Claude Code • Video Workflow

3 أدوات تخلي الـAI يدخل معاك في المونتاج بجد

FFmpeg Skill • Claude Shorts • OpenMontage

شرح، تثبيت، استخدام، وPrompts جاهزة من البداية للنهاية.

إعداد: أحمد العايدي - Ahmed Alaydi

تم التحقق من المصادر الرسمية بتاريخ 12 نوفمبر 2026

AHMED ALAYDI أحمد العايدي

قبل ما تبدأ: كل أداة دورها مختلف

الأدوات الثلاث لا تستبدل بعضها. أفضل نتيجة تحصل عليها عندما تختار الأداة بحسب نوع المهمة، وليس لمجرد أنها "AI Video Tool".

الأداة	أفضل استخدام	الفكرة ببساطة
FFmpeg Skill	المونتاج التقني المحلي	42 أداة منظمة للقص، الكابشن، الصوت، المقاس، التصدير والتحقق.
Claude Shorts	تحويل الفيديو الطويل إلى Shorts	يحلل Transcript، يقترح المقاطع، ثم يحول المقاطع المختارة إلى فيديوهات 9:16 بكابشن.
OpenMontage	Production System كامل	Research + Script + Assets + Editing + Composition، ويقدر يبدأ من Reference Video.

مهم: اعمل دائمًا على نسخة من الفيديو، ولا تسمح لأي Agent بالكتابة فوق الـSource الأصلي. أي API مدفوعة أو توليد Assets مدفوع يجب أن يتم بعد موافقة واضحة منك.

الروابط الرسمية

FFmpeg Skill:
https://github.com/kajisho5/ffmpeg-skill

Claude Shorts:
https://github.com/AgriciDaniel/claude-shorts

OpenMontage Source:
https://github.com/OpenMontageInc/OpenMontage
Download builds: https://github.com/Open-Montage-app/OpenMontage

TOOL 01

FFmpeg Skill

Skill للمونتاج المحلي تعطي Claude Code أو Codex أو أي Agent يقرأ SKILL.md مجموعة أدوات ثابتة بدل ما يخمن أوامر FFmpeg بنفسه.

ما الذي تفعله؟

- Cut / Join / Reframe
- Silence removal
- Captions + Karaoke
- Audio cleanup + Loudness
- Motion graphics / overlays
- Platform export + checks

لماذا مهمة؟

- Tool 42 موثقة
- Local-first
- لا تحتاج Cloud أو API key للأدوات الأساسية
- تبدأ بـ probe وتنتهي بـ verification
- تدعم dry-run قبل الرندر

المتطلبات الأساسية

FFmpeg 6.1+ وPython 3.10+. إذا ستستخدم Installer عبر npx تحتاج Node.js. الـSpeech-to-text للكابشن من الصوت اختياري ويحتاج محرك محلي مثل faster-whisper.

أبسط تثبيت

```
npx ffmpeg-skill
npx ffmpeg-skill doctor
npx ffmpeg-skill contract --json
```

على Windows يمكن تثبيت FFmpeg مثلًا من خلال winget حسب README الرسمية: Gyan.FFmpeg. افحص doctor بعد التثبيت بدل افتراض أن كل filters موجودة.

AHMED ALAYDI

التثبيت FFmpeg Skill - Prompt

انسخ البرومبت كما هو داخل Claude Code. الهدف أن يفحص جهازك أولًا ثم يثبت ويختبر، بدل تنفيذ أوامر عشوائية.

```
راجع المصدر الرسمي أولًا: https://github.com/kajisho5/ffmpeg-skill\n أريد تثبيت FFmpeg Skill على هذا الجهاز واستخدامها مع Claude Code.\n1) افحص نظام التشغيل، Node، Python، FFmpeg وffprobe.\n2) استخدم طريقة التثبيت الرسمية الأنسب لنسختي الحالية.\n3) لا تستبدل أي إعدادات أو Skills موجودة.\n4) بعد التثبيت شغّل npx ffmpeg-skill doctor واقرأ النتيجة.\n5) لو هناك FFmpeg filters أو codecs ناقصة، اذكرها لي قبل أي تغيير إضافي.\n6) اعرض لي مكان تثبيت الـ Skill وكيف أحدثها أو أحذفها.\n7) نفذ اختبارًا صغيرًا على ملف تجريبي فقط، ثم توقف وانتظر موافقتي.
```

الاستخدام FFmpeg Skill - Prompt

```
استخدم FFmpeg Skill لتحليل الفيديو الذي سأحدده. لا تعدل الملف الأصلي.\nابدأ بـ probe ثم اعرض خطة المونتاج قبل التنفيذ. المطلوب عند الموافقة: قص نظيف، إزالة الأجزاء الصامتة فقط إذا طلبت ذلك صراحة، تحسين مستوى الصوت بدون clipping، إضافة Captions واضحة، وتجهيز نسخة 9:16 للسوشيال ونسخة أصلية عند الحاجة.\nاستخدم dry-run أو plan قبل أي Render كبير، ثم تحقق من الناتج بـ probe/check ولقطات preview. لا تحذف أو تعيد ترتيب كلامي إلا إذا طلبت ذلك صراحة. احفظ كل النتائج داخل مجلد outputs مستقل.
```

أفضل Workflow: Probe → Plan/Dry Run → Edit → Check → Preview. لا تبدأ Render طويل قبل التأكد من المقاس والصوت والكابشن على Preview قصير.

AHMED ALAYDI · أحمد العايدي

TOOL 02

Claude Shorts

Skill تفاعلية لـClaude Code هدفها تحويل فيديو طويل إلى Short-form. الأداة لا تختار وتقص بشكل أعمى؛ تمر على Pipeline من 10 خطوات وتطلب موافقتك على المقاطع قبل الرندر.

الـPipeline

Preflight → Transcribe → Detect Content → Analyze → Present Candidates → Approve → Snap Boundaries → Prepare → Render → Export

أهم المميزات

- Transcription باستخدام faster-whisper مع word-level timestamps.
- Claude يقيم 8-12 مقطعًا مرشحًا حسب الـvalue density، emotion، coherence، Hook وpayoff.
- Caption Styles: Bold / Bounce / Clean 3.
- Reframe للفيديو العمودي، مع Face/Cursor tracking حسب نوع المحتوى.
- Remotion للرندر وFFmpeg للتقطيع والتصدير.

المتطلبات

Claude Code، Node.js 18+، Python 3.10+، FFmpeg. الـNVIDIA GPU موصى به وليس شرطًا.

Windows: المشروع نفسه يوصي باستخدام WSL2 لأنه يعتمد على Bash/Unix tools. لا تعتبر PowerShell بديلًا مباشرًا لـsetup.sh بدون تعديل واعٍ.

Claude Shorts - التثبيت والاستخدام

التثبيت الرسمي

```
git clone https://github.com/AgriciDaniel/claude-shorts.git
cd claude-shorts
bash setup.sh
bash install.sh
```

بعدها داخل Claude Code:

```
/shorts ~/Videos/my-video.mp4
```

Prompt التثبيت

```
راجع README الرسمي: https://github.com/AgriciDaniel/claude-shorts\n أريد تثبيت Claude Shorts لاستخراج Shorts من فيديوهات طويلة.\n- افحص FFmpeg و Python 3.10+ و Node.js 18+ و Claude Code.\n- إذا كان النظام Windows استخدم WSL2 كما توصي الوثائق. لا تحاول إجبار setup.sh على PowerShell مباشرة.\n- Clone المشروع، شغّل setup.sh ثم install.sh، وسجل أي Dependency أو خطأ بوضوح.\n- لا تغير ملفات Claude Code الأخرى.\n- بعد التثبيت تحقق أن Skill /shorts متاحة.\n- نفذ Preflight على فيديو تجريبي قصير فقط، ولا تبدأ Transcription طويلة أو Render بدون موافقتي.
```

Prompt الاستخدام

```
استخدم /shorts على الفيديو الذي سأعطيه لك.\nأريدك أن:\n1) تعمل Preflight وتخبرني بالمدة والدقة والـ GPU المتاح.\n2) تعمل Transcription بكلمات وتوقيتات دقيقة.\n3) تقترح أفضل 8-12 مقطعًا مناسبًا لـ TikTok / Reels / Shorts، مع Score وHook وسبب الاختيار.\n4) لا تقص أو ترندر أي مقطع قبل أن أختار أنا المقاطع.\n5) بعد اختياري، اضبط حدود القص على نهاية جملة/سكون طبيعي، واستخدم 9:16 مع Caption Style مناسب.\n6) حافظ على المعنى ولا تنشئ Cold Open أو تعيد ترتيب الكلام من نفسك.\n7) صدّر النتائج في مجلد منفصل واذكر أسماء الملفات النهائية.
```

TOOL 03

OpenMontage

OpenMontage ليس مجرد محرر فيديو. هو Agentic Video Production System يحول الـAI Coding Assistant إلى Production Studio: Research، Script، Assets، Editing، Composition ومراجعة نهائية.

Reference-driven

تقدر تعطيه YouTube / Reel / TikTok / Clip محلي. يحلل transcript، pacing، scenes، keyframes والـstyle، ثم يقترح Concepts جديدة بدل نسخ الفيديو.

Local + Cloud

فيه مسارات بدون API Keys باستخدام FFmpeg وRemotion/Piper ومصادر مفتوحة، مع Providers مدفوعة اختيارية لو احتجتها.

ماذا يحتوي حاليًا؟

المشروع يتطور بسرعة؛ الـREADME الحالية تصفه بـ12 Production Pipelines و100+ Tools و700+ Agent Skill / production-knowledge files. اعتبر الأرقام متغيرة مع الإصدارات.

المتطلبات الحالية

Python 3.10+، FFmpeg، Node.js ضمن الإصدارات التي تحددها الـREADME، وAI coding assistant مثل Claude Code أو Codex أو Cursor.

قاعدة مهمة: لا تحط API Keys أو خدمات مدفوعة من البداية. ابدأ بالمسار Local/Free، وبعدها أضف Provider مدفوع فقط لو عندك سبب واضح وتكلفة معروفة.

AHMED ALAYDI أحمد العايدي

OpenMontage - التثبيت

المصدر الأساسي:

https://github.com/OpenMontageInc/OpenMontage

ولو عايز Builds جاهزة:

https://github.com/Open-Montage-app/OpenMontage

Quick Start من المصدر

```
git clone https://github.com/calesthio/OpenMontage.git
cd OpenMontage
make setup
```

الرابط القديم calesthio يعيد التوجيه حاليًا إلى OpenMontageInc/OpenMontage. على Windows، الـREADME توفر مسار PowerShell بديل عندما لا يكون make متاحًا.

Prompt التثبيت

```
أريد تجهيز OpenMontage محليًا مع Claude Code.
اقرأ README.md وAGENT_GUIDE.md وPROJECT_CONTEXT.md قبل تثبيت أي شيء.
افحص Python 3.10+ وFFmpeg وNode.js حسب المتطلبات الحالية في الـ README.
استخدم Quick Start الرسمي لنظام التشغيل الحالي. على Windows استخدم أوامر PowerShell الموثقة إذا لم يكن make متاحًا.
ابدأ بمسار Zero API Keys / local-free، ولا تطلب أو تحفظ أي API key مدفوع بدون موافقتي.
بعد الإعداد، اكتشف الـ tool registry والـ provider menu الفعلي بدل الافتراض.
لا تبدأ توليد Assets مدفوعة. اعمل فقط فحص بيئة وتجربة قصيرة مجانية/محلية، ثم أعطني تقريرًا بما يعمل وما لا يعمل.
```

OpenMontage - Prompt Reference Video

```
سأعطيك Reference Video أو رابط فيديو يعجبني وموضوعًا جديدًا. استخدم OpenMontage لتحليل المرجع من ناحية: Hook، pacing، structure، scene rhythm، transitions، caption grammar، visual density، audio rhythm، ونمط السرد.\nبعد التحليل: قدم لي 3 Concepts أصلية للموضوع الجديد؛ لا تنسخ هوية صانع المحتوى أو النص أو اللقطات حرفيًا. وضح ما الذي سنحتفظ به كـ principles وما الذي سيتغير ليصبح الفيديو أصليًا.\nاستخدم المسار المجاني/local قدر الإمكان، واعرض أي تكلفة متوقعة قبل استخدام مزود مدفوع.\nاعرض Production Plan وScene Plan وSample/Preview قبل الرندر الكامل، وخذ موافقتي في نقاط القرار الإبداعية. في النهاية نفذ Self-review للصورة والصوت والكابشن قبل تسليم الفيديو.
```

مثال مختصر

```
هذا Reference Video يعجبني من ناحية الـHook والـPacing فقط. حلله، ثم اقترح 3 أفكار أصلية لنفس جمهوري عن موضوع مختلف. لا تنسخ النص أو هوية المبدع. ابدأ بالـProduction Plan وSample قبل أي رندر كامل.
```

النقطة الأقوى هنا: استخدم الـReference كـ"لغة إنتاج" لا كقالب للنسخ. احتفظ بالمبادئ العامة - سرعة القطع، كثافة المعلومات، شكل الهوك - وغيّر المحتوى والزاوية والهوية البصرية.

AHMED ALAYDI • أحمد العايدي

Prompt شامل: ثبّت الثلاثة

ده البرومبت لو عايز تدي Claude Code المهمة من البداية: يفحص جهازك ويجهز الأدوات الثلاث بأقل تغييرات ممكنة.

```
أريدك أن تجهز بيئة Video Production محلية على جهازي باستخدام الأدوات الثلاث التالية فقط، بعد مراجعة الـ README الرسمي لكل مشروع وعدم افتراض أي أوامر غير موثقة:\n1) FFmpeg Skill: https://github.com/kajisho5/ffmpeg-skill\n2) Claude Shorts: https://github.com/AgriciDaniel/claude-shorts\n3) OpenMontage: https://github.com/OpenMontageInc/OpenMontage\n\nالمطلوب:\n- اكتشف نظام التشغيل وإصداره، Architecture، ونسخ Python / Node.js / FFmpeg / Git / Claude Code الموجودة بالفعل.\n- لا تعدل أو تحذف أي إعداد موجود بدون نسخة احتياطية.\n- اشرح لي أي Dependency ناقصة قبل تثبيتها.\n- ثبّت FFmpeg Skill بالطريقة الرسمية الأنسب لبيئتي، ثم شغّل doctor وتأكد من الـ capabilities المتاحة.\n- ثبّت Claude Shorts حسب الـ README الرسمي. إذا كنت على Windows فاستخدم WSL2 لهذا المشروع كما توصي الوثائق، ولا تدّع أن التثبيت اكتمل قبل اختبار shorts/ فعليًا.\n- جهّز OpenMontage من المصدر الرسمي، واقرأ AGENT_GUIDE.md وREADME.md قبل التشغيل. استخدم المسار المجاني/local افتراضيًا ولا تضف أي API مدفوع بدون موافقتي.\n- اختبر كل أداة باختبار صغير وغير مدمر. لا تستخدم ملفاتي الأصلية كمخرجات؛ أنشئ مجلد outputs مستقل.\n- في النهاية أعطني تقريرًا مختصرًا: ما تم تثبيته، الإصدارات، ما يعمل، ما يحتاج إجراء يدوي، وكيف أبدأ وأوقف وأحدث كل أداة.\n- لا تبدأ مونتاج أي فيديو حقيقي قبل أن أوافق على خطة التشغيل.
```

أحمد العايدي · AHMED ALAYDI

أفضل Workflow يجمع الثلاثة

1 **Claude Shorts**

لو عندك فيديو طويل: اكتشف أفضل المقاطع أولًا. لا تعمل Render نهائي قبل اختيارك.

2 **FFmpeg Skill**

نظف المقاطع المختارة: قص دقيق، كابشن، صوت، Reframe، Export، ثم Verification.

3 **OpenMontage**

لو عايز إنتاج جديد، Reference-based أو Pipeline أكبر: خليه مشروع منفصل له Research وScript وAssets وReview.

Prompt جاهز للـWorkflow

```
عندي فيديو طويل وأريد أفضل Workflow باستخدام الأدوات الثلاث المثبتة.\n1) استخدم Claude Shorts أولًا لاكتشاف المقاطع المرشحة فقط، واعرضها عليّ بدون Render نهائي.\n2) بعد أن أختار المقاطع، استخدم FFmpeg Skill للتنظيف التقني: حدود قص دقيقة، captions/export ،reframe ،loudness والتحقق من الناتج.\n3) إذا احتجت نسخة جديدة مبنية على Reference أو إنتاج أكثر تعقيدًا، استخدم OpenMontage كـ production pipeline مستقل ولا تخلط ملفاته مع الماستر الأصلي.\n4) لا تعدّل الـ source ولا تحذف الكلام ولا تعيد ترتيب المحتوى بدون موافقة.\n5) افصل outputs لكل Tool داخل مجلدات واضحة، واكتب تقريرًا نهائيًا بالملفات الناتجة والإعدادات المستخدمة.
```

أحمد العايدي · AHMED ALAYDI

اختار الأداة الصح بسرعة

الموقف	ابدأ بـ
عايز تشيل سكوت / تزبط LUFS / تضيف Captions / Reframe	FFmpeg Skill
عندك Podcast أو فيديو طويل وعايز تختار منه أفضل Shorts	Claude Shorts
عندك Reference Video وعايز فيديو جديد أصلي بنفس طاقة الإنتاج	OpenMontage
عايز فيديو من فكرة فقط: Research → Script → Assets → Render	OpenMontage
عايز خطوات تقنية predictable ومحلية على ملف موجود	FFmpeg Skill
عايز Candidate scoring وموافقة بشرية قبل القص	Claude Shorts

3 أخطاء تتجنبها

- **ما تقولش للـAgent "منتج الفيديو" وخلاص:** حدد الهدف، المنصة، المقاس، هل يسمح بالقص أم لا.
- **ما تشتغلش على الـSource:** خليه Read-only، وكل Output في مجلد جديد.
- **ما تعتمدش على "Done":** لازم probe/check + Preview + مراجعة صوت وصورة.

AHMED ALAYDI • أحمد العايدي

Troubleshooting سريع

FFmpeg Skill

- شغّل npx ffmpeg-skill doctor.
- لو Caption/Filters ناقصة، المشكلة غالبًا في Build الـFFmpeg وليس الـSkill نفسها.
- الـTranscription اختيارية وتحتاج Speech Engine محلي لو هتولد captions من الصوت.

Claude Shorts

- على Windows استخدم WSL2.
- لو الرندر بطيء، الـGPU اختياري لكن يفرق في transcription/export.
- تأكد من FFmpeg وNode وPython قبل تشغيل setup.

OpenMontage

- راجع الـREADME مع كل تحديث لأن المشروع يتغير بسرعة.
- ابدأ بدون API Keys؛ بعض الـProviders اختيارية ومدفوعة.
- لو عندك Windows بدون make، استخدم مسار PowerShell الموثق في README الحالية.

أمان: لا تضع Tokens أو API Keys داخل Prompt أو Screenshot. استخدم .env محلي ولا ترفعه إلى GitHub.

AHMED ALAYDI أحمد العايدي

Checklist قبل أول فيديو حقيقي

- ✓ كل أداة تم تثبيتها من المصدر الرسمي.
- ✓ FFmpeg وPython وNode نسخهم مناسبة.
- ✓ جرّبت Clip صغير قبل ملف طويل.
- ✓ Source محفوظ ولم يتم Overwrite.
- ✓ اتفقت مع الـAgent: هل مسموح حذف السكوت أو إعادة ترتيب المحتوى؟
- ✓ اخترت Target: TikTok / Reels / Shorts / YouTube.
- ✓ Preview اتراجع بصريًا.
- ✓ الصوت اتقاس ولم يحصل clipping.
- ✓ أي تكلفة Cloud معروفة وموافق عليها قبل التنفيذ.

القاعدة الذهبية: الـAI يسرع المونتاج، لكن أنت تفضل صاحب القرار الإبداعي. اطلب Plan، راجع Preview، وبعدها Render.

المصادر

FFmpeg Skill: https://github.com/kajisho5/ffmpeg-skill
Claude Shorts: https://github.com/AgriciDaniel/claude-shorts
OpenMontage: https://github.com/OpenMontageInc/OpenMontage
OpenMontage Downloads: https://github.com/Open-Montage-app/OpenMontage

AHMED ALAYDI • أحمد العايدي

SAVE THIS GUIDE

ابدأ بأداة واحدة
واستخدمها صح

مش لازم تستخدم الثلاثة في كل فيديو. اختار الأداة حسب المشكلة، وخلي كل مرحلة قابلة للمراجعة قبل التنفيذ.

Ahmed Alaydi • أحمد العايدي

Video Editing with AI Coding Agents - دليل تعليمي

AHMED ALAYDI أحمد العايدي