نصيحة احترافية

أتمتة ناجحة = موظفون يفعلون ما يحبون، Claude يفعل الباقي. اشركهم في التصميم.

# ابن أنظمتك الخاصة

## 89

Build Your Systems — Claude في كل عملك

### • الأساس النظري

أعلى مستوى: لا تستخدم منتجاً جاهزاً — ابن نظامك الخاص حول Claude API. أمثلة: chatbot داخلي لشركتك، أداة تحليل خاصة بمجالك، تطبيق mobile مع Claude، منصة SaaS لعملاء، Anthropic SDK + Web framework (مثلًا) + قاعدة بيانات = أي تطبيق ذكاء اصطناعي تريده.

### • التطبيق العملي

1. حدد فكرة تطبيق تستخدم Claude.
2. ابن MVP في عطلة أسبوع (مع Claude Code).
3. اعرضه على 5 مستخدمين محتملين.
4. حسن بناءً على ردودهم.
5. أطلق — حتى لو لمجموعة صغيرة.

#### • نصيحة احترافية

أصعب جزء ليس التقنية — هو إيجاد فكرة تحلّ مشكلة حقيقية. ابحث عن «وجع» لتعالجه.

## Master Claude

إتقان Claude — أنت الآن في الـ 1%

# 90

### • الأساس النظري

إن أتممت كل الخطوات السابقة، أنت الآن من بين الـ 1% الأعلى في استخدام Claude. هذا لا يعني التوقف — يعني الانتقال إلى: تعليم الآخرين، الإسهام في المجتمع، البناء على ما تعلمت. Claude يتطور باستمرار. اشترك في تحديثات Anthropic، اقرأ release notes، جرب الميزات الجديدة فور إصدارها.

### • التطبيق العملي

1. اشترك في مدونة Anthropic.
2. تابع twitter/X الرسمي لـ Anthropic.
3. انضم لمجتمع r/ClaudeAI على Reddit.
4. علم شخصاً واحداً كل أسبوع شيئاً تعلمته.
5. ساهم في skill / template / tool مفتوح المصدر.

#### • نصيحة احترافية

أعظم درس: لا «نهاية» في تعلم AI. كل شهر يأتي بميزات جديدة. ابقَ فضولياً، تجرب باستمرار.



الفصل

# العاشر

## المستخدم الخارق

أن تصير نخبة النخبة

الهدف من هذا الفصل

أن تصير Power User حقيقياً

# الفصل 10: المستخدم الخارق

آخر 10 خطوات. هنا تنتقل من «مستخدم محترف» إلى «صانع نخبة». ليست تقنيات جديدة بقدر ما هي عقلية جديدة وممارسات راسخة تُميز النخبة الحقيقية.

الهدف من هذا الفصل

أن تصبح Power User حقيقياً

## خطوات هذا الفصل:

- 91 Prompting المتقدم جداً
- 92 Workflows مخصصة عميقاً
- 93 Multi-Agent Systems متقدمة
- 94 تكاملات متقدمة جداً
- 95 أنظمة قابلة للتوسعة
- 96 ضبط الأداء
- 97 Deep Research كمحترف
- 98 أتمتة الأعمال — مستوى تنفيذي
- 99 ابن أنظمتك
- 100 Master Claude — للأبد

# Prompting المتقدم جداً

Advanced Prompting — أبعد من الكتب

# 91

## • الأساس النظري

Power Users يكتبون prompts تبدو سحرية. سرّها: عقود الـ output (مخرج بصيغة JSON محددة), negative examples (أمثلة على ما لا تريده), meta-instructions (كيف يفكر النموذج). يستخدمون XML tags داخل الـ Claude — ( , , ) prompt مدرّب على فهمها بدقة.

يضبطنون temperature في 0 API للحسم, +0.7 للإبداع).

## • التطبيق العملي

1. أعد كتابة prompt مهم عندك باستخدام XML tags.
2. حدّد الـ output بصيغة JSON schema صريحة.
3. أضف negative example: «لا تُنتج شيئاً يشبه هذا».
4. اختبر — ستجد الإجابة أدق بكثير.

### نصيحة احترافية •

Anthropic تنشر prompts «أمثلة» لتطبيقاتها الخاصة على [docs.claude.com](https://docs.claude.com). ادرسها — تتعلم منها أكثر من أي كتاب.

## Workflows مخصصة عميقاً

Custom Workflows — نظامك المهني الكامل

# 92

### • الأساس النظري

Power User لا يستخدم workflow عام. عنده workflows مفصلة على مقاسه: workflow صباحي، workflow الكتابة، workflow الاجتماعات، workflow اتخاذ القرار.

كل workflow موثق: ما يدخل، ما الخطوات، ما يخرج. مع الوقت يصير «دستوراً شخصياً». بهذا الشكل: لا تفكر في «كيف أفعل هذا؟» — تتبع الـ workflow.

### • التطبيق العملي

1. هذا الأسبوع: وثق 3 workflows لمهام أساسية عندك.
2. كل واحد على شكل: output | tools | steps | trigger.
3. اطبعه أو اعرضه على Notion في مكان ظاهر.
4. اتبعه بدقة — حسنه عبر الاستخدام.

#### • نصيحة احترافية

بعد 6 شهر من workflows موثقة، يكون عندك «نظام تشغيل شخصي» يحسدك عليه الآخرون.

# Multi-Agent Systems متقدمة

Multi-Agent — التفكير كقائد فريق

# 93

## • الأساس النظري

Power User يبني أنظمة multi-agent تعمل بدونها: 5 agents يتعاونون على إنتاج تقرير أسبوعي، أو إدارة جدول، أو متابعة مشروع.

هذا يتطلب: تحديد أدوار بدقة، تصميم البروتوكول بينهم، آليات حل النزاع، human-in-the-loop عند نقاط حرجية. النتيجة: أنت تدير «فريقاً افتراضياً» يعمل بينما تنام.

## • التطبيق العملي

1. صمّم نظام multi-agent لمهنة عندك (مثلاً: «إدارة محتوى الـ LinkedIn»).
2. أدوار محتملة: researcher, writer, editor, scheduler, analyst.
3. ابن النظام بـ CrewAI أو LangGraph.
4. شغله أسبوعاً، ادرس مخرجاته.

### • نصيحة احترافية

نظام agents جيد < human متعب. لكن نظام agents سيء > human متعبه. الجودة هي كل شيء.

# تكاملات متقدمة جداً

## 94

Advanced Integrations — الذكاء في كل النظام

### • الأساس النظري

Power User لا يربط Claude بأداة — يربطه بكامل نظامه: الحاسوب، الهاتف، السيارة، البيت الذكي. أمثلة: «أول ما أصحو، Claude يلخص لي البريد بصوت عبر السماعة»، «عند دخول السيارة، يقترح أفضل طريق بناءً على تقويمي». يستخدم: Home Assistant، Apple Watch، Tasker (Android)، shortcuts (iOS).

### • التطبيق العملي

1. تخيل: لو كنت أنت تتحكم بكل ما تستخدمه يومياً، أين سيكون Claude مفيداً؟
2. ابن shortcut واحد: مثلاً «استدع Claude بضغطة على Apple Watch ليُلخص ما حدث اليوم».
3. وسّع شيئاً فشيئاً — اعمر بيئتك بـ Claude.

#### • نصيحة احترافية

أعمق تكامل: لا يصير «أداة تفتحها»، بل «حضور صامت يساعدك» في كل اللحظة.

# أنظمة قابلة للتوسعة

Scalable Systems — للمؤسسات

# 95

## • الأساس النظري

Power User يبني للمؤسسات: نظام يخدم 100+ موظف، يلتزم بـ compliance، يحافظ على بيانات حساسة.

اعتبارات: SSO, data residency, audit logs, rate limiting, monitoring, disaster recovery.

Anthropic Enterprise plans تقدم: SOC2, ZDR (Zero Data Retention), خصوصية كاملة.

## • التطبيق العملي

1. إن كنت في شركة: تعلم compliance المطلوب في مجالك.

2. اقرأ Anthropic Enterprise docs.

3. صمم نظاماً نظرياً يلبي متطلبات شركتك.

### • نصيحة احترافية

Power User في المؤسسات يجمع بين معرفة تقنية + فهم للأعمال + التزام بالأمان. مزيج نادر = قيمة عالية.

## ضبط الأداء

Performance Tuning — تحسين دقيق

## 96

## • الأساس النظري

في الأنظمة الكبيرة، كل  $1 \text{ millisecond}$  مهم. Power User يقيس، يحلل، يحسن بدقة. أدوات: Helicone، LangSmith لتتبع استدعاءات Claude. تحليل: متى Sonnet كافٍ بدل Opus؟ متى يمكن تقصير `prompt`؟ نتائج: تطبيق يخدم آلاف المستخدمين بتكلفة معقولة وسرعة جيدة.

## • التطبيق العملي

1. اربط Helicone بمشروع API عندك.
2. حمل البيانات لأسبوع.
3. اكتشف 3 تحسينات (نموذج، `prompt`، `cache`).
4. طبقها، قس الأثر.

## • نصيحة احترافية

كل  $1 \text{ ms}$  أقل في `latency` = تجربة مستخدم أفضل. كل  $1 \text{ cent}$  أقل في `cost` = هامش أعلى. الـ `marginal gains` تتراكم.

## Deep Research كمحترف

منهجية إعلامية / أكاديمية — Deep Research Ops

# 97

### • الأساس النظري

Power User في البحث: يدمج Claude مع databases أكاديمية (arXiv, PubMed, JSTOR)، أدوات scraping متقدمة، أنظمة citation management.

يبنى pipelines: قراءة 100 paper | استخراج معلومات منظمة | بناء قاعدة معرفة | جوازة أسئلة بحثية. هذا مستوى الباحثين الأكاديميين والصحفيين الاستقصائيين.

### • التطبيق العملي

1. اختر مجال بحث يثير فضولك.
2. صمّم pipeline: scrape | process | store | query.
3. ابن MVP يمكنك من سؤال «ماذا قيل في الـ 50 paper الأخيرة عن X؟».
4. استخدمه لبحث حقيقي.

#### • نصيحة احترافية

نظام بحث شخصي يفوق كثيراً من «أدوات البحث الجاهزة» لأنه مفصل على اهتماماتك بدقة.

# أتمتة الأعمال — مستوى تنفيذي

Business Automation — قائد التحول الرقمي

# 98

## • الأساس النظري

Power User في الشركات يقود التحول: يحدد فرص أتمتة، يبني proof-of-concepts، يقنع القيادة، يدير الـ rollout.

مهارات مطلوبة: تقنية + استراتيجية + change management + management.  
النتيجة: شركات أكثر كفاءة، موظفون يفعلون عملاً أعمق، نمو مستدام.

## • التطبيق العملي

1. إن لم تكن في موقع قيادة: اقترح pilot صغير لرئيسك.
2. وثق النتائج بدقة (وقت موثر، جودة، رضا).
3. بعد نجاح Pilot، اقترح توسعة.
4. خلال سنة، قد تصير «رئيس أتمتة» شركتك.

### • نصيحة احترافية

أعظم Power Users في 2026 هم «جسور» بين التقنية والأعمال. يفهمون اللغتين، يترجمون بينهما.

## ابن أنظمتك

Build Your Systems — انتقل من مستهلك إلى صانع

## 99

## • الأساس النظري

أعلى مستوى Power User: لا تستخدم أنظمة الآخرين — تبني أنظمة يستخدمها الآخرون. هذا قد يعني: tool، SaaS، مفتوح المصدر، agency، استشارات، تأليف كتب، يوتيوب تعليمي. Claude يقصر طريق البناء. ما كان يحتاج فريقاً، يحتاج الآن شخصاً واحداً مع Claude.

## • التطبيق العملي

1. حدد منتجاً تريد بناءه (حتى صغيراً جداً).
2. اعط نفسك 30 يوماً لـ MVP.
3. استخدم Claude Code، Claude API، Connectors.
4. أطلق ولو لمجموعة من 10 مستخدمين.
5. تعلم من ردودهم، حسن، أعد الإطلاق.

## • نصيحة احترافية

أفضل وقت لبدء «شركة فردية» (solo business) هو الآن. Claude يجعل ذلك ممكناً كما لم يكن قط.

## Master Claude — للأبد

أنت الآن نخبة النخبة

100

## • الأساس النظري

إن قرأت 100 خطوة وطبقت أغلبها، أنت في 0.1% من مستخدمي Claude في العالم. لكن «الإتقان» في عصر AI ليس وجهة، بل ممارسة مستمرة. كل شهر تأتي قدرات جديدة، تحتاج لتفكير جديد. ركائز الـ Master: فضول دائم، تعلم يومي، مشاركة مع المجتمع، استخدام مسؤول، عدم الكبر على المبتدئين (كنت مثلهم).

## • التطبيق العملي

1. اكتب «contract» مع نفسك: ماذا ستفعل أسبوعياً للبقاء على القمة؟
2. اقترح: ساعتان تعلم، تجربة ميزة جديدة، تعليم شخص، التأمل في الاستخدام.
3. علقه في مكان تراه يومياً.
4. بعد 90 يوماً، راجع التزامك. عدل العقد.
5. استمر — هذه ليست نهاية، بل بداية رحلة العمر.

## • نصيحة احترافية

أعظم درس بعد كل هذا: Claude أداة عظيمة، لكن قيمتك أنت — تفكيرك، خبرتك، إنسانيتك. Claude يضخمك، لا يستبدلك. ابق أنت.

# الخاتمة

وصلت إلى نهاية الرحلة — لكنك في الحقيقة وصلت إلى البداية. كل خطوة في هذا الكتاب ليست محطة نهائية، بل بوابة لاكتشاف أعمق.

Claude يتطور باستمرار. المهارة التي بنيتها ليست في حفظ ميزات اليوم، بل في تطوير حدسك حول كيفية التفكير مع نموذج لغوي. هذا الحدس يظل معك حتى لو تغيرت الواجهات والإصدارات.

أكبر فرق بين «من يستخدم Claude» و«من يتقن Claude» ليس عدد الأوامر التي يعرفها، بل قدرته على رؤية فرصة الأتمتة في كل مكان، وتحويل المهمة المتكررة إلى نظام، والفكرة الخام إلى منتج.

ابق فضولياً. جرب. اكسر. ابن. شارك ما تعلمته. والأهم: استمتع بالرحلة.

«من تعلم خطوة، علمها»