

รู้จัก Claude ก่อนเริ่มใช้งานจริง

ทุกอย่างที่คุณควรรู้ก่อนใช้ Claude
เพื่อให้เลือกใช้งานได้เหมาะกับตัวเอง

หัวข้อใน EP-1 นี้

- 1 Claude คืออะไร? ใครสร้าง? ต่างจาก ChatGPT ยังไง?
- 2 โมเดล Sonnet, Opus, Haiku ต่างกันยังไง?
- 3 Claude Chat, Cowork, Code ใช้งานยังไง?
- 4 แอป Claude Desktop ทำอะไรได้บ้าง?
- 5 Claude Code: Desktop App vs VS Code vs Terminal

Claude คืออะไร? เรื่องราวต้นกำเนิด

ใครเป็นผู้สร้าง · ทำไมมันถึงมีอยู่ · ต่างจาก ChatGPT ยังไง · "Constitutional AI" คืออะไร

Claude คือใคร?

Claude คือ AI ผู้ช่วย (AI assistant) ที่สร้างโดยบริษัทชื่อ **Anthropic** เป็นบริษัทอเมริกันก่อตั้งในปี 2021 โดยพี่น้อง Dario และ Daniela Amodei พร้อมทีมงาน 11 คนที่ลาออกจากบริษัท OpenAI (บริษัทที่สร้าง ChatGPT)

📖 **ศัพท์: AI assistant (ผู้ช่วย AI)**

โปรแกรมที่เราพิมพ์คุยด้วยได้เหมือนคุยกับคน มันเข้าใจคำถามและตอบกลับเป็นข้อความ ช่วยคิด ช่วยเขียน ช่วยทำงานได้

ทำไมถึงลาออกมาตั้ง Anthropic?

Dario เคยเป็นรองประธานฝ่ายวิจัย (VP Research) ของ OpenAI Daniela เป็นรองประธานฝ่ายความปลอดภัย และนโยบาย (VP Safety & Policy) ที่ OpenAI พวกเขาเห็นว่า AI กำลังพัฒนาเร็วมาก แต่เรื่องความปลอดภัย และจริยธรรมยังตามไม่ทัน เลยตัดสินใจลาออกมาตั้งบริษัทใหม่ที่เน้น "AI ที่ปลอดภัยและซื่อสัตย์" เป็นหลัก

ทำไม Claude ต่างจาก ChatGPT?

ทั้ง ChatGPT (จาก OpenAI) และ Claude (จาก Anthropic) เป็น AI ที่คุยกับเราได้เหมือนกัน แต่วิธีฝึก AI นั้น ต่างกันมาก

ChatGPT ใช้วิธี RLHF

RLHF ย่อมาจาก Reinforcement Learning from Human Feedback ภาษาง่ายๆ คือ "สอน AI ด้วยการให้คนคลิก like/dislike" – ให้คนหลายๆ คนมาดูคำตอบของ AI แล้วโหวตว่าชอบหรือไม่ชอบ AI จะเรียนรู้ว่าควรตอบแบบไหน

Claude ใช้วิธี Constitutional AI

Constitutional AI แปลว่า "AI ที่มีรัฐธรรมนูญ" คือ Anthropic เขียนหลักการขึ้นมาเป็นกฎ (เหมือนรัฐธรรมนูญ) แล้วให้ AI เรียนรู้ตามหลักการนั้น เช่น "ต้องไม่หลอกคน" "ต้องช่วยเหลือคน" "ต้องไม่ทำร้ายคน" AI จะตัดสินใจตามหลักการ ไม่ใช่ตามการ vote ของคน

ผลที่ต่างกัน: Claude มีแนวโน้มเชื่อสัตย์มากกว่า ยอมรับเมื่อไม่รู้คำตอบ ไม่แต่งเรื่อง และคิดยาวกว่าก่อนตอบ เหมาะกับงานที่ต้องการความถูกต้องและความรอบคอบ

Anthropic เติบโตเร็วแค่ไหน?

จาก 0 บาทในปี 2021 มูลค่าบริษัทตอนนี้คือ **380 พันล้านเหรียญสหรัฐ** (ประมาณ 13 ล้านล้านบาท) ใหญ่กว่า PTT ของไทยหลายเท่า รายได้เพิ่มจาก 1 พันล้านเหรียญเป็น 30 พันล้านเหรียญในเวลาเพียง 15 เดือน

บริษัทใหญ่ในอเมริกา 8 ใน 10 อันดับแรก (Fortune 10) เป็นลูกค้าของ Claude และ **4% ของโค้ดทั้งหมดบน GitHub** ตอนนี้เขียนโดย Claude Code

สรุปบทที่ 1

Claude = AI ที่สร้างโดย Anthropic บริษัทที่ก่อตั้งโดยอดีตทีม OpenAI ที่ลาออกมาเพื่อสร้าง AI ที่ปลอดภัยกว่า

ต่างจาก ChatGPT = ใช้วิธี Constitutional AI สอนด้วยหลักการ ทำให้เชื่อสัตย์กว่า รอบคอบกว่า

โมเดล Claude: Sonnet · Opus · Haiku

"สมอง" ภายในผลิตภัณฑ์ Claude ทุกตัว · ทำให้ไม่มีหลายเวอร์ชัน · ใช้เวอร์ชันไหนดี

โมเดล (Model) คืออะไร?

📖 ศัพท์: Model (โมเดล)

"สมอง" ของ AI ที่ทำงานเบื้องหลัง โมเดลตัวเดียวกัน = สมองเดียวกัน Claude มี 3 โมเดล (3 ขนาดสมอง) ให้เลือกตามงานที่ใช้

Claude มี 3 โมเดล ตั้งชื่อตามรูปแบบบทกวี เพราะแต่ละแบบมีลักษณะต่างกัน:

Haiku 4.5

เร็วที่สุด · ราคาถูกที่สุด

\$1/M tokens

งานเร็วๆ ง่ายๆ
สรุปข้อความ
ถาม-ตอบทั่วไป

ความจำ: 200K

(หนังสือ ~150 หน้า)

เหมือนคนช่วยตอบทั่วไป

Sonnet 4.6

สมดุล · ใช้บ่อยที่สุด

\$3/M tokens

เขียนโค้ดทั่วไป
เขียนบทความ
วิเคราะห์งานทั่วไป

ความจำ: 1M tokens

(หนังสือ ~750 หน้า)

เหมือนพนักงานเก่ง

Opus 4.7

ฉลาดสุด · แพงสุด

\$5/M tokens

งานยากซับซ้อน
วางแผนระยะยาว
วิจัยลึกๆ

ความจำ: 1M tokens

(หนังสือ ~750 หน้า)

เหมือนผู้เชี่ยวชาญ

📖 ศัพท์: Token (โทเค็น)

ชิ้นเล็กๆ ของคำที่ AI ใช้นับ "Hello" = ประมาณ 1 token | ข้อความยาวๆ = หลายร้อย token | ราคา \$1/M tokens = 1 ดอลลาร์ต่อ 1 ล้าน tokens

📖 ศัพท์: Context window (ความจำ)

จำนวน tokens สูงสุดที่ AI จำได้ในการสนทนาเดี่ยว Sonnet/Opus จำได้ 1M tokens = อ่านหนังสือทั้งเล่ม 750 หน้าได้ในครั้งเดียว

ทำไมต้องมีหลายโมเดล?

เพราะแต่ละงานต้องการ "สมอง" ต่างขนาดกัน เปรียบเทียบเหมือนรถ:

- **Haiku** = มอเตอร์ไซค์ → เร็ว ประหยัด ไปได้ไกลๆ เหมาะกับงานง่าย
- **Sonnet** = รถเก๋ง → สมดุล ใช้งานได้ทุกอย่าง ใช้บ่อยที่สุด
- **Opus** = รถสปอร์ตหรู → แรง แพง สำหรับงานยากซับซ้อน

คำแนะนำ: เริ่มต้นใช้ **Sonnet 4.6** เป็นค่าเริ่มต้น แล้วสลับเป็น **Opus 4.7** เมื่อมีงานยากมาก หรือใช้ **Haiku 4.5** เมื่อต้องการเร็วและประหยัด คุณกำลังคุยกับ **Sonnet 4.6** ตอนนี้

สรุปบทที่ 2

Haiku 4.5 = เร็ว/ถูก สำหรับงานง่ายๆ

Sonnet 4.6 = สมดุล ใช้บ่อยที่สุด (default)

Opus 4.7 = ฉลาดสุด/แพงสุด สำหรับงานยาก

Claude Chat · Cowork · Code (Claude Code)

3 ผลิตภัณฑ์หลัก · ต่างกันยังไง · "Code" = "Claude Code" เป็นสิ่งเดียวกัน

3 ผลิตภัณฑ์หลักของ Claude

⚠️ ข้อสำคัญ — อ่านก่อน:

"Code" และ "Claude Code" คือสิ่งเดียวกัน — เราเรียกสั้นๆ ว่า "Code" แต่ชื่อเต็มคือ "Claude Code"

- Claude = AI ทั้งหมดที่ Anthropic สร้าง
- Claude Code = พีเจอร์หนึ่งของ Claude สำหรับเขียนโค้ด

ในเอกสารนี้ เราจะเขียน "Code/Claude Code" เพื่อให้ชัดเจนว่าเป็นสิ่งเดียวกัน



Chat

คุยกับ Claude
ในเบราว์เซอร์

ไม่เข้าถึงไฟล์
พิมพ์ทุกอย่างเอง

ฟรี ✓



Cowork

ทำงานบนคอมพิวเตอร์จริง
(ในโฟลเดอร์ที่อนุญาต)

อ่าน/แก้ไฟล์ได้
เชื่อม Gmail/Slack

เสียเงิน (Pro+) 💳



Code / Claude Code

เขียนโค้ดจริงจัง
เข้าถึงทั้งเครื่อง

รันคำสั่งเทอร์มินัล
ติดตั้งโปรแกรม

เสียเงิน (Pro+) 💳

ความต่างที่สำคัญที่สุด: ระดับการเข้าถึง (Permission)

เปรียบเทียบกับ การให้คนเข้าบ้านคุณ:

💬 Chat = คุยกันที่หน้าประตู

เพื่อนมาคุยที่หน้าประตู คุยเสร็จกลับ ไม่ได้เข้าบ้าน ไม่เห็นของในบ้าน ไม่แตะอะไร = Chat คุยกับ Claude ในเบราว์เซอร์ ไม่มีการเข้าถึงไฟล์ในเครื่องคุณ

💻 Cowork = ให้เข้าบางห้อง

เพื่อนเข้าได้แค่ห้องนั่งเล่นและห้องครัว ห้องนอนเข้าไม่ได้ = Cowork เข้าถึงเฉพาะ "โฟลเดอร์ที่คุณอนุญาต" เท่านั้น อ่าน/แก้ไฟล์ในโฟลเดอร์นั้นได้ ทำงานในระบบความปลอดภัย (sandbox) ที่กั้นจากส่วนอื่นของเครื่อง

📖 ศัพท์: Sandbox (แซนด์บ็อกซ์)

"กล่องทราย" = ห้องปลอดภัยที่ AI ทำงานได้ในนั้นเท่านั้น ออกไปทำลายส่วนอื่นของเครื่องไม่ได้ เหมือนเด็กเล่นในกระบะทราย ทรายไม่กระจายออกข้างนอก

🔑 Code = ให้กุญแจบ้านทั้งหลัง

เพื่อนเข้าได้ทุกห้อง เปิดอุปกรณ์ได้ ทำได้ทุกอย่าง = Code เข้าถึงทั้งเครื่อง รันคำสั่งเทอร์มินัล ติดตั้ง/ลบโปรแกรม แก้ไฟล์ระบบได้ ไม่มี sandbox เพราะนักพัฒนาต้องการอิสระเต็มที่

ใช้ตัวไหนเหมาะกับอะไร?

ผลิตภัณฑ์	เหมาะกับใคร	ตัวอย่างงาน
Chat	ทุกคน · ผู้เริ่มต้น	ถามคำถาม คิดไอเดีย เรียนรู้ เขียนข้อความ
Cowork	คนทำงานเอกสาร · คนไม่ใช่ นักพัฒนา	จัดการไฟล์ ทำรายงาน เชื่อมต่อ Gmail สร้างงาน
Code	นักพัฒนา (Developer)	เขียนเว็บ แก้บั๊ก สร้างแอป งานเขียนโปรแกรม

สรุปบทที่ 3

Chat = เบราร์เซอร์ คุยเฉยๆ ไม่แตะเครื่อง – สำหรับทุกคน ฟรี

Cowork = ทำงานในโฟลเดอร์ที่อนุญาต ใน **sandbox** – สำหรับงานเอกสาร เสียเงิน

Code = เข้าถึงทั้งเครื่อง รันคำสั่งได้ทุกอย่าง – สำหรับนักพัฒนา เสียเงิน

แอป Claude Desktop

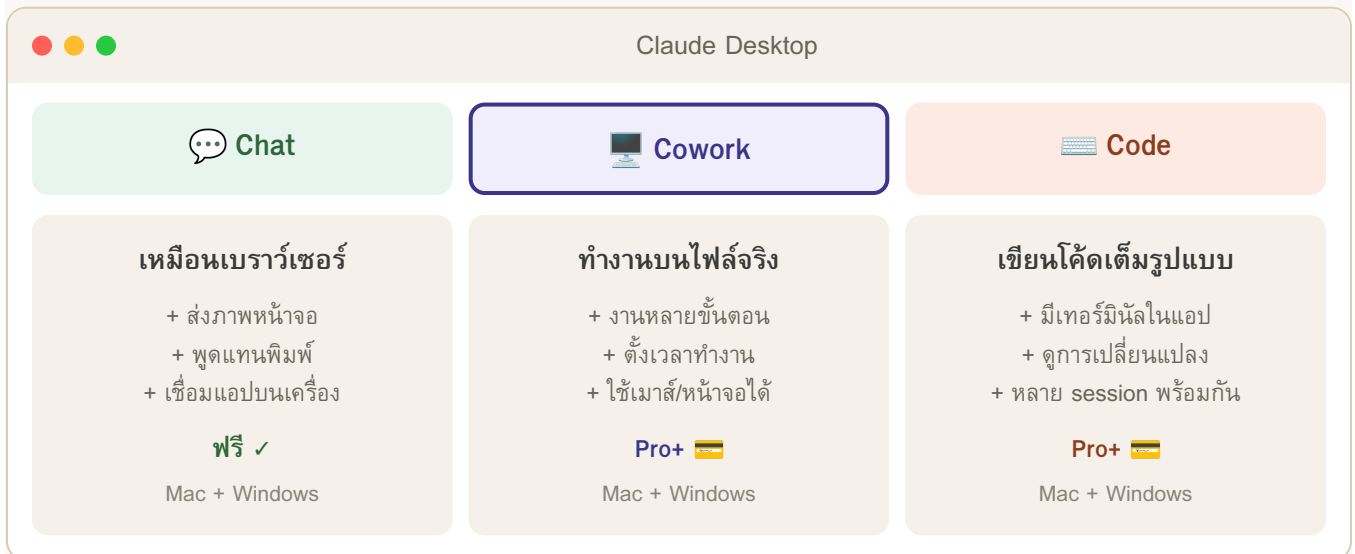
แอปดาวนโหลดติดเครื่อง · 3 แท็บ (Chat, Cowork, Code/Claude Code) · ต่างจากเบราว์เซอร์ยังไง

ทำไมต้องมีแอป Desktop?

ในเบราว์เซอร์ (claude.ai) คุณคุยกับ Claude ได้แล้ว แล้วทำไมต้องดาวนโหลดแอปแยก?

คำตอบง่ายๆ: **เบราว์เซอร์ติดอยู่ใน sandbox** เข้าถึงไฟล์ในเครื่องคุณไม่ได้ ปิดเบราว์เซอร์ = ความจำหายหมด
แอป Desktop ทำงานเป็นโปรแกรมจริงในเครื่องคุณ ทำงานเบื้องหลังได้ เข้าถึงไฟล์ได้ (ถ้าอนุญาต)

เบราว์เซอร์ = เหมือนคุยโทรศัพท์กับเพื่อน · แอป Desktop = เหมือนมีเพื่อนนั่งทำงานในห้องคุณ



แท็บ Cowork: ฟีเจอร์ที่คนส่วนใหญ่ยังไม่รู้

Cowork เปิดตัวต้นปี 2026 มีเฉพาะในแอป Desktop เท่านั้น ไม่มีในเบราว์เซอร์ ต้อง Pro ขึ้นไปถึงใช้ได้ มี 5 ฟีเจอร์เด่นที่คนไม่ค่อยรู้:

1. ทำงานกับไฟล์จริงในเครื่อง

ให้สิทธิ์ Cowork เข้าถึงโฟลเดอร์เฉพาะ มันจัดเรียงไฟล์เป็นร้อย แปลงใบเสร็จเป็นรายงานค่าใช้จ่าย เปลี่ยนชื่อไฟล์ตามแพทเทิร์น รวมข้อมูลจากเว็บ + บันทึก + บทความเป็นรายงาน

2. Scheduled Tasks (ตั้งเวลาทำงานอัตโนมัติ)

ตั้งให้ Claude ดึงตัวเลข เขียนรายงานทุกเช้าจันทร์ 9 โมง ตั้งครั้งเดียวทำเองทุกสัปดาห์

3. ส่งงานจากมือถือ

มอบหมายงานจากมือถือระหว่างเดินทาง Claude หยิบงานไปทำต่อบน Mac ที่ออฟฟิศ พอกลับถึงโต๊ะ งานเสร็จแล้ว

4. Computer Use (ควบคุมเมาส์ + หน้าจอ)

ฟีเจอร์ใหม่ล่าสุด (มีนาคม 2026) Claude ใช้เมาส์คลิก พิมพ์ เปิดแอปได้เอง เช่น "เปิดเว็บบริษัท ดาวน์โหลดรายงานยอดขาย ใส่ Excel แล้วส่งให้หัวหน้า" – Claude จะใช้เมาส์ของคุณทำให้จริงๆ

5. Dispatch (Beta)

แชทถาวรใน Cowork ส่งข้อความบอกงาน Dispatch ตัดสินใจเองว่าจะส่งให้ Cowork ทำ หรือเปิด Code session มาทำ

เปรียบเทียบเบราว์เซอร์ vs แอป Desktop

ฟีเจอร์	เบราว์เซอร์ (claude.ai)	แอป Desktop
Chat (คุย)	✓ ได้	✓ ได้ + ส่งภาพหน้าจอ + พุด
Cowork (ทำงานบนไฟล์)	✗ ไม่ได้	✓ ได้ (Pro+)
Code (เขียนโค้ด)	✗ ไม่ได้	✓ ได้ (Pro+)
ตั้งเวลาทำงาน	✗ ไม่ได้	✓ ได้
ควบคุมเมาส์/หน้าจอ	✗ ไม่ได้	✓ ได้
ค่าใช้จ่าย	ฟรี + Pro	ดาวน์โหลดฟรี · Cowork/Code ต้อง Pro

ข้อสำคัญ: แอป Desktop ต้องเปิดค้างไว้ตอน Cowork ทำงาน ถ้าปิดแอป งานจะหยุด ใช้คอมต้องเปิดอยู่ แต่เดินไปทำอย่างอื่นได้

ดาวน์โหลดที่: claude.com/download · ใช้ได้บน Mac + Windows · ฟรีดาวน์โหลด

Code / Claude Code: ใช้งานได้ 3 ที่

Desktop App vs VS Code vs Terminal · "Code" = "Claude Code" เป็นสิ่งเดียวกัน · เพจนี้จะสอนผ่าน VS Code

สิ่งสำคัญที่สุดที่ต้องเข้าใจก่อน

🚩 สำหรับสมาชิกเพจ:

เพจนี้จะสอน Claude Code ผ่าน VS Code Extension เป็นหลัก เพราะเป็นวิธีที่เหมาะสมกับคนส่วนใหญ่ที่สุด – ใช้งานง่าย เห็นโค้ดชัดเจน และ Anthropic แนะนำเป็นทางการว่าเป็น "the recommended way"

ทั้ง 3 ที่ใช้ Claude Code ตัวเดียวกัน · สมองเดียวกัน · พลังเท่ากัน · โมเดลเดียวกัน

ต่างกันเฉพาะ "คุณจะทำหน้าที่ไหน" ซึ่งทำให้คุณเห็นอะไร และได้ฟีเจอร์เสริมต่างกัน

เปรียบเทียบเป็นภาพ

🖥️ Desktop App

แอปติดเครื่อง · แท็บ Code

เหมือนหัวหน้า
คุณ 3 คนทำงาน

- ✓ ดูการเปลี่ยนแปลง
- ✓ หลาย session
- ✓ มีสติการใช้งาน
- ✓ เทอร์มินัลในแอป

- ✗ ไม่มี code editor
- ✗ ไม่มี Linux
- ✗ ใช้ token เร็วกว่า

คุณ "ดู" Claude Code ทำงาน

💠 VS Code Extension

ปลั๊กอินใน VS Code

เหมือนทำกับข้าวกัน
กับผู้ช่วยฉลาด

- ✓ ดูไฟล์ทั้งหมดได้
- ✓ diff ในไฟล์เลย
- ✓ @-mention ไฟล์
- ✓ Anthropic แนะนำ

- ✗ กิน RAM มาก
- ✗ rewind ไม่เต็มที่
- ✗ SSH ใช้ไม่ค่อยดี

คุณ "เขียน" โค้ดกับ Claude Code

📄 Terminal

คำสั่งล้วน (CLI)

เหมือนส่ง SMS
กับหุ่นยนต์

- ✓ เบาที่สุด เร็วที่สุด
- ✓ ทำงานบนเซิร์ฟเวอร์
- ✓ rewind ย้อนกลับได้
- ✓ ฟีเจอร์ใหม่ก่อนเสมอ

- ✗ ยากสำหรับมือใหม่
- ✗ ไม่มีรูป ไม่มีไฟล์
- ✗ ต้องเรียน tmux

คุณ "พิมพ์" คำสั่งให้ Claude Code

1. Desktop App (แท็บ Code)

เปิดแอป Claude Desktop คลิกแท็บ Code คุณจะเห็นแถบด้านซ้ายมีรายการ session ทั้งหมด ตรงกลางมีแชทกับ Claude มี file editor มีเทอร์มินัลในตัว มี preview area ลากปรับขนาดได้เหมือน IDE คุณสามารถเปิดหลาย session พร้อมกันได้ (โปรเจกต์ละ session) สลับไปมาในแถบซ้าย

📖 ศัพท์: Session (เซสชัน)

"การสนทนาหนึ่งครั้ง" ที่ Claude ทำงานหนึ่งอย่าง เช่น session 1 = แก่บ๊วกเว็บร้านอาหาร · session 2 = ทำหน้า login ของแอป สามารถเปิดหลาย session พร้อมกันได้

✓ ข้อดี:

- รัน AI หลายตัวพร้อมกัน – backend + frontend + testing ในหน้าต่างเดียว
- มี terminal ในแอป ไม่ต้องเปิดแยก
- มี file editor ในแอป แก้ไฟล์ตรงๆ ได้ ไม่ต้องเปิด Finder
- ลากจัด panel ได้ตามใจ
- เชื่อม SSH ไป remote server ได้
- preview HTML/PDF live ได้
- มี usage dashboard ดูสถิติ tokens, sessions
- Cloud Routines ทำงานต่อแม้ปิดเครื่อง

✗ ข้อเสีย:

- ไม่มี code editor เต็มรูปแบบสำหรับเขียนโค้ดเอง – มีแค่ basic editor
- ใช้ทำ CI/CD pipeline ไม่ได้ (อัตโนมัติบนเซิร์ฟเวอร์)
- ผูกกับ Claude อย่างเดียว ใช้ AI ตัวอื่นไม่ได้
- ยังไม่มีบน Linux
- ใช้ token เร็วกว่าเมื่อเปิดหลาย session

2. VS Code Extension (ตัวที่คุณใช้)

VS Code Extension คือปลั๊กอินที่ติดตั้งใน VS Code คุณอยู่ใน VS Code ตามปกติ Claude อยู่ใน sidebar ด้านขวา/ซ้าย เมื่อ Claude แก่โค้ด การเปลี่ยนแปลงปรากฏ **ในไฟล์ที่คุณกำลังดู** – สีเขียวบรรทัดที่เพิ่ม สีแดงบรรทัดที่ลบ

📖 ศัพท์: Extension (เอ็กซ์เทนชัน/ปลั๊กอิน)

"ส่วนเสริม" ที่ติดตั้งเพิ่มในโปรแกรมเพื่อเพิ่มความสามารถ คล้ายแอปเสริมในมือถือ Claude Code Extension = ส่วนเสริมที่ทำให้ VS Code คุยกับ Claude ได้

📖 ศัพท์: @-mention

พิมพ์ @ ตามด้วยชื่อไฟล์ เช่น @login.js Claude จะอ่านไฟล์นั้นทันที ไม่ต้องอธิบายว่าไฟล์อยู่ที่ไหน

📖 ศัพท์: Diff (ดิฟฟ์)

"ส่วนต่าง" – แสดงว่าโค้ดเปลี่ยนตรงไหน สีเขียว = บรรทัดใหม่ที่เพิ่ม สีแดง = บรรทัดที่ลบ ทำให้รู้ทุกการเปลี่ยนแปลงก่อนยอมรับ

✓ ข้อดี:

- เหมาะที่สุดสำหรับ **เขียนโค้ดเองทุกวัน**
- เห็น diff ในไฟล์เลย ไม่ต้องไปที่อื่น
- @-mention ไฟล์ได้ทันที + เลือกบรรทัด
- เปิดหลายแชทเป็นแท็บแยก
- คู่เคย – file tree, git, terminal ใน VS Code ที่คุณรู้จัก
- 2 ล้านคนใช้ มีผู้ใช้เยอะ stable
- Anthropic แนะนำเป็นทางการ "the recommended way"
- มี terminal ใน VS Code ใช้พีเจเออร์ terminal ได้ในนี้

✗ ข้อเสีย:

- ช้าตอนเพิ่ม context ในโปรเจกต์ใหญ่ที่มีไฟล์เป็นพัน
- multi-root workspace ทำให้ skills โหลดช้า เปลือง tokens
- VS Code ใช้ RAM เยอะ คอมเก่าอาจช้า
- rewind ไม่เต็มเท่า terminal
- ผ่าน SSH remote ใช้ได้ไม่ดี
- รัน script อัตโนมัติไม่ได้ตรงๆ ต้องใช้ terminal

3. Terminal (CLI)

Terminal คือ Claude Code ตัวต้นฉบับ เปิด Terminal ของ Mac (หรือ Linux) พิมพ์ `cd` เข้าโฟลเดอร์โปรเจกต์ พิมพ์ `claude Claude Code` เริ่มทำงาน ทุกอย่างเป็นข้อความ — ไม่มี GUI ไม่มีปุ่ม ไม่มี file tree พิมพ์คำสั่ง Claude อ่านไฟล์ แก้ไฟล์ แสดงผลเป็นข้อความสี่ๆ

📖 ศัพท์: CLI = Command Line Interface

หน้าจอที่พิมพ์คำสั่งเป็นข้อความล้วน ไม่มีปุ่มให้คลิก ไม่มีรูป ตรงข้ามกับ GUI (Graphical User Interface) ที่มีรูปและปุ่ม Terminal = ที่ใช้ CLI

📖 ศัพท์: tmux

โปรแกรมที่แบ่ง terminal หนึ่งหน้าจอเป็นหลายช่อง ทำให้รัน Claude หลายตัวพร้อมกันใน terminal เดียวได้

📖 ศัพท์: Remote server (เซิร์ฟเวอร์ทางไกล)

คอมพิวเตอร์ที่ **ไม่ใช่**เครื่องของคุณ อยู่ที่ไหนสักแห่ง — อาคารของ Amazon, Google โค้ดของคุณรันที่นั่นเพื่อให้คนในอินเทอร์เน็ตเข้าใช้ได้

📖 ศัพท์: SSH

วิธีเชื่อมต่อจาก Mac ของคุณไปยัง remote server เหมือนทำอุโมงค์ พอ SSH เข้าไปแล้ว พิมพ์คำสั่งบน Mac คำสั่งจะรันที่เซิร์ฟเวอร์

📖 ศัพท์: CI/CD Pipeline

ระบบหุ่นยนต์ที่ทำงานเองเมื่อคุณ save โค้ดขึ้น GitHub ตรวจสอบโค้ด ทดสอบ และเอาขึ้นเว็บอัตโนมัติ บริษัทใหญ่ใช้ ตั้งครั้งเดียวรันตลอด

✓ ข้อดี:

- เบาที่สุด — กิน RAM น้อย CPU น้อย
- เร็วที่สุด — แค่พิมพ์ `claude`
- ทำงานบน remote server ผ่าน SSH ได้ดีที่สุด
- รันเป็น script อัตโนมัติได้ — เรียก `claude` จาก script ได้
- Double-ESC rewind — ย้อนกลับและ undo โค้ดได้
- ผสม AI ตัวอื่นได้ — สลับ Claude, GPT ฯลฯ
- ได้พีเจอรี่ใหม่ก่อนเสมอ — Anthropic ปล่อยที่ terminal ก่อน
- CLI flags ปรับละเอียดได้ทุกอย่าง

X ข้อเสีย:

- เรียนยากสุด – ต้องรู้คำสั่ง terminal เช่น cd, ls, git
- ไม่เห็น file tree ต้องพิมพ์คำสั่งดู
- diff เป็นข้อความสีไม่ใช่ side-by-side
- ต้องเรียน tmux ถ้าจะรันหลาย agent
- มือใหม่กลัวได้
- ไม่มีหน้าจอสถิติแบบ visual

เปรียบเทียบรวม

เรื่อง	Desktop App	VS Code	Terminal
หลาย session	✓ มี sidebar	✓ แท็บแยก	✓ tmux
เห็น file tree	✓ มี	✓ ครบที่สุด	✗ ไม่มี
Diff ในไฟล์	มี diff viewer	✓ ในไฟล์เลย	ข้อความเท่านั้น
Rewind ประวัติ	checkpoint	/rewind	✓ Double-ESC
Remote server (SSH)	✓ ได้	มีปัญหา	✓ ดีที่สุด
กิน RAM	กลาง	มากที่สุด	✓ น้อยสุด
พีเจอรี่ใหม่	ตามหลัง	ตามหลัง	✓ ได้ก่อน
เรียนยาก	ง่าย	กลาง	ยาก

ใช้ตัวไหนเหมาะกับใคร?

นักพัฒนาคนเดียว / Indie Hacker

ใช้: VS Code Extension เพียงพอ ไม่ต้องใช้ multi-session
เริ่มใช้ Desktop App เมื่อ: มี 5+ โปรเจกต์พร้อมกัน

Tech Lead ดูแลหลาย repo

ใช้: Desktop App สำหรับ orchestrate หลาย agent + VS Code สำหรับเขียนโค้ดเอง

ทีม Platform / DevOps

ใช้: Desktop App + Cloud Routines (อัตโนมัติ) + Terminal สำหรับ CI

Power User

ใช้: Terminal เป็นหลัก + Routines สำหรับงานตามเวลา

มือใหม่

ใช้: เริ่มที่ VS Code Extension – เรียนรู้ง่ายที่สุด มีคนช่วยเหลือเยอะที่สุด

สรุปบทที่ 5

Desktop App = คุณ "ดู" Claude ทำงาน → เหมาะกับคนที่ direct AI มากกว่าเขียนเอง

VS Code = คุณ "เขียน" โค้ดด้วยกัน → เหมาะกับนักพัฒนาที่เขียนโค้ดทุกวัน

Terminal = คุณ "พิมพ์" คำสั่งล้วน → เหมาะกับ power user, remote server, อัตโนมัติ

สรุป EP-1

Claude มี 3 ผลิตภัณฑ์หลัก (Chat, Cowork, Code/Claude Code) · มี 3 โมเดล (Haiku, Sonnet, Opus) · มีแอป Desktop กับ 3 แท็บ · Code/Claude Code ใช้ได้ 3 ที่

เลือกใช้ตามงานและความถนัด ไม่จำเป็นต้องใช้ทุกอย่าง

ติดตาม EP-2 ต่อ – เจาะลึก Code/Claude Code ผ่าน VS Code (พื่อนี้จะสอนผ่าน VS Code เป็นหลัก)