

สร้างวิดีโอ ด้วยการคุยกับ AI

คู่มือทำวิดีโอด้วย HyperFrames และ Claude Code
สำหรับคนไทยที่ไม่รู้โค้ดเลย

PROMPT-FIRST

ภาษาไทยกึ่งเล่น

FOR NON-CODERS

TABLE OF CONTENTS • สารบัญ

สารบัญทั้งเล่ม

หนังสือแบ่งเป็นสามภาค: เริ่มต้นและตั้งค่า สร้างวิดีโอจริง และ template/แก้ปัญหา/สรุป workflow ภาคผนวกท้ายเล่มเก็บ prompt copy-ready และคำศัพท์

PART I • เริ่มต้นและตั้งค่า • บทที่ 1-4

01	หนังสือเล่มนี้ทำอะไร และไม่ทำอะไร ภาพรวม prompt-first ทั้งเล่ม + กฎ preview-first	05
02	เตรียมเครื่องแบบไม่ต้องรู้โค้ด เปิด Terminal ติดตั้ง Claude Code ใช้ CLI ขั้นต่ำสุด	11
03	ตั้งค่าโปรเจกต์วิดีโอแรกด้วย Prompt สร้างไฟล์เดอร้งาน ให้ Claude อ่าน official HyperFrames repo	17
04	ให้ Claude ติดตั้ง HyperFrames Skills เอง ติดตั้ง skills จาก official repo / รู้จัก 4 skills หลัก	24

PART II • สร้างวิดีโอจริง • บทที่ 5-8

05	สร้างวิดีโอแรกด้วย Prompt เดี่ยว โครง prompt 8 ส่วน + workflow preview-first	33
06	แก้วิดีโอแบบคุยกับ AI เหมือน Editor ลูป feedback → แก้ → preview จนพอใจ	40
07	ใช้ HyperFrames Catalog โดยไม่ต้องจำชื่อ เพิ่ม caption · transition · effect จาก official catalog	46
08	ใส่เสียงบรรยายด้วย Gemini TTS ตั้งค่า Google Cloud + gcloud และ sync เสียงใน preview	53

TABLE OF CONTENTS • สารบัญ

สารบัญ (ต่อ)

แต่ละบจบเป็นตัวของตัวเอง ไม่จำเป็นต้องอ่านเรียงจาก 1 ถึง 11 แต่บท 1-4 ควรอ่านก่อน เพราะเป็นพื้นฐานของทั้งเล่ม

PART III • ไขข้อสงสัยและแก้ปัญหา • บทที่ 9-11

09	ทำวิดีโอจากไอเดีย 5 แบบที่ใช้จริง Template ความรู้ ขาย สรุป chart และ quote	63
10	วิธีสั่งให้ Claude แก้ปัญหาเองเมื่อพัง Prompt กลาง + วิธีจัดการ error เป็นระบบ	72
11	Workflow สั้นที่สุดตั้งแต่เริ่มจนได้วิดีโอ Master prompt + workflow 10 นาที + ทฎ 7 บั๊อ	80

ภาคผนวก • APPENDICES

A	Prompt สั้นพร้อมใช้ คลัง prompt 9 แบบสำหรับ copy ไปใช้ทันที	88
B	คำศัพท์ที่ต้องรู้แบบภาษาคนทั่วไป Glossary 18 คำสำคัญ — Claude Code, preview, render และอื่นๆ	95

PART ONE

I

เริ่มต้นและตั้งค่า

สื่อกแรกพวกคุณตั้งแต่ยังไม่รู้ว่า Claude Code คืออะไร ไปจนถึงเปิดโปรเจกต์ที่ Claude อ่าน HyperFrames repo และติดตั้ง skills เสร็จแล้ว

บทที่ หนึ่ง • CHAPTER ONE

01

หนังสือเล่มนี้ทำอะไร และไม่ทำอะไร

หนังสือเล่มนี้สอนวิธีสร้างเว็บไซต์ด้วยการคุยกับ AI ไม่ใช่หนังสือสอนเขียนโค้ด เป้าหมายคือให้คุณสร้างเว็บไซต์ได้ด้วยภาษาไทยธรรมดา โดยใช้ **Claude Code** เป็นผู้ช่วย และใช้ **HyperFrames** เป็นเครื่องมือสร้างเว็บไซต์

ถ้าคุณພພພພออกได้ว่าอยากได้เว็บไซต์ไหน คุณก็เริ่มได้

✘ ไม่ใช่หนังสือสอนเขียนโค้ด

✘ ไม่ใช่คู่มือ HTML / CSS / JavaScript

✘ ไม่ใช่คอร์สตัดต่อวิดีโอแบบมืออาชีพ

■ ภาพรวมแบบสั้นที่สุด

เราจะใช้สองอย่างหลัก คุณไม่ต้องเข้าใจว่าไฟล์เว็บข้างในเขียนอย่างไรในระดับลึก หน้าแรกของคุณคือบทความต้องการให้ชัด ส่วนหน้าที่ของ Claude คือจัดการงานเทคนิค

01

Claude Code

ผู้ช่วย AI ที่ทำงานในไฟลเดอร์โปรเจกต์ให้เราได้ เช่น สร้างไฟล์ ติดตั้งของที่เป็น แก้ error และสั่ง render

02

HyperFrames

เครื่องมือที่ AI สร้างวิดีโอจากไฟล์เว็บ แล้วแปลงออกมาเป็นไฟล์วิดีโอ MP4 ที่เอาไปใช้ได้จริง

ตัวอย่างสิ่งที่คุณจะสั่งได้

นี่คือสไลด์ของทั้งเล่ม คุณสั่งเป็นภาษาไทย แล้วให้ Claude ทำงาน

prompt · claude code

```
> สร้างวิดีโอแนวตั้ง 9:16 ความยาว 15 วินาที  
เรื่อง "3 วิธีเริ่มใช้ AI ในงานประจำวัน"
```

```
ขอสไลด์กันสมัย ตัวหนังสือใหญ่ อ่านง่าย  
มี caption ภาษาไทย และจบด้วยคำว่า  
"เริ่มวันนี้ ง่ายกว่าที่คิด"
```

```
ช่วยสร้างไฟล์ทั้งหมด เปิด HTML preview  
ให้ฉันดูก่อน แก้ error และยังไม่ต้อง render  
MP4 จนกว่าฉันจะสั่ง final render
```

สร้างวิดีโอสั้นแนวตั้งด้วย AI
ด้วย **HyperFrames + Claude Code**

Prompt ไทย

- 📄 สไลด์: 3 สไลด์เท่านั้น
- 🗣️ คำพูดภาษาไทย: 50 คำเท่านั้น
- 📺 โหมด: ไม่ใส่เสียง หรือใส่เสียง
- 🕒 ระยะเวลา: 15 วินาที
- ➔ รูปถ่าย: 300 ไฟล์ในแอมวี (0-10)

HTML Preview

- 🔗 ดูตัวอย่างก่อนเรนเดอร์
- 🔍 ตรวจสอบเนื้อหาบนภาพ
- 🔄 เปลี่ยนภาษา SRT
- 🔗 พรีวิวเสียงก่อนเรนเดอร์

Final MP4

MP4
ไฟล์วิดีโอพร้อมใช้งาน

ภาพรวม workflow ทั้งหมด · คนอ่านสั่งด้วย prompt ภาษาไทย ดูวิดีโอผ่าน preview ก่อน แล้วค่อยทำ MP4 ตอน final เท่านั้น

■ Workflow ของทั้งเล่ม

ทั้งเล่มจะเดินตามเส้นทางนี้ หลังจากติดตั้ง Claude Code แล้ว เราจะพยายามไม่ให้คุณกลับไปจำคำสั่ง terminal เองอีก

- 1 ติดตั้ง Claude Code
- 2 เปิด Claude Code ในโฟลเดอร์งาน
- 3 ให้ Claude อ่าน official HyperFrames repo
- 4 ให้ Claude ติดตั้ง HyperFrames skills เอง
- 5 สั่ง Claude สร้างวิดีโอแรก
- 6 ดู preview แล้วสั่งแก้ด้วยภาษาไทย
- 7 ให้ Claude ใช้ของสำเร็จรูปจาก catalog
- 8 ทำเสียงบรรยายด้วย Gemini TTS
- 9 เมื่อ preview พอใจแล้ว ค่อย render เป็น MP4
- 10 ถ้าพัง ให้ Claude อ่าน error และแก้เอง

■ วิธีคิดที่ต้องเปลี่ยนก่อนเริ่ม

คนส่วนใหญ่ติดตรงนี้: เห็นเครื่องมือชื่อ Claude Code แล้วคิดว่าต้องเขียนโค้ดเป็น **ไม่จำเป็น** ในหนังสือเล่มนี้ เราจะมอง Claude Code เป็น "คนทำงานเทคนิค" ไม่ใช่ "โปรแกรมสำหรับโปรแกรมเมอร์เท่านั้น"



คุณไม่ต้องถามคำถามเหล่านี้

- ต้องเขียนไฟล์อะไร · ต้องลง package อะไร
- ต้องใช้คำสั่ง render แบบไหน · error นี้แปลว่าอะไร



ให้ตามแบบนี้แทน

ฉันต้องการผลลัพธ์นี้ ช่วยดูว่าต้องสร้างไฟล์อะไร ติดตั้งอะไร และทำขั้นตอนทั้งหมดให้ฉันได้ไฟล์ MP4 ถ้ามี error ให้คุณอ่านแล้วแก้เองก่อน อธิบายเฉพาะที่ฉันจำเป็นต้องรู้

ให้ Claude อ่าน official source ก่อนเสมอ

เครื่องมือสาย AI เปลี่ยนเร็วมาก วิธีติดตั้งหรือชื่อ model อาจเปลี่ยนได้ แทนที่หนังสือจะบังคับให้คุณจำคำสั่งตายตัว เราจะสอนให้คุณสั่ง Claude แบบนี้

```
prompt · official-first

> ก่อนทำงานนี้ ให้คุณอ่าน official docs
หรือ official GitHub repo ล่าสุดก่อน
แล้วใช้วิธีที่เป็น official เท่านั้น

ถ้าข้อมูลในเว็บกับสิ่งที่ฉันจำได้ไม่ตรงกัน
ให้ยึดข้อมูลจาก official source ล่าสุด
```

■ HyperFrames Skills คืออะไร

ถ้า Claude Code เป็นผู้ช่วย **HyperFrames skills** ก็คือคู่มือเฉพาะทางที่ทำให้ผู้ช่วยทำวิดีโอด้วย HyperFrames ได้ถูกวิธี คุณไม่จำเป็นต้องรู้รายละเอียดทุก skill จำแค่สั้นๆนี่พอ

- `/hyperframes` ใช้เวลาต้องการสร้างหรือแก้วิดีโอ
- `/hyperframes-cli` ใช้เวลาต้องการให้ Claude จัดการ preview, render, check
- `/hyperframes-registry` ใช้เวลาต้องการเพิ่มของจาก catalog
- `/hyperframes-media` ใช้เวลาทำงานเสียง ไฟล์ media หรือ asset

ในหลายกรณี คุณไม่ต้องพิมพ์ชื่อ skill เองด้วยซ้ำ แค่บอก Claude ว่า "ใช้ **HyperFrames skills ที่เหมาะสม**"

■ Prompt ที่ดีหน้าตาเป็นอย่างไร

Prompt ที่ดีไม่ใช่ prompt ยาวเสมอไป Prompt ที่ดีต้องบอกห้าอย่างให้ชัด

- 1 ต้องการทำอะไร
- 2 ผลลัพธ์สุดท้ายคือไฟล์อะไร
- 3 สไตล์ภาพและเสียงเป็นแบบไหน
- 4 ให้ Claude รับผิดชอบขั้นตอนไหนบ้าง
- 5 ถ้า error ให้ Claude ทำอย่างไรต่อ

■ ผลลัพธ์ที่คุณควรได้หลังอ่านเล่มนี้

เมื่ออ่านจบ คุณควรทำสิ่งเหล่านี้ได้

- เปิด Claude Code และเริ่มโปรเจกต์วิดีโอใหม่
- ให้ Claude อ่าน HyperFrames official repo ก่อนทำงาน
- ให้ Claude ตัดตั้ง HyperFrames skills เอง
- สร้างวิดีโอสั้นจาก prompt ภาษาไทย
- สั่งแก้ไข ข้อความ จังหวะ transition และ layout
- ใช้ catalog ของ HyperFrames โดยไม่ต้องจำชื่อ component
- สร้างเสียงบรรยายด้วย Gemini TTS โดยให้ Claude ตรวจสอบ official docs ก่อน
- ใช้ preview ระหว่างแก้ และ render วิดีโอเป็น MP4 ตอน final
- ให้ Claude แก้ error เองเมื่อเกิดปัญหา

KEY TAKEAWAY

คุณจะใช้ ภาษาไทย สั่ง AI ให้สร้างวิดีโอ แทนการเปิดโปรแกรมตัดต่อ และแทนการเขียนโค้ดเอง

→ [บทที่ 02 · เตรียมเครื่องแบบไม่ต้องรู้โค้ด](#)

READ ON

บทที่ สอง • CHAPTER TWO

02

เตรียมเครื่อง แบบไม่ต้องรู้โค้ด

บทที่แล้วเราวางภาพรวมแล้วว่าคุณจะใช้ภาษาไทยสั่ง Claude Code ให้ทำงานเทคนิคแทนคุณ บทนี้
ทำแค่อย่างเดียว: ทำให้ Claude Code เปิดใช้งานได้

ยังไม่ต้องติดตั้ง HyperFrames ยังไม่ต้องสร้างวิดีโอ ยังไม่ต้องเข้าใจโค้ด ขอแค่เปิด Claude Code
ให้สำเร็จก่อน

✘ ยังไม่ต้องติดตั้ง HyperFrames

✘ ยังไม่ต้องสร้างวิดีโอ

✘ ยังไม่ต้องเข้าใจโค้ด

■ สิ่งที่ต้องมีก่อนเริ่ม

คุณต้องมี 3 อย่าง

- 1 คอมพิวเตอร์ที่ต่ออินเทอร์เน็ต
- 2 บัญชี Claude ที่ใช้ Claude Code ได้
- 3 Terminal หรือ PowerShell สำหรับติดตั้งและเปิด Claude Code

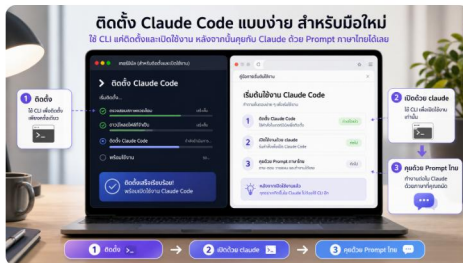


เรื่องบัญชี

Claude Code ไม่ใช่ Claude chat ธรรมดาในเว็บอย่างเดียว จาก official docs ต้องใช้บัญชีที่รองรับ Claude Code เช่น Pro, Max, Team, Enterprise หรือ Console – ถ้าใช้ free plan แล้วเปิดไม่ได้ ให้ตรวจสอบสิทธิ์บัญชีก่อนตรวจเครื่อง

Terminal คืออะไร

Terminal คือช่องที่เราพิมพ์คำสั่งให้คอมพิวเตอร์ ในหนังสือนี้เราใช้ Terminal น้อยมาก หลักๆ แค่ 2 ครั้งเท่านั้น: ติดตั้ง Claude Code และเปิด Claude Code ด้วยคำว่า `cClaude`



Terminal มีหน้าแค่สองอย่าง · ติดตั้ง Claude Code และเปิดมัน หลังจากนั้นจะย้ายไปอยู่ใน prompt ภาษาไทยแทนทั้งหมด

■ เปิด Terminal / PowerShell

เลือกตามเครื่องของคุณ

M

macOS

เปิดแอปชื่อ Terminal · หาได้จาก Spotlight โดยกด **⌘ + Space** แล้วพิมพ์ Terminal

W

Windows

เปิด PowerShell · กด Start แล้วพิมพ์ PowerShell — ถ้ามีตัวเลือก Terminal ก็ใช้ได้

L

Linux

เปิดแอป Terminal ตามปกติ

■ ติดตั้ง Claude Code

ใช้คำสั่งจาก official docs ของ Claude Code เลือกให้ตรงกับเครื่องของคุณ

macOS / Linux / WSL

```
terminal
$ curl -fsSL https://claude.ai/install.sh | bash
```

Windows PowerShell

```
powershell
$ irm https://claude.ai/install.ps1 | iex
```

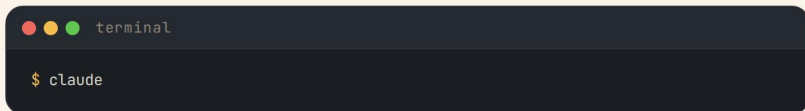


หลังติดตั้งเสร็จ

ปิด Terminal / PowerShell แล้วเปิดใหม่ 1 ครั้ง — บางเครื่องต้องเปิดหน้าต่างใหม่ก่อนจะรู้จักคำสั่ง `claude` ไม่ต้องเข้าใจเรื่อง PATH ตอนนี้ แค่ปิดแล้วเปิดใหม่

■ เปิด Claude Code

ใน Terminal / PowerShell ใหม่ พิมพ์

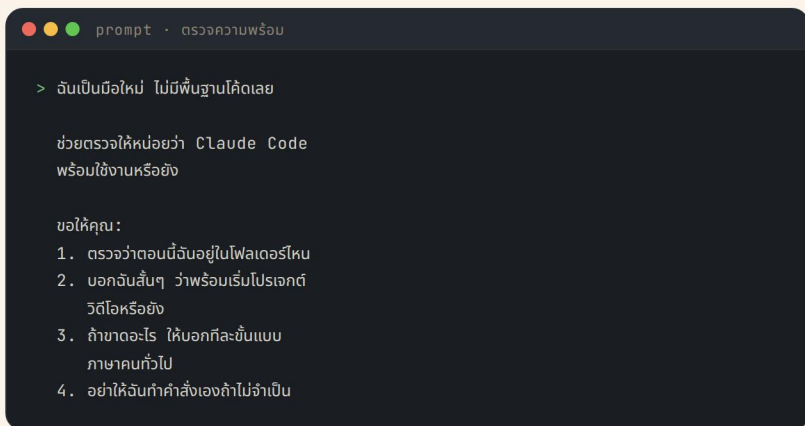


```
terminal  
$ claude
```

แล้วกด Enter ครั้งแรก Claude Code อาจให้คุณ login ผ่าน browser ให้ทำตามขั้นตอนบนหน้าจอ ถ้าทุกอย่างสำเร็จ คุณจะเห็นหน้าจอที่พร้อมให้คุยกับ Claude Code

■ Prompt แรกหลังเปิด Claude Code

เมื่อเข้า Claude Code ได้แล้ว ให้พิมพ์ prompt นี้



```
prompt - ตรวจสอบพร้อม  
  
> ฉันเป็นมือใหม่ ไม่มีพื้นฐานโค้ดเลย  
  
ช่วยตรวจให้หน่อยว่า Claude Code  
พร้อมใช้งานหรือยัง  
  
ขอให้คุณ:  
1. ตรวจสอบว่าตอนนี้ฉันอยู่ในโฟลเดอร์ไหน  
2. บอกฉันสั้นๆ ว่าพร้อมเริ่มโปรเจกต์  
วิธีใดหรือยัง  
3. ถ้าขาดอะไร ให้บอกทีละขั้นตอน  
ภาษาคุณทั่วไป  
4. อย่าให้ฉันทำคำสั่งเองถ้าไม่จำเป็น
```

Claude อาจตอบว่าอยู่ในโฟลเดอร์บ้านของเครื่อง เช่น Desktop, Users, Home หรือชื่ออื่น ไม่เป็นไร ทนนี้ยังไม่ต้องเริ่มโปรเจกต์ แค่ยืนยันว่า Claude Code ใช้งานได้

■ ถ้าพิมพ์ "claude" แล้วไม่เจอคำสั่ง

อาการที่พบบ่อยคือ Terminal บอกประมาณว่า `claude: command not found` ให้ทำตามนี้ตามลำดับ

- 1 ปิด Terminal / PowerShell
- 2 เปิดใหม่
- 3 ลองพิมพ์ `claude` อีกครั้ง
- 4 ถ้ายังไม่ได้ ให้ติดตั้งซ้ำด้วยคำสั่ง official ของเครื่องคุณ

ถ้ายังไม่สำเร็จ ให้ copy ข้อความ error เก็บไว้ แล้วใช้ prompt นี้ใน Claude chat บนเว็บหรือเครื่องมือ AI ที่ใช้อยู่ชั่วคราว

```
prompt - แก้ command not found

> ฉันกำลังติดตั้ง Claude Code แต่เปิด
คำสั่ง claude ไม่ได้

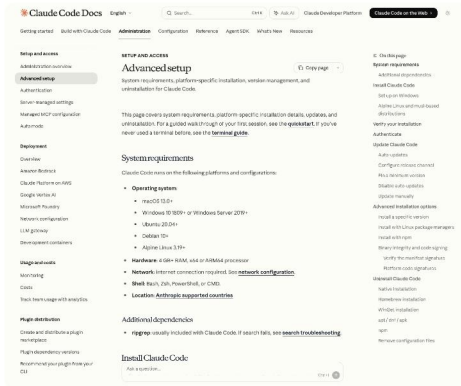
นี่คือ error ที่ขึ้น:
[วาง error ตรงนี้]

ฉันไม่รู้โค้ด ช่วยบอกวิธีแก้แบบทีละขั้น
สำหรับเครื่องของฉัน โดยใช้วิธี official
และไม่ให้ฉันทำอะไรจำเป็น
```



อย่าเพิ่งติดตั้ง HyperFrames ตอนนี้

ถึงแม้ HyperFrames จะเป็นเครื่องมือหลักของหนังสือ แต่ยังไม่ต้องติดตั้งในขณะนี้ เราต้องการให้ Claude Code เป็นคนอ่าน official HyperFrames repo แล้วตัดสินใจวิธีติดตั้งที่ถูกต้องเอง



หน้า **official setup** ของ Claude Code · ภาพนี้ช่วยยืนยันว่าเราจัดวิธีจาก official docs ไม่ใช่คำสั่งที่จำต่อๆ กัน มา ถ้าหน้าเว็บเปลี่ยน ให้ยึดข้อมูลล่าสุดเสมอ

■ เช็กก่อนจบบท

ก่อนข้ามไปบทถัดไป คุณควรทำได้ 3 อย่าง

- เปิด Terminal / PowerShell ได้
- พิมพ์ `cLaude` แล้วเข้า Claude Code ได้
- ส่ง prompt แรกให้ Claude Code ตอบได้

KEY TAKEAWAY

หลังบทนี้คุณจะไม่**ต้อง**จำคำสั่ง **terminal** อีก นอกจากเปิด Claude — งานที่เหลือจะย้ายเข้าไปอยู่ใน prompt ภาษาไทยทั้งหมด

→ **บทต่อไป**
บทที่ 03 · ตั้งค่าโปรเจกต์วิดีโอแรกด้วย Prompt

READ ON

บทที่ สาม • CHAPTER THREE

03

ตั้งค่าโปรเจกต์ วิดีโอแรกด้วย Prompt

ตอนนี้คุณเปิด Claude Code ได้แล้ว บทนี้เราจะสร้างพื้นที่ทำงานสำหรับวิดีโอแรก และสั่งให้ Claude อ่าน official HyperFrames repo ก่อนทำอะไร

เป้าหมายคือให้ Claude เข้าใจว่าโปรเจกต์นี้จะใช้ HyperFrames และต้องยึดข้อมูลจาก **official source**

✘ ยังไม่ต้องสร้างวิดีโอ

✘ ยังไม่ต้องติดตั้ง skills

✘ ยังไม่ต้องจำคำสั่ง HyperFrames

■ โปรเจกต์คืออะไร

โปรเจกต์ในที่นี้คือไฟล์เดอร์หนึ่งไฟล์เดอร์ที่เก็บทุกอย่างของวิดีโอ

- ไฟล์งานวิดีโอ
- รูปภาพ
- เสียง
- ไฟล์ preview
- ไฟล์ MP4 ที่ render เสร็จ

เราแยกเป็นไฟล์เดอร์เพื่อไม่ให้ไฟล์ปนกับงานอื่น

ตั้งค่าโปรเจกต์วิดีโอแรกด้วย HyperFrames ใน Claude Code

✓ ผลลัพธ์ของขั้นตอนนี้: ไฟล์พร้อมใช้งาน + โครงสร้างพร้อม + เพิ่มสร้างวิดีโอด้วย HyperFrames

เป้าหมายของบทนี้ - มีไฟล์เดอร์งานที่เป็นระเบียบ ให้ Claude อ่าน official repo ก่อน และยังไม่รับสร้างวิดีโอจริงจนกว่าจะพร้อม

■ สร้างโฟลเดอร์งานแบบง่าย

คุณเลือกได้ 2 วิธี

วิธีที่ง่ายที่สุด: สร้างด้วยเมาส์

- 1 ไปที่ Desktop
- 2 สร้างโฟลเดอร์ใหม่ ตั้งชื่อว่า `hyperframes-video`
- 3 เปิด Terminal / PowerShell ในโฟลเดอร์นั้น

ถ้าไม่รู้วิธีเปิด Terminal ในโฟลเดอร์ ให้ใช้วิธีคำสั่งด้านล่างแทน

วิธีใช้คำสั่งครั้งเดียว

macOS / Linux

```
terminal
$ mkdir -p ~/Desktop/hyperframes-video && cd ~/Desktop/hyperframes-video && claude
```

Windows PowerShell

```
powershell
$ mkdir "$HOME\Desktop\hyperframes-video"; cd "$HOME\Desktop\hyperframes-video";
claude
```

คำสั่งนี้ทำสามอย่างติดกัน: สร้างโฟลเดอร์บน Desktop เข้าไปในโฟลเดอร์นั้น และเปิด Claude Code นี่เป็นหนึ่งในไม่กี่ครั้งที่เราชอบใช้คำสั่ง

■ ให้ Claude ตรวจสอบว่าอยู่ถูกโฟลเดอร์

เมื่อ Claude Code เปิดในโฟลเดอร์ `hyperframes-video` แล้ว ให้ส่ง prompt นี้

```
prompt · ตรวจสอบโฟลเดอร์

> ช่วยตรวจสอบให้หน่อยว่าตอนนี้เราอยู่ใน
โฟลเดอร์โปรเจกต์วิดีโอที่ถูกต้องหรือยัง

สิ่งที่ต้องการ:
1. บอก path โฟลเดอร์ปัจจุบัน
2. ตรวจสอบว่าโฟลเดอร์นี้เหมาะสำหรับเริ่ม
โปรเจกต์ HyperFrames หรือไม่
3. ถ้ายังไม่เหมาะ ให้บอกวิธีแก้แบบสั้นที่สุด
4. ยังไม่ต้องติดตั้งอะไร และยังไม่ต้อง
สร้างวิดีโอ
```

✓ ถ้าโฟลเดอร์ถูก

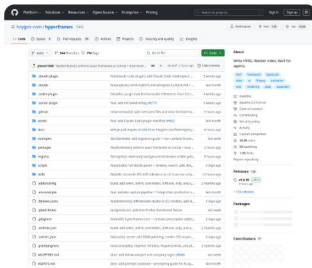
Claude ตอบว่าอยู่ถูกแล้ว ให้ไปต่อในหน้าถัดไป

✗ ถ้าโฟลเดอร์ผิด

ให้ทำตามที Claude แนะนำแบบสั้นๆ – ปกติคือสั่งให้ Claude พาเปลี่ยนโฟลเดอร์ให้

■ ให้ Claude อ่าน HyperFrames official repo

เราจะไม่เริ่มจากสิ่งที่ Claude 'จำได้' แต่จะให้ Claude อ่าน source จริงก่อน



official GitHub repo ของ HyperFrames - เวลาเห็น prompt ให้ Claude อ่าน repo นี้ให้เข้าใจว่าเราใช้หน้า official เป็นฐาน ไม่ได้ให้ AI เดาสุ่มจากความจำเก่า

prompt · อ่าน HyperFrames official repo

> ฉันต้องการสร้างโปรเจกต์วิดีโอด้วย
HyperFrames ในโฟลเดอร์นี้

ก่อนทำอะไร ให้คุณอ่าน official source
เหล่านี้ก่อน:

1. GitHub official ของ HyperFrames:
<https://github.com/heygen-com/hyperframes>
2. documentation และ catalog ที่ repo
official แนนะ

หลังจากอ่านแล้ว ให้สรุปสั้นๆ ว่า
HyperFrames ใช้ทำอะไร workflow พื้นฐาน
คืออะไร และขั้นตอนไปควรทำอะไร

สำคัญ:

- ยังไม่ต้องติดตั้งอะไร
- ยังไม่ต้องสร้างโฟลเดอร์วิดีโอ
- ใช้ข้อมูลจาก official source เท่านั้น

■ สิ่งที่ Claude ควรสรุปได้

คำตอบของ Claude ไม่จำเป็นต้องเหมือนกันทุกเครื่อง แต่ควรมีใจความประมาณนี้

- HyperFrames ใช้สร้างวีดิโอจาก HTML/CSS/JS
- เหมาะกับ AI agent เพราะ AI เขียนไฟล์เหล่านี้แทนคนได้
- preview ได้ก่อน render
- render ออกมาเป็น MP4
- มี skills และ catalog ช่วยให้งานง่ายขึ้น
- ขั้นตอนต่อไปคือให้ Claude ดึงตั้ง HyperFrames skills ตามวิธี official

ถ้าคำตอบยาวเกินไป ให้สั่ง: 'สรุปใหม่ให้เหลือไม่เกิน 8 บรรทัด เขาเฉพาะสิ่งที่ฉันต้องรู้เพื่อเริ่มท้าวดีโอ ไม่ต้องลงลึกเรื่องเทคนิค'

■ สร้างไฟล์บันทึกเป้าหมายโปรเจกต์

ขั้นนี้ไม่จำเป็น แต่แนะนำให้ Claude สร้างไฟล์ `PROJECT_BRIEF.md` ไว้ในโฟลเดอร์โปรเจกต์ เพื่อให้ทุก session ต่อจากนี้รู้ว่าโปรเจกต์นี้ทำอะไร

```
prompt · สร้าง PROJECT_BRIEF.md

> ช่วยสร้างไฟล์บันทึกเป้าหมายโปรเจกต์ชื่อ
PROJECT_BRIEF.md ในโฟลเดอร์นี้

เนื้อหาให้สืบและชัดเจน:
- โปรเจกต์ใช้ HyperFrames + Claude Code
- ผู้ใช้ไม่รู้โค้ด ใช้ prompt ภาษาไทย
- Claude ต้องอ่าน official docs ก่อนติดตั้ง
- ใช้ HTML preview จนพอใจ แล้วค่อย render MP4
- ถ้ามี error ให้ Claude แก้เองก่อน
```

■ อย่าเพิ่งทำวิดีโอแรก

ตอนนี้คุณอาจจะอยากลองสั่งสร้างวิดีโอทันที แต่อย่าเพิ่ง ลำดับที่ดีกว่าคือ

- 1 Claude อ่าน HyperFrames official repo แล้ว
- 2 Claude เข้าใจเป้าหมายโปรเจกต์แล้ว
- 3 ติดตั้ง HyperFrames skills ก่อน
- 4 ค่อยสร้างวิดีโอแรก

ถ้าข้าม skills ไป Claude ยังอาจทำได้ แต่มีโอกาสเขียนโครงงานไม่ตรง pattern มากกว่า

■ เช็กก่อนจบบท

- โฟลเดอร์ `hyperframes-video`
- เปิด Claude Code ในโฟลเดอร์นั้นได้
- Claude อ่าน official HyperFrames repo แล้ว
- Claude สรุป workflow พื้นฐานให้คุณเข้าใจแล้ว
- มีไฟล์ `PROJECT_BRIEF.md` ถ้าคุณเลือกให้ Claude สร้าง

→ [undo ไป](#)
บทที่ 04 · ให้ Claude ติดตั้ง HyperFrames Skills เอง

READ ON

บทที่ สี่ • CHAPTER FOUR

04

ให้ Claude ติดตั้ง HyperFrames Skills เอง

บทที่แล้ว Claude อ่าน official HyperFrames repo แล้ว บทนี้เราจะให้ Claude ติดตั้ง
HyperFrames skills

จุดสำคัญคือคุณไม่ต้องจำคำสั่งติดตั้งเอง ให้ Claude อ่านวิธีจาก official repo แล้วจัดการ

✘ อย่าพิมพ์คำสั่งติดตั้งเองจากความจำ

✘ อย่าเดาชื่อ skill

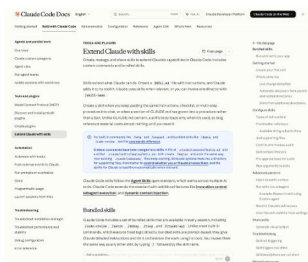
✘ ไม่ต้องจำทุก skill – จำแค่ 4 ตัว

■ Skills คืออะไร

skills คือชุดคำแนะนำเฉพาะทางที่ทำให้ Claude Code ทำงานบางอย่างได้ถูกวิธีขึ้น ในกรณีนี้ HyperFrames skills จะช่วยให้ Claude รู้ว่า

- โครงสร้างวิดีโอ HyperFrames ควรเป็นแบบไหน
- จะ preview อย่างไร
- จะ render เป็น MP4 อย่างไร
- จะใช้ catalog block/component อย่างไร
- จะจัดการ caption, transition, audio และ media อย่างไร

ถ้าไม่มี skills Claude ยังอาจพอทำงานได้ แต่เหมือนให้ช่างทำงานโดยไม่มีคู่มือเฉพาะทาง



Skills เป็นแนวคิด official ของ Claude Code · เราแก้ไข Claude ติดตั้ง skill ที่เหมาะกับงาน HyperFrames โดยอ่านวิธีจาก official repo

■ หลักการของบทนี้

อย่าพิมพ์คำสั่งติดตั้งเองจากความจำ ให้ Claude ทำตามนี้แทน

- 1 อ่าน official HyperFrames repo
- 2 หาวิธีติดตั้ง skills จาก repo
- 3 รั้นคำสั่งที่เป็น official เอง
- 4 ตรวจสอบว่าติดตั้งสำเร็จไหม
- 5 สรุปลikeคุณรู้เฉพาะสิ่งที่ต้องใช้

นี่คือวิธีที่ปลอดภัยกว่า เพราะถ้า HyperFrames เปลี่ยนวิธีติดตั้งในอนาคต Claude จะเห็นจาก official source ก่อน

Claude Code ติดตั้ง HyperFrames Skills ให้อัตโนมัติ

อ่านจากแหล่งทางการ → ติดตั้ง skills → ตรวจสอบ → พร้อมใช้งาน

- 1

แหล่งทางการ

คลิกดูจาก Official Repository
www.HyperFrames
- 2

ติดตั้ง skills

ตามวิธีการ
ติดตั้งในวิดีโอ
- 3

ตรวจสอบ

ตรวจสอบจนเห็น
installate



Claude Skill เชิงขอ
เมื่อจำเป็น

- ติดตั้งจำนวน
- ระบุคำสั่งติดตั้ง
- ตรวจสอบสถานะติดตั้ง
- ติดตั้งเฉพาะบางตัว

พร้อมแล้วจึงวิดีโอแรก

หลังจาก 4 พ้องเสียง
นั้นจะจบวิดีโอแรก

ลำดับที่ต้องการ · Claude อ่านแหล่งทางการ ติดตั้ง skills ตรวจสอบว่าพร้อมใช้ แล้วค่อยเริ่มสร้างวิดีโอแรก

■ Prompt ติดตั้ง HyperFrames skills

ส่ง prompt นี้ใน Claude Code ที่เปิดอยู่ในโฟลเดอร์โปรเจกต์

```
prompt · ติดตั้ง skills จาก official repo

> จาก official HyperFrames repo ที่คุณ
อ่านแล้ว ช่วยติดตั้ง HyperFrames skills
ให้ Claude Code ใช้งานในโปรเจกต์นี้

ให้ทำตามนี้:
1. ตรวจสอบวิธีติดตั้ง skills จากหน้า
   official repo: github.com/heygen-com/hyperframes
2. ใช้วิธี official เท่านั้น
3. ถ้าต้องรับคำสั่ง ให้คุณรับเอง
4. ถ้าระบบถาม permission ให้บอกฉันทันที
   ว่าคำสั่งนั้นทำอะไร ก่อนให้ฉันอนุมัติ
5. หลังติดตั้งเสร็จ ให้ตรวจว่ามี skills
   สำคัญอะไรใช้งานได้น่าง
6. สรุปเป็นภาษาไทยแบบสั้น ไม่เกิน 10 บรรทัด

ถ้าติด error ให้แก้เองก่อน
```

ถ้า Claude ขอรันคำสั่ง ให้รู้ว่าคำสั่งนั้นมาจาก official repo จริง แล้วอนุมัติ

■ ถ้า Claude บอกให้ restart

บางครั้งหลังติดตั้ง skills แล้ว Claude Code อาจต้อง restart session เพื่อโหลด skills ใหม่ ถ้า Claude บอกให้ restart ให้ทำแบบนี้

- 1 ออกจาก Claude Code ตามวิธีที่หน้าจอนำเสนอ
- 2 เปิด Terminal / PowerShell ในโฟลเดอร์โปรเจกต์เดิม
- 3 พิมพ์ `cclaude`
- 4 กลับมาถาม Claude ว่าเห็น skills แล้วหรือยัง

Prompt ตรวจสอบหลัง restart

```
prompt · ตรวจสอบ skills พร้อมใช้

> ช่วยตรวจให้หน่อยว่า HyperFrames skills
ที่ติดตั้งไว้พร้อมใช้งานหรือยัง

ขอให้คุณ:
1. บอกชื่อ skills ที่เกี่ยวกับ HyperFrames
ที่พบ
2. อธิบายสั้นๆ ว่าแต่ละตัวใช้ทำอะไร
3. บอกว่าตอนนี้พร้อมเริ่มสร้างวิดีโอแรก
หรือยัง
4. ถ้ายังไม่พร้อม ให้แก้หรือแนะนำขั้นตอน
ถัดไปแบบสั้นที่สุด
```

■ Skills ที่ควรรู้ – จำแค่ 4 ตัว

หลังติดตั้งแล้ว คุณอาจเห็นหลาย skill ไม่ต้องจำทุกตัว จำเฉพาะตัวที่ใช้บ่อยพอ

01

/hyperframes

ใช้เมื่อต้องการสร้างหรือแก้วิดีโอ – เป็นตัวหลักที่ใช้บ่อยที่สุด

02

/hyperframes-cli

ใช้เมื่อเกี่ยวกับขั้นตอนทำงาน เช่น init, preview, lint, inspect, render

03

/hyperframes-registry

ใช้เมื่อต้องการติดตั้ง block หรือ component จาก catalog

04

/hyperframes-media

ใช้เมื่อต้องการจัดการเสียง ไฟล์ media หรือ asset



Skills animation อื่นๆ

เช่น `/gsap`, `/css-animations`, `/lottie`, `/three`, `/waapi` – กลุ่มนี้ใช้เมื่ออยากได้ animation เฉพาะทาง ไม่ต้องเริ่มจากกลุ่มนี้ในวิดีโอแรก

■ วิธีเรียกใช้ skill แบบง่าย

แบบที่ 1 – เรียกชื่อ skill ตรงๆ

```
prompt · เรียก skill โดยตรง
```

```
> ใช้ /hyperframes สร้างวิดีโอแนะนำสินค้า  
15 วินาที แนวตั้ง 9:16
```

แบบที่ 2 – ให้ Claude เลือกเอง

```
prompt · ให้ Claude เลือก skill
```

```
> ช่วยใช้ HyperFrames skills ที่เหมาะสมเพื่อ  
สร้างวิดีโอนี้และเปิด HTML preview ให้ฉัน  
ดูก่อน ยังไม่ต้อง render MP4 จนกว่าฉันจะ  
สั่ง final render
```

สำหรับมือใหม่ แบบที่ 2 ง่ายกว่า แต่ถ้าอยากให้ Claude เข้า context แม่นขึ้น ให้ใส่ `/hyperframes` ไปด้วย

เลือกแบบไหนเมื่อไหร่?

A

ไม่แน่ใจ ใช้แบบ 2

บอกเป้าหมายเป็นภาษาไทย ให้ Claude เลือก skill ที่เหมาะให้เอง – เริ่มต้นง่ายสุด

B

ทำซ้ำหลายรอบ ใช้แบบ 1

ถ้าเริ่มเข้าใจว่า skill โหนดเหมาะกับงานแบบไหน เรียกตรงๆ จะเร็วและตรงกว่า

■ ถ้าติดตั้งแล้วมี error

อย่าพยายามแก้เองด้วยการสุ่มคำสั่ง ให้ส่ง prompt นี้

```
prompt · แก้ error ติดตั้ง skills
```

> การติดตั้ง HyperFrames skills ยังไม่สำเร็จ

ช่วยทำตามนี้:

1. อ่าน error ล่าสุดทั้งหมด
2. ตรวจสอบจาก official HyperFrames repo ว่าปัญหานี้เกี่ยวกับอะไร
3. ถ้าขาด dependency เช่น Node, Git, Git LFS หรืออย่างอื่น ให้บอกฉันสั้นๆ
4. ถ้าคุณแก้เองได้ ให้แก้เอง
5. ถ้าจำเป็นต้องให้ฉันทำอะไร ให้บอกทีละขั้น และบอกเหตุผลสั้นๆ
6. หลังแก้แล้ว ให้ตรวจซ้ำว่า skills พร้อมใช้งานหรือยัง

KEY TAKEAWAY

หน้าที่ของคุณคือ **อนุมัต** หรือทำตามเฉพาะขั้นตอนที่ Claude อธิบายชัดเจน ไม่ต้องจำ error

[ไปดู](#)

→ [บทที่ 05 - สร้างวิดีโอแรกด้วย Prompt เดียว](#)

[READ ON](#)

PART TWO



สร้างวิดีโอจริง

สี่บทกลางคือหัวใจของเล่ม — สร้างวิดีโอแรก แก้แบบคุยกับ editor ใช้ catalog ของ HyperFrames และใส่เสียงบรรยายด้วย Gemini TTS

บทที่ ห้า • CHAPTER FIVE

05

สร้างวิดีโอแรก ด้วย Prompt เดียว

ตอนนี้โปรเจกต์พร้อมแล้ว Claude อ่าน official HyperFrames repo แล้ว และติดตั้ง HyperFrames skills แล้ว บทนี้เราจะสร้างวิดีโอแรก แต่มีหลักสำคัญหนึ่งข้อ

ระหว่างลองงาน ให้ดูจาก HTML preview ก่อน อย่า render MP4 ทุกครั้ง

✘ ห้าม render MP4 ระหว่าง iterate

✘ ห้ามทำวิดีโอแรกซ้ำซ้อนเกินไป

✘ ห้ามใส่เกิน 3 ประเด็นในวิดีโอแรก

■ ทำไม preview-first

การ render MP4 ช้ากว่ามาก เพราะต้องแปลงทุก frame เป็นไฟล์วิดีโอจริง ตอนทำงานจริง workflow ที่เร็วกว่าคือ

- 1 ให้ Claude สร้าง/แก้ไฟล์ HTML ของ HyperFrames
- 2 เปิด preview แล้วกด play ดูใน browser
- 3 สั่งแก้จากสิ่งที่เห็น
- 4 วนแบบนี้จนพอใจ
- 5 ค่อย render MP4 final ครั้งเดียว

1 Prompt เดียว
สร้างวิดีโอ 9:16
เกี่ยวกับ 3 เทคนิคเพิ่มสมาธิ
สำหรับการทำงาน
โทนเป็นมิตร เข้าใจง่าย
ความยาว 45 วินาที

2 HTML Preview ก่อน
✓ ดูตัวอย่าง
แก้ไขไฟล์ HTML ก่อน
• แก้ข้อความ
• เปลี่ยนภาพ/วิดีโอ
• เปลี่ยนระยะเวลา
• เปลี่ยนเพลง/เสียง

3 เทคนิค เพิ่มสมาธิ สำหรับการทำงาน
1. แก้ไฟล์ HTML
2. ฝึกงานเป็นงาน
3. สอนตัวเองไป

ยังไม่ Render MP4
ตอนนี้เป็นแค่ Preview HTML เท่านั้น
ยังไม่สามารถ Render เป็นวิดีโอ (MP4) เพราะใช้เวลา

4 Final เมื่อพอใจ
Final Video.mp4
อันนี้เสร็จ

Workflow ใช้งาน: Prompt ใช้งาน → HTML Preview → แก้ไขจนพอใจ → Render เป็น MP4

จังหวะทำงานจริง - เริ่มจาก prompt เดียว ให้ Claude สร้าง HTML preview ก่อน แก้จนพอใจ แล้วค่อยกดผลิตขึ้น final render เป็น MP4

■ วิดีโอแรกควรเรียบง่าย

อย่าเริ่มด้วยวิดีโอที่ซับซ้อนเกินไป วิดีโอแรกควรมีแค่นี้พอ

- ความยาว 10–15 วินาที
- แนวตั้ง 9:16
- ข้อความ 3 ประเด็น
- พื้นหลังเคลื่อนไหวง่ายๆ
- caption อ่านง่าย
- outro สั้นๆ

■ โครง prompt สำหรับวิดีโอ – 8 ส่วน

เวลาสั่ง Claude ให้สร้างวิดีโอ ให้บอกแปดอย่างนี้ – ครั้งแรกเป็นข้อมูลพื้นฐาน ครั้งหลังเป็นสไตล์และเงื่อนไข

1

สัดส่วน

9:16 · 16:9 · square

2

ความยาว

เช่น 12 วินาที

3

หัวข้อ

วิดีโอพูดเรื่องอะไร

4

กลุ่มคนดู

ใครจะดูวิดีโอนี้

5

สไตล์

สดใส · มินิมอล · พรีเมียม

6

ข้อความ

ประเด็นหลักที่ขึ้นในจอ

7

โหมดทำงาน

ทำเป็น HTML preview ก่อน

8

เงื่อนไข final

render เมื่อสั่ง final

✓

ผลลัพธ์

ได้ HTML preview · ไม่มี MP4ทันที

■ Prompt สร้างวิดีโอแรกแบบ preview-first

ใช้ prompt นี้ได้เลย — เปลี่ยนเฉพาะหัวข้อและประเด็น

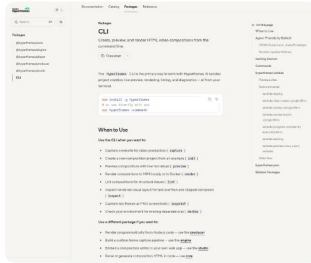
```
prompt · สร้างวิดีโอแรก

> ใช้ /hyperframes สร้างวิดีโอแนวตั้ง 9:16
ความยาว 12 วินาที หัวข้อ
"3 เหตุผลที่ควรเริ่มใช้ AI ทำวิดีโอ"

สไลด์ภาพ:
- สดใส ทันสมัย เหมาะกับ TikTok/Reels
- พื้นหลัง gradient เคลื่อนไหวเบาๆ
- ตัวหนังสือภาษาไทยใหญ่ อ่านง่าย
- มี caption เล็งตามจังหวะ แต่ไม่รก

โครงวิดีโอ:
1. hook: "ทำวิดีโอด้วย AI ไม่ต้องเขียนโค้ด"
2. ประเด็น 1: "เริ่มจาก prompt ภาษาไทย"
3. ประเด็น 2: "ให้ AI จัดการไฟล์และ preview"
4. ประเด็น 3: "แก้ไขได้เหมือนคุยกับ editor"
5. CTA: "ลองทำวิดีโอแรกวันนี้"

ให้คุณทำอะไร:
- สร้างไฟล์ที่จำเป็น
- ใช้ HyperFrames skills ที่เหมาะสม
- เปิด HTML preview ให้ฉันกด play q
- ตรวจสอบว่าข้อความไทยไม่ล้นจอ
- ยังไม่ต้อง render MP4 จนกว่าฉันจะ
พิมพ์ว่า "final render"
```



HyperFrames CLI · เรื่อง preview/check/render เป็นงานฝั่งเครื่องมือ แต่ในเล่มนี้คุณไม่ต้องจำคำสั่ง ให้ Claude เป็นคนจัดการและอธิบายก่อนนะ

■ ถ้า Claude รีบ render MP4

บางครั้ง Claude อาจทำตามนิสัยเดิม คือสร้างเสร็จแล้ว render ทันที ให้หยุดด้วย prompt นี้

```

prompt · หยุดก่อน render

> หยุดก่อน ยังไม่ต้อง render MP4

ตอนนี้ฉันต้องการดู HTML preview ใน browser ก่อน เพื่อแก้ไข

ช่วยทำให้ preview เล่นได้ แล้วบอกฉันว่า
ต้องเปิด URL หรือไฟล์ไหนเพื่อกด play ดู
so feedback จากฉันก่อน อย่า render จวบ
กว่าฉันจะสั่งว่า "final render"
    
```



จำไว้

preview ก่อน render เสมอ — เป็นกฎที่ใช้ตลอดเล่ม

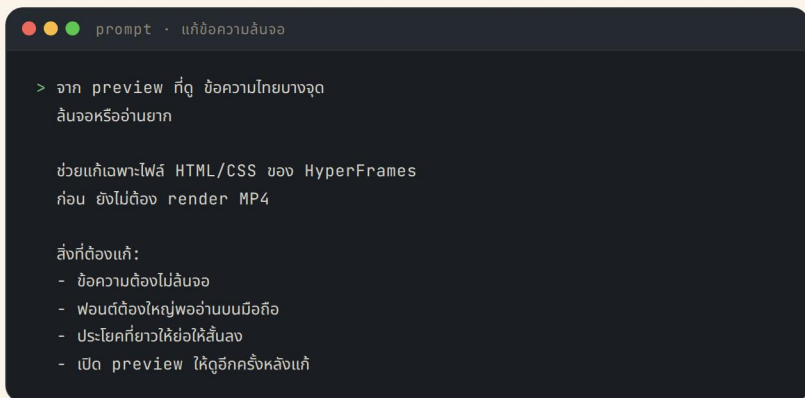
■ วิธีดูวิดีโอแรก

เมื่อ Claude เปิด preview หรือบอก URL ให้คุณแล้ว ให้ดู 5 อย่าง

- hook 2 วินาทีแรกถึงความสนใจไหม
- ข้อความไทยอ่านง่ายไหม
- มีอะไรสั้นจ่อไหม
- animation สกเกินไปไหม
- call to action ตอนท้ายชัดไหม

ยังไม่ต้องสนใจความ perfect ขอแค่เห็นว่าโครงวิดีโอถูกทาง

■ ถ้าข้อความสั้นจ่อ



■ ถ้าพอใจแล้วค่อย final render

เมื่อดู preview แล้วพอใจ ค่อยสั่ง render เป็น MP4

```
prompt · final render

> ตอนนี้จับพ้องกับ preview แล้ว

ช่วยทำ final render เป็น MP4

ให้คุณ:
1. ตรวจสอบไฟล์ HyperFrames ล่าสุด
2. render เป็น MP4 คุณภาพเหมาะสมสำหรับ
   ส่งงาน
3. ตั้งชื่อไฟล์ output-final-v1.mp4
4. ถ้ามี error ตอน render ให้แก้เอง
5. หลังเสร็จ ให้ออก path ของไฟล์ MP4 final

ตอนนี้อนุญาตให้ render MP4 ได้
```

คำว่า "ตอนนี้อนุญาตให้ render MP4 ได้" ช่วยให้ Claude รู้ว่าผ่านขั้น preview แล้ว

KEY TAKEAWAY

ใช้ HTML preview เป็นวงจรหลักตอน iterate และ render MP4 เฉพาะตอน final หรือ milestone เท่านั้น

undo ไป
→ [บทที่ 06 · แก้ววิดีโอแบบคุยกับ AI เหมือนคุยกับ Editor](#)

READ ON

บทที่ หก • CHAPTER SIX

06

แก้วิดีโอแบบ คุยกับ Editor

บทที่แล้วคุณสร้างวิดีโอและดูผ่าน HTML preview แล้ว บทนี้คือการ **iterate** หรือทำงานหลายรอบ

หลักสำคัญ: ระหว่างแก้ ให้ Claude อัปเดต HTML preview ก่อน อย่า render MP4 ทุกครั้ง — render MP4 ใช้ตอน final หรือ milestone เท่านั้น

✘ อย่าสั่งกว้างๆ ว่า 'ทำให้สวยขึ้น'

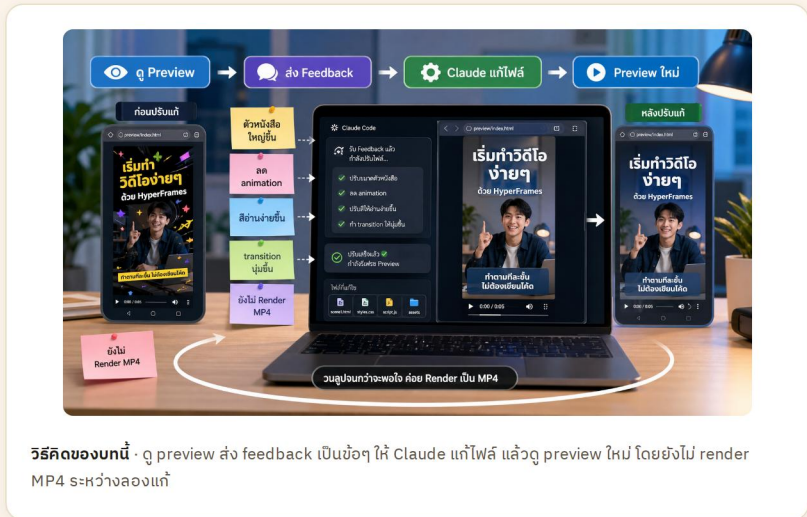
✘ อย่า render MP4 ทุกครั้งที่แก้

✘ อย่าให้ Claude ปรับดีไซน์ใหม่บ่อยๆ ตอน render

■ Workflow แก้งานที่เร็วที่สุด

ใช้แบบนี้

- 1 ดู preview ใน browser
- 2 จดว่าอยากแก้ไขอะไร
- 3 ส่ง feedback เป็นข้อๆ ให้ Claude
- 4 Claude แก้ HTML / CSS / asset
- 5 กด refresh หรือ play ดูใหม่
- 6 ทำซ้ำจนพอใจ
- 7 ค่อย final render MP4



■ วิธีให้ feedback ที่ดี

อย่าสั่งกว้างๆ ว่า "ทำให้สวยขึ้น" อย่างเดียว ให้บอกเป็นหมวด



ภาพ

สี · ฟอนต์ · ขนาดตัวอักษร · ความรกของพื้นหลัง



จังหวะ

ความเร็ว scene · timing caption · transition



เนื้อหา

ข้อความสั้นลง · ความยาวรวม · CTA ชัดขึ้น

และย้ำว่า "ยังไม่ต้อง render"

■ Prompt แก่รอบแรก

prompt · แก่รอบแรก

> จาก preview ล่าสุด ช่วยแก้วีดีโอนี้ให้ดูดีขึ้น โดยยังคงเนื้อหาเดิม

สิ่งที่ต้องปรับ:

- ทำภาพรวมให้สะอาดขึ้น ไม่รก
- เพิ่ม contrast ให้ตัวหนังสืออ่านง่าย
- ทำตัวหนังสือใหญ่ขึ้น 15-20%
- ลด animation ที่ทำให้เสียสมาธิ
- ทำ transition ระหว่างประเด็นให้นุ่มขึ้น

ให้คุณ:

1. แก้ไฟล์ HTML/CSS/asset ที่จำเป็นเอง
2. อัปเดต preview ให้ฉันกด play ดู
3. ยังไม่ต้อง render MP4
4. รอ feedback จากฉันก่อน

■ แก้สีให้ตรง mood

แนวพรี่เมียม

prompt · mood พรี่เมียม

> จาก preview ล่าสุด ช่วยปรับ mood สีให้
ดูพรี่เมียมขึ้น

แนวทาง:

- ใช้สีหลักเป็นดำ ทอง ขาว
- พื้นหลังเรียบขึ้น
- ลดสีสดที่ไม่จำเป็น
- ให้ตัวหนังสือเด่นและอ่านง่าย
- transition ดูนิ่งและแพจ ไม่ตึงเกินไป

แก้ไข HTML preview ก่อน ยังไม่ต้อง render
เปิด preview ให้จับดูอีกครั้งหลังแก้

แนวสดใส

prompt · mood สดใส

> จาก preview ล่าสุด ช่วยปรับ mood สีให้
สดใส เหมาะกับ TikTok/Reels

แนวทาง:

- ใช้สีหลักไม่เกิน 3 สี
- พื้นหลังมี movement เบาๆ
- ตัวหนังสือสว่าง อ่านง่าย
- animation สุกๆ แต่ไม่รก
- จังหวะโดยรวมเร็วขึ้นเล็กน้อย

■ ให้ feedback แบบเจาะจงหลังดู preview

✗ แบบไม่ดี

ยังไม่สวย แก้ให้หน่อย

✓ แบบดี

หลังดู preview แล้ว ขอแก้ 4 จุดนี้:

1. ตัวหนังสือช่วง hook ยังเล็กไป เพิ่มอีก 20%
2. พื้นหลัง scene 2 เคลื่อนไหวเยอะ ลดลงครึ่งหนึ่ง
3. transition ก่อน outro เร็วไป ทำให้บูมขึ้น
4. CTA ตอนท้ายอยากให้อ่านนานขึ้น 1 วินาที

feedback ที่เป็นข้อๆ ทำให้ Claude แก้ตรงกว่า ใช้ตัวเลข 1-4 เสมอ

■ เมื่อไหร่ควร render MP4 ระหว่างทาง

โดยทั่วไปไม่ควร render ทุกครั้ง แต่ render ได้ใน 3 กรณี

- 1 ลูกคำหรือทีมต้องดูไฟล์ MP4 จริง
- 2 ต้องตรวจสอบเสียงกับภาพแบบ final
- 3 preview พอใจแล้วและจะจบงานรอบนี้

ถ้าแค่แก้สี ข้อความ timing หรือ layout ให้ใช้ preview wv

■ Prompt final render หลัง iterate จบ

```

prompt · final render sau iterate

> ตอนนี้ฉันพอใจกับ preview ล่าสุดแล้ว

ช่วย final render เป็น MP4

ให้คุณ:
1. ตรวจสอบ preview ล่าสุดอีกครั้ง
2. render เป็น MP4 ใหม่ชื่อ
   output-final-v2.mp4
3. ถ้ามี error ตอน render ให้แก้เอง
4. ตรวจสอบไฟล์ MP4 เปิดได้
5. บอก path ของไฟล์ final

ตอนนี้อนุญาตให้ render MP4 ได้

```

■ บันทึกสถานะหลัง final

```

prompt · บันทึก

> ช่วยสรุปการแก้ไขวีดิโอเรื่องนี้ลง
PROJECT_BRIEF.md

ขอให้บันทึก:
- preview ล่าสุดใช้ไฟล์/URL อะไร
- MP4 final ล่าสุดอยู่ที่ไหน
- สิ่งที่ปรับจากเวอร์ชันแรก
- style ปัจจุบันของวีดิโอ
- จุดที่ยังควรปรับต่อ

```

KEY TAKEAWAY

preview ก่อน · feedback เป็นข้อๆ · render เฉพาะตอน final — นี่คือลูปที่ใช้ตลอดเล่ม

→ [บทต่อไป](#)
 → **บทที่ 07 · ใช้ HyperFrames Catalog โดยไม่ต้องจำชื่อ**

READ ON

บทที่ เจ็ด • CHAPTER SEVEN

07

ใช้ HyperFrames Catalog โดยไม่ต้องจำชื่อ

ตอนนี้คุณทำงานด้วย HTML preview ได้แล้ว บทนี้เราจะเพิ่มลูกเล่นจาก HyperFrames catalog

หลักยังเหมือนเดิม: **เพิ่ม component** → **ดู preview** → **แท่งพอใจ** → **ค่อย render MP4** อย่า render ทุกครั้งหลังเพิ่ม component เพราะจะช้ามาก

✘ อย่าจำชื่อ component เอง

✘ อย่าใส่เกิน 2-3 component ต่อรอบ

✘ อย่าให้ลูกเล่นกลบเนื้อหา

■ Catalog ใช้ทำอะไร

catalog คือคลังของสำเร็จรูป เช่น



caption

caption style สำหรับ
ข้อความบนจอ



transition

จังหวะเปลี่ยน scene



effect

shader · grain · overlay



lower third

แถบข้อมูลล่างจอ



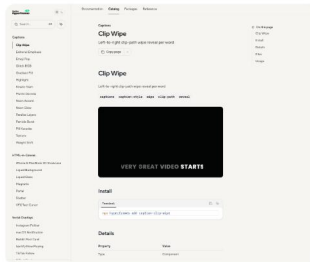
data chart

กราฟ ตาราง ตัวเลข



social overlay

ปุ่ม like share follow



official HyperFrames catalog · เหตุผลที่เราไม่ต้องจำชื่อ component เอง ให้ Claude เปิด official catalog แล้วเลือกของที่เหมาะสมกับปัญหาในวิดีโอ

■ เริ่มจากปัญหา ไม่ใช่ชื่อ component

เช่น

- caption ยังธรรมดาไป
- transition แข็งไป
- ภาพดูแบนไป
- outro ยังไม่เด่น
- อยากให้เหมาะกับ TikTok มากขึ้น

นี่คือภาษาที่คนทำวิดีโอใจจริง — กว่าพูดว่า "ติดตั้ง component ชื่อนั้นชื่อนี้"

ใช้ Catalog อย่างฉลาด ไม่ต้องจำชื่อคอมโพเนนต์

เลือกเท่าที่ช่วยเนื้อหา วิดีโอจะดูดีและเป็นมืออาชีพ

วิธีใช้ catalog ที่ปลอดภัย · เริ่มจากปัญหา เลือกออกจาก catalog ไม่เกิน 2-3 อย่าง ใส่ใน preview แล้วลดทอนที่ถ้าวิดีโอเริ่มรถ

■ Prompt ให้ Claude เลือกจาก catalog

```
prompt · เลือกจาก catalog

> ช่วยเปิด official HyperFrames catalog
แล้วเลือก block/component ที่เหมาะกับ
วิดีโอ preview ล่าสุดของฉัน

เป้าหมาย:
- caption อ่านง่ายและมี movement
- transition ระหว่าง scene นุ่มนวล
- effect เบาๆ ให้ภาพไม่แบน แต่ล้ำรก

เงื่อนไข:
1. ใช้เฉพาะของจาก official catalog
2. เลือกไม่เกิน 3 รายการในรอบนี้
3. ก่อนติดตั้ง ให้สรุปชื่อที่เลือกและ
เหตุผลสั้นๆ
4. ติดตั้งและใส่ในโปรเจกต์เอง
5. อัปเดต HTML preview ให้ฉันกด play ดู
6. ยังไม่ต้อง render MP4

ถ้า component ไหนทำให้ preview พัง
ให้ถอดออกหรือเลือกตัวที่ง่ายกว่า
```

■ เพิ่ม caption style

```
prompt · caption  
  
> ใช้ /hyperframes-registry ช่วยหา caption  
style ที่เหมาะกับวิดีโอภาษาไทย
```

เกณฑ์เลือก:

- อ่านง่ายบนมือถือ
- movement ชัดแต่ไม่เวียนหัว
- รองรับข้อความสั้นหลาย scene
- ไม่ทำให้ Layout สั้นจ่อ

เลือก 1 แบบ ติดตั้ง ใส่ในวิดีโอ
แล้วอัปเดต HTML preview ให้ดู

■ เพิ่ม transition

```
prompt · transition  
  
> ใช้ /hyperframes-registry หา transition  
จาก catalog ที่เหมาะกับวิดีโอสั้นเพื่อ  
ความรู้
```

ต้องการ:

- transition เรียบและนุ่ม
- ไม่ flashy เกินไป
- ไม่ทำให้อ่านข้อความยาก
- ใช้เฉพาะช่วงเปลี่ยนประเด็นหลัก

เลือก 1 แบบที่เหมาะสมที่สุด ใส่ในวิดีโอ
แล้วเปิด preview ให้ดู

■ เพิ่ม effect เบาๆ

```
prompt · effect

> ช่วยเลือก effect เบาๆ จาก catalog
เพื่อทำให้วิดีโอ texture มากขึ้น

ต้องการ:
- grain หรือ shimmer เบาๆ ได้
- ห้ามทำให้ข้อความอ่านยาก
- ห้ามเพิ่มความรกของพื้นหลัง
- ใช้ opacity ต่ำพอประมาณ
```

■ ถ้าเพิ่ม component แล้ว preview พัง

```
prompt · แก้ preview พัง

> หลังเพิ่ม component/block แล้ว preview
มีปัญหา

ช่วยทำตามนี้:
1. อ่าน error ล่าสุด
2. ระบุว่า component ไหนน่าจะเป็นต้นเหตุ
3. ถ้าแก้ไขเข้ากับโปรเจกต์ได้ ให้แก้เอง
4. ถ้าไม่คุ้น ให้ถอดออก
5. คืน preview ให้เล่นได้ก่อน
6. ยังไม่ต้อง render MP4
```

ความเสถียรมาก่อนลูกเล่น

■ อย่านำให้ catalog กลบเนื้อหา

ง่ายๆ

- 1 caption style พอ
- 1 transition หลักพอ
- 1 effect เบาๆ พอ
- ทุกอย่างต้องช่วยให้ข้อความชัดเจน

ถ้าดู preview แล้วรก ให้ลด

prompt · ลอดลูกเล่นที่รก

> จาก preview ล่าสุด วิดีโอเริ่มรกเกินไป

ช่วยลดลูกเล่นจาก catalog ให้เหลือเฉพาะ
ที่จำเป็น

ให้คุณ:

- เก็บเฉพาะ caption style ที่ช่วยให้อ่านง่าย
- ลอดหรือถอด effect ที่ไม่จำเป็น
- ใช้ transition ให้น้อยลง
- ทำภาพรวมให้สะอาดขึ้น

KEY TAKEAWAY

catalog คือ เครื่องเร่ง ไม่ใช่กระดาษห่อ — ใช้เฉพาะเมื่อช่วยให้ข้อความชัดเจน

undo

→ บทที่ 08 · ใส่เสียงบรรยายด้วย Gemini TTS

READ ON

บทที่ แปด • CHAPTER EIGHT

08

ใส่เสียงบรรยาย ด้วย Gemini TTS

ถึงตอนนี้วิดีโอของคุณมีภาพ ข้อความ caption และลูกเล่นจาก catalog แล้ว บทนี้เราจะเพิ่ม **เสียงบรรยาย**

เป้าหมาย: ให้ Claude สร้างเสียงจากข้อความภาษาไทย ใส่เสียงลงในวิดีโอ HyperFrames แล้วให้เรากด play ตรวจสอบใน HTML preview ก่อน เมื่อเสียงกับภาพตรงจังหวะแล้วค่อย render

✘ อย่าเดาชื่อ model เอง

✘ อย่าแปะ API key ลง chat

✘ อย่า render ก่อน sync ใน preview

■ TTS คืออะไร

TTS ย่อมาจาก text-to-speech แปลตรงตัวคือ "ข้อความเป็นเสียงพูด" เราจะเขียนสคริปต์เสียงเป็นภาษาไทย แล้วให้ Gemini TTS สร้างไฟล์เสียง จากนั้นให้ Claude ใส่ไฟล์เสียงเข้าไปในวิดีโอ

- คุณไม่ต้องตัดเสียงเอง
- ไม่ต้องลากไฟล์เข้า timeline
- ไม่ต้อง sync ด้วยมือ

■ เส้นทางที่แนะนำ — Google Cloud + gcloud

สำหรับงานในเล่มนี้ เหมาะที่สุดคือใช้ Gemini TTS ผ่าน Google Cloud โดยให้ Claude จัดการผ่าน gcloud และ official docs

- ใช้กับโปรเจกต์จริงได้เป็นระบบ
- จัดการ project, billing, permission ได้ชัดเจน
- ไม่ต้องแปะ API key ลง prompt
- ใช้ Application Default Credentials จาก gcloud ได้
- มี Free Trial สำหรับผู้ใช้ใหม่ตามเงื่อนไข Google Cloud

■ Google Cloud Free Trial – สิ่งที่ต้องรู้



เครดิตฟรีและวันหมดอายุ

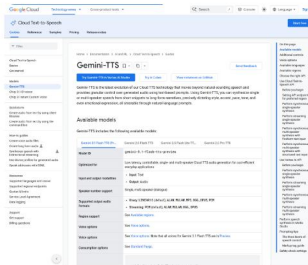
จาก official Google Cloud Free Program: ผู้ใช้ใหม่ที่เข้าเงื่อนไขจะได้ Welcome credit ตามที่ docs ระบุ ใช้ได้ในระยะเวลาที่กำหนด และระหว่าง Free Trial จะไม่ถูก bill สำหรับ usage ในช่วง trial ตามเงื่อนไขของโปรแกรม

พูดแบบภาษาคนทั่วไป: ถ้าคุณยังไม่เคยใช้ Google Cloud แบบเสียเงินมาก่อน ให้สมัคร Free Trial สิทธิประโยชน์เพื่อยืนยันตัวตน แล้วใช้เครดิตฟรีทดลองทำเสียงได้ก่อน



ต้องรู้ 3 เรื่อง

1. เครดิตฟรีมีวันหมดอายุ
2. หลังหมด trial หรือใช้เครดิตหมด ต้องดู billing ให้ดี
3. บริการและเงื่อนไขอาจเปลี่ยนได้ให้ Claude ตรวจสอบ official docs ล่าสุดก่อนเสมอ



official Google Cloud docs สำหรับ Gemini TTS · ใช้ย่่าว่าเส้นทางในบทนี้คือ official docs และให้ Claude ตรวจสอบข้อมูลล่าสุดก่อนเสมอ

■ ชื่อ model ต้องให้ Claude ตรวจจาก official docs

คุณอาจเรียกติดปากว่า "Gemini 3 Flash TTS"

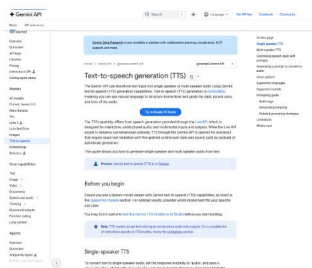
แต่ชื่อ model จริงใน official docs อาจเป็น `gemini-3.1-flash-tts-preview`, `gemini-2.5-flash-tts` หรือชื่ออื่นในอนาคต ขึ้นอยู่กับว่าคุณใช้ผ่าน Google AI API, Vertex AI API หรือ Cloud Text-to-Speech API



อย่าให้หนังสือหรือคนอ่านเดาชื่อ model เอง

ให้ Claude ทำ 4 อย่างนี้:

- อ่าน official Gemini TTS docs ล่าสุด
- อ่าน Google Cloud Gemini-TTS docs ล่าสุด
- เลือก Flash TTS model ที่ official docs รองรับจริงในตอนนั้น
- ถ้าชื่อที่เราพูดไม่ตรงกับ docs ให้บอกตรงๆ และใช้ชื่อ official ที่ถูกต้อง



official Gemini speech generation docs · ถ้า Claude พบชื่อ model ไม่ตรงกับคำที่เราพูด ให้ใช้ชื่อใน docs เป็นหลัก

■ ตั้งค่า Google Cloud แบบให้ Claude พาเดิน

ขั้นนี้คนอ่านอาจต้องกดในเว็บเอง เช่น สมัคร Google Cloud, ใส่วิธีชำระเงิน, เปิด billing, สร้าง project, enable API แต่ไม่ต้องรู้ทั้งหมดล่วงหน้า ให้ Claude เป็นคนบอกทีละขั้น

```
prompt · ตั้งค่า Google Cloud + gcloud
```

```
> ฉันต้องการใช้ Gemini Flash TTS ผ่าน  
Google Cloud/gcloud เพื่อสร้างเสียง  
บรรยายให้วิดีโอ HyperFrames
```

ช่วยพาฉันตั้งค่าตาม official docs ทีละขั้น

ให้คุณ:

1. อ่าน Free Trial docs ว่าได้เครดิตเท่าไร
2. อ่าน docs ติดตั้ง gcloud CLI
3. อ่าน docs ของ Gemini-TTS / Vertex AI
4. บอกฉันว่าต้องทำอะไรในเว็บ Google Cloud
เช่น สมัคร Free Trial, สร้าง project,
เปิด billing, enable API
5. ถ้าเครื่องยังไม่มี gcloud ให้บอกวิธี
ติดตั้งจาก official docs
6. หลังติดตั้งแล้ว ให้พาฉัน login
ด้วย gcloud และตั้งค่า project
7. ใช้ Application Default Credentials
ถ้าเหมาะสม
8. ห้ามให้ฉันแปะ API key ลง chat

■ เขียนสคริปต์เสียงให้พอดีกับวิดีโอ

เสียงบรรยายต้องสั้นกว่าวิดีโอเล็กน้อย ไม่ควรพูดแน่นจนไม่มีจังหวะหายใจ สำหรับวิดีโอ 12 วินาที สคริปต์ควรสั้นมาก ประมาณ 1-3 ประโยค

prompt · เขียนสคริปต์ 3 เวอร์ชัน

> ช่วยเขียนสคริปต์เสียงบรรยายภาษาไทย สำหรับวิดีโอนี้

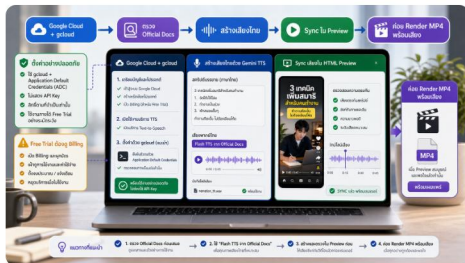
เงื่อนไข:

- ความยาววิดีโอ 12 วินาที
- เสียงต้องพูดจบภายใน 10-11 วินาที
- น้ำเสียงสดใส เข้าใจง่าย
- ภาษามกทั่วไป ไม่เป็นการกำกวม

ขอ 3 เวอร์ชัน:

1. เวอร์ชันกระชับ
2. เวอร์ชันสนุก
3. เวอร์ชันพรีเมียม

ยังไม่ต้องสร้างเสียง รอให้ฉันเลือกก่อน



ภาพรวมของบทนี้ · ตั้งค่า Google Cloud อย่างปลอดภัย ตรวจสอบ official docs ก่อนเลือก model สร้างเสียงไทย ใส่เสียงใน preview แล้วค่อย render MP4 พร้อมเสียงเมื่อพอใจ

■ สร้างไฟล์เสียงด้วย Gemini TTS

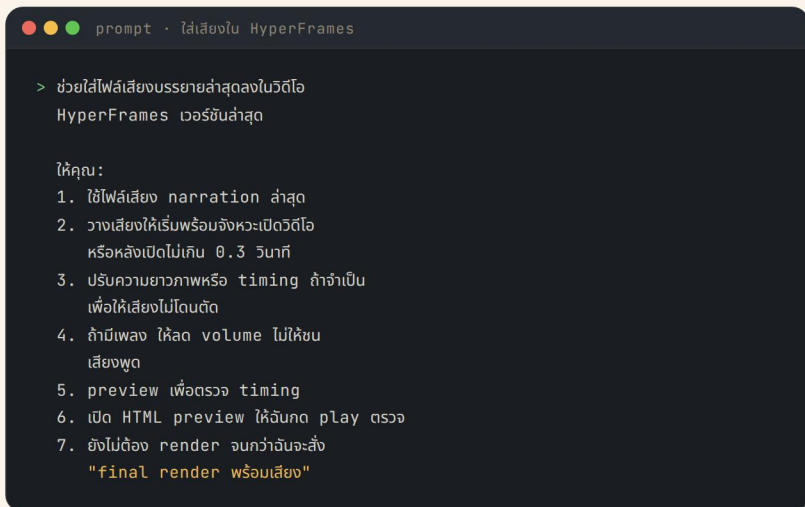
เมื่อเลือกสคริปต์แล้ว ให้สั่งสร้างเสียง

prompt · สร้างไฟล์เสียง

```
> ใช้สคริปต์ที่เลือกไว้ สร้างไฟล์เสียงด้วย
Gemini TTS ผ่าน Google Cloud/gcloud
ตาม official docs ล่าสุด

เงื่อนไข:
- ใช้ Google Cloud project ที่ตั้งค่าไว้
- ใช้ Application Default Credentials
หรือวิธี official ที่ปลอดภัย
- ใช้ Flash TTS model ที่ official docs
รองรับจริงตอนนี้
- ถ้ามี Gemini 3.x Flash TTS ใน docs
ให้ใช้ตัวนั้น
- ถ้าไม่มี ให้ใช้ Flash TTS model
official ที่เหมาะที่สุด และบอกฉัน
ว่าใช้ชื่อ model อะไร
- เลือกเสียงที่เหมาะสมกับวิดีโอสั้น
- เก็บไฟล์ใน assets/audio
- ตั้งชื่อ narration-v1.wav หรือ .mp3
- ห้ามแสดง token / credential / API key
ในคำตอบ
```

ใส่เสียงเข้าไปในวิดีโอ



ถ้าเสียงยาวเกินวิดีโอ ให้ย่อสคริปต์ก่อน ไม่ใช่ขีดวิดีโอทุกครั้ง

5 จุดที่ Claude ควรตรวจให้อัตโนมัติ

<p>●</p> <p>Latency เปิด</p> <p>เสียงเริ่มไม่เกิน 0.3 วินาทีหลังภาพแรก</p>	<p>●</p> <p>ไม่ตัดท้าย</p> <p>เสียงจบก่อนภาพจบ ≥ 0.5 วิ</p>	<p>●</p> <p>Volume</p> <p>เสียงพูดดังกว่าเพลง/effect</p>
<p>●</p> <p>Sync จังหวะ</p> <p>เสียงเปลี่ยนช่วงตรงกับจุด transition หลัก</p>	<p>●</p> <p>ไม่ทับ caption</p> <p>เสียงไม่พูดซ้ำเดียวกับข้อความบนจอ</p>	<p>●</p> <p>Loop test</p> <p>ฟังซ้ำ 2 รอบ ไม่มีความสะดุด</p>

■ Checklist ตรวจวิดีโอพร้อมเสียง

- preview เปิดเล่นได้
- มีเสียงบรรยายจริง
- เสียงเริ่มถูกจังหวะ
- เสียงไม่โดนตัดท้าย
- เสียงพูดดังชัดกว่าเพลงหรือ effect
- ข้อความบนจอกับเสียงไม่ขัดกัน
- ความยาววิดีโอเหมาะสม

■ Final render พร้อมเสียง

```

prompt · final render + เสียง

> ตอนนี้ฉันพอใจกับ preview พร้อมเสียงแล้ว

ช่วย final render เป็น MP4 พร้อมเสียง

ให้คุณ:
1. ตรวจสอบ preview ล่าสุดอีกครั้ง
2. render เป็นไฟล์ output-final-with-voice-v1.mp4
3. ถ้ามี error ตอน render ให้แก้เอง
4. ตรวจสอบว่า MP4 เปิดได้และมีเสียง
5. บอก path ของไฟล์ final

ตอนนี้อนุญาตให้ render MP4 ได้

```

KEY TAKEAWAY

เสียงดีไม่ใช่แค่ 'อ่านถูก' แต่ต้อง เข้ากับ mood ของวิดีโอ – sync ใน preview ก่อนเสมอ

→ [บทต่อไป](#)
 → [บทที่ 09 · ทำวิดีโอจากไอเดีย 5 แบบที่ใช้จริง](#)

READ ON

PART THREE



ใช้จริงและแก้ปัญหา

สามบทสุดท้ายคือคลัง template ที่ใช้จริง วิธีสั่ง Claude แก้ปัญหาเอง
เมื่อพัง และสูตร workflow รวบรวมสำหรับงานเร่ง

บทที่ เก้า · CHAPTER NINE

09

ทำวิดีโอจาก ไอเดีย 5 แบบ

ตอนนี้คุณมี workflow ครบแล้ว สร้าง แก้ ใช้ catalog ใส่เสียง — บทนี้ให้ **prompt template** สำหรับงานจริง 5 แบบ ให้ copy ไปใช้ได้เลย

เปลี่ยนเฉพาะข้อความในวงเล็บเหลี่ยม **[แบบนี้]** ทุก template ในบทนี้ใช้หลัก preview-first

✘ อย่าใส่เกิน 3-4 ประเด็นในวิดีโอ 30 วิ

✘ อย่าใส่ CTA หลายตัวในวิดีโอขาย

✘ อย่ายัดข้อมูลทั้งบทความลงจอเดียว

■ Template ที่จะได้ในบทนี้

01

วิดีโอความรู้ 30 วิ

คลิปสอนสั้น สรุปล 3-5 ประเด็น เหมาะ
TikTok/Reels/Shorts

02

วิดีโอขายสินค้า 15 วิ

โปรโมตสินค้า/บริการ ads สั้น launch offer

03

วิดีโอสรุปบทความ

เอาบทความยาวมาทำคลิปสั้น สรุปลโพสต์ ข่าว

04

วิดีโอ chart / data

ยอดขาย เปรียบเทียบ growth ranking survey

05

วิดีโอ quote / ประกาศ

quote สั้น ประกาศ event teaser ข้อความสร้าง
แรงบันดาลใจ

X

Prompt กลาง

ถ้ายังไม่รู้เริ่มจากไหน ใช้ prompt นี้ให้ Claude
เลือก format ก่อน



5 template + prompt กลาง · ถ้ายังไม่รู้จะเริ่มจากไหน ใช้ prompt กลางให้ Claude ช่วยเลือก format ก่อนสร้าง preview

■ Template 1 – วิดีโอความรู้ 30 วินาที

```
prompt · วิดีโอความรู้

> ใช้ /hyperframes สร้างวิดีโอความรู้
แนวตั้ง 9:16 ความยาว ~30 วินาที

หัวข้อ: "[หัวข้อ]"
กลุ่มเป้าหมาย: [กลุ่ม]
โทน: [เข้าใจง่าย/สนุก/จริงจัง/มืออาชีพ]

โครงวิดีโอ:
1. hook 2 วินาทีแรก: ดึงความสนใจทันที
2. ประเด็นหลัก 3 ข้อ:
  - [ประเด็นที่ 1]
  - [ประเด็นที่ 2]
  - [ประเด็นที่ 3]
3. สรุปท้ายคลิปแบบจำง่าย
4. call to action: "[CTA]"

ขอให้มี:
- caption ภาษาไทยใหญ่ อ่านง่ายบนมือถือ
- ภาพรวมสะอาด ไม่รก
- transition เรียบระหว่างประเด็น
- เสียงบรรยายไทยด้วย Gemini TTS
- HTML preview ก่อน ไม่ render MP4
  จนกว่าจะสั่ง final render
```

เคล็ดลับ: ถ้าวิดีโอยาว 30 วินาที อย่าใส่เกิน 3-4 ประเด็น

■ Template 2 – วิดีโอขายสินค้า 15 วินาที

```
prompt · วิดีโอขายสินค้า

> ใช้ /hyperframes สร้างวิดีโอขายสินค้า
แนวตั้ง 9:16 ความยาว 15 วินาที

สินค้า/บริการ: "[ชื่อสินค้า]"
ขายให้ใคร: [กลุ่ม]
ปัญหาลูกค้า: [ปัญหา]
จุดเด่น: [จุดเด่น 1-3 ข้อ]
call to action: "[คลิกแชท / สั่งวันนี้ / กดลองฟรี]"

โครง:
1. เปิดด้วยปัญหาที่ลูกค้าเจอ
2. แนะนำสินค้าเป็นทางออก
3. โชว์จุดเด่น 2-3 ข้อแบบสั้น
4. ปิดด้วย CTA ชัดเจน

สไตล์:
- [พรีเมียม/สดใส/มีมออล/สนุก]
- ตัวหนังสือไทยใหญ่ อ่านง่าย
- ภาพไม่รก เน้นขายชัด

เสียงบรรยายสั้นด้วย Gemini TTS
เปิด preview ก่อน · ไม่ render MP4
จนกว่าเงินจะสั่ง final
```

เคล็ดลับ: คลิปขาย 15 วินาทีต้องมี CTA เดียว อย่าให้คนดูเลือกหลายอย่าง

■ Template 3 – วิดีโอสรุปบทความ/โพสต์

```
prompt · สรุปบทความ

> ใช้ /hyperframes สร้างวิดีโอสรุป
เนื้อหาแมตต์ 9:16 ~30-45 วินาที

เนื้อหาต้นฉบับ:
[วางบทความ/โพสต์/ไต่ตี่มี]

ช่วยทำตามนี้:
1. อ่านเนื้อหาต้นฉบับ
2. สรุปเป็น 3-5 ประเด็นที่คนทั่วไป
   เข้าใจง่าย
3. เขียน hook 2 วินาทีแรก
4. สร้างวิดีโอจากสรุปนั้น
5. ใช้ caption ภาษาไทยอ่านง่าย
6. เลือก catalog component เฉพาะ
   ที่ช่วยให้อ่านง่าย
7. สร้างเสียงบรรยายภาษาไทย
8. ใส่เสียงแล้วเปิด HTML preview
9. รอฉันทันทีก่อน final render

โทน: [สรุปข่าว/เข้าใจง่าย/มีอาชีพ]
CTA ก๊วยคิลิป: "[CTA]"

ห้ามคิดลอกถึงบทความมายึดลอก
```

■ Template 4 – วิดีโอ Chart / Data

```

prompt · วิดีโอ data

> ใช้ /hyperframes สร้างวิดีโอข้อมูล
แนวตั้ง 9:16 ~20-30 วินาที

หัวข้อ: "[หัวข้อข้อมูล]"
โทน: [มีอารมณ์/เข้าใจง่าย/สนุก]

ข้อมูล:
[วางข้อมูลเป็น list หรือ table ง่ายๆ
เช่น
เดือน, ยอดขาย
มกราคม, 120000
กุมภาพันธ์, 150000
มีนาคม, 210000]

ให้คุณ:
1. ตรวจสอบข้อมูลว่าทำวิดีโอได้ไหม
2. ถ้าข้อมูลไม่ชัด ให้ถามจับต่อน
3. เลือกวิธีนำเสนอที่ง่ายที่สุด
   เช่น bar chart, number count,
   หรือ highlight card
4. ทำให้ตัวเลขอ่านง่ายบนมือถือ
5. ใส่ caption สรุป insight
6. เพิ่มเสียงบรรยาย
7. เปิด HTML preview ให้ตรวจ

Insight ที่อยากให้จำ: "[เช่น ยอดโต 75%]"

```

เคล็ดลับ: วิดีโอข้อมูลต้องมี '**ข้อสรุป**' ไม่ใช่แค่กราฟสวย

■ Template 5 – วิดีโอ Quote / ประกาศ

```

prompt · วิดีโอ quote

> ใช้ /hyperframes สร้างวิดีโอ quote
แนวตั้ง 9:16 ความยาว 8-12 วินาที

ข้อความหลัก:
"[ข้อความหลัก]"

ข้อความรอง ถ้ามี:
"[ข้อความรอง]"

โทนภาพ: [มินิมอล/พรีเมียม/cinematic/สดใส]
CTA: "[เจอกันวันที่... / ติดตามต่อ / สมัครเลย]"

สไตล์:
- ตัวหนังสือเป็นพระเอกของวิดีโอ
- animation น้อยแต่ดูดี
- พื้นหลังไม่แย่งความสนใจ
- เสียงเป็นแบบสลับมากหรือเสียงจ๋า
  quote แบบนุ่ม

สร้าง HTML preview · ตรวจสอบข้อความไทย
ไม่ลื่นจ่อ · ไม่ render MP4 จบกว่าจะ
สั่ง final

```

เคล็ดลับ: วิดีโอ quote ไม่ควรมีข้อความเยอะ ถ้ายาว ให้แบ่งเป็น 2-3 จังหวะ

■ Prompt กลาง – ถ้ายังไม่รู้จะเริ่มจากไหน

```
prompt · กวาง

> ฉันต้องการสร้างวิดีโอด้วย HyperFrames
จากไอเดียนี้:
[วางไอเดียของคุณ]

ช่วยเลือก format ที่เหมาะที่สุดให้เอง
เช่น ความรู้ ขายสินค้า สรุปบทความ
ข้อมูล หรือ quote

ให้คุณ:
1. ลามจับเฉพาะข้อมูลที่จำเป็นถ้ายังขาด
2. เขียนโครงวิดีโอให้ฉันตรวจสอบสั้นๆ ก่อน
3. ใช้ HyperFrames skills ที่เหมาะสม
4. ใช้ catalog เฉพาะถ้าช่วยให้งานดีขึ้น
5. สร้างเสียงด้วย Gemini TTS ถ้าจำเป็น
6. เปิด HTML preview ให้ตรวจ
7. รอฉันอนุมัติก่อน final render

ฉันไม่รู้คิด ใช้ภาษาไทยง่ายๆ
```

prompt นี้เหมาะเวลาคุณมีแค่อิเดีย แต่ยังไม่รู้ว่าควรทำเป็นวิดีโอแบบไหน

■ 3 อย่างที่ควรเติมก่อนกดส่ง

1 ตัวอย่าง style

สไตล์อ้างอิง: คลิปสั้นแบบ startup tech, พื้นหลังสะอาด, ตัวหนังสือใหญ่, motion น้อยแต่คม

2 สิ่งที่ไม่ต้องการ

ห้ามใช้สีเยอะ · ห้ามให้ข้อความเล็ก · ห้ามใส่ effect จนอ่านยาก · ห้าม render MP4 ระหว่างแก้ไข preview

3 หน้าที่ของ Claude

คุณต้องจัดการไฟล์ ติดตั้ง component ตรวจสอบ preview และแก้ไข error เองทั้งหมด ห้าม render MP4 จนกว่าจะสั่ง final render

KEY TAKEAWAY

เลือก template ตามรูปแบบงาน เปลี่ยนค่าใน [...] เติม 3 อย่างนี้ก่อนกดส่ง ทุก template ใช้ preview-first เสมอ

→ [ไปดู](#)
บทที่ 10 · วิธีสั่งให้ Claude แก้ปัญหาเองเมื่อพัง

[READ ON](#)

10

วิธีสั่งให้ Claude แก้ปัญหาเองเมื่อพัง

การทำวิดีโอด้วย AI ไม่ได้แปลว่าจะไม่เจอ error คุณจะเจอแน่นอน แต่สิ่งที่ต้องจำคือ **error ไม่ใช่** งานของคุณคนเดียว

ใน workflow นี้ Claude Code ต้องอ่าน error หาเหตุผล แก้ไฟล์ ติดตั้งสิ่งที่ขาด และทดสอบซ้ำให้มากที่สุด หน้าทีของคุณคือสั่งให้ Claude แก้อย่างเป็นระบบ

✘ อย่ากลัว error

✘ อย่า copy error ไปค้นเอง

✘ อย่าสั่ง render ซ้ำถ้า error เกิดในขั้น preview

■ หลักการแก้ error แบบคนไม่รู้โค้ด

อย่าเริ่มจากการ copy error ไปค้นมั่วๆ ให้ทำตามลำดับนี้

- 1 บอ Claude ว่าเกิดอะไรขึ้น
- 2 ให้ Claude อ่าน error ล่าสุดทั้งหมด
- 3 ให้ Claude หาสาเหตุ
- 4 ให้ Claude แก่ทีละจุด
- 5 ให้ Claude เปิด preview เพื่อตรวจ
- 6 ถ้าเป็น final แล้วค่อย render MP4

อย่าสั่งให้ render ช้าทันทีถ้า error เกิดขึ้นใน preview แก้ preview ให้เล่นได้ก่อน



กฎหลักของบทนี้ - ให้ Claude อ่าน error ล่าสุด ภาษาเหตุ แก้ทีละจุด เปิด preview ช้า และยังไม่ render MP4 ระหว่างแก้

■ Prompt กลางสำหรับแก้ error ทุกชนิด

ใช้ prompt นี้เป็นตัวหลัก — เป็น prompt ที่ควรใช้บ่อยที่สุดตอนพัง

```
prompt · แก้ error กลาง

> งานนี้ยังไม่สำเร็จ ช่วยแก้ไขรูปแบบ
เป็นระบบ

ให้คุณทำตามนี้:
1. อ่าน error ล่าสุดทั้งหมดก่อน
2. อธิบายสาเหตุที่น่าจะเป็นไปได้
   แบบสั้นๆ ภาษาไทย
3. ตรวจสอบไฟล์ที่เกี่ยวข้องในโปรเจกต์
4. แก้เฉพาะจุดที่จำเป็น
5. เปิดหรืออัปเดต HTML preview เพื่อ
   ตรวจสอบว่ากลับมาเล่นได้ไหม
6. ยังไม่ต้อง render MP4 จนกว่าจะจับ
   สิ่ง final render
7. ถ้าจำเป็นต้องให้จับทำอะไร ให้ออก
   ทีละขั้นแบบคนไม่รู้ก็ได้

อย่าให้ฉันแก้ไขเองถ้าไม่จำเป็นจริงๆ
```

■ เมื่อ preview เปิดไม่ได้

```

prompt · preview เปิดไม่ได้

> ตอนนี้ HTML preview เปิดไม่ได้หรือ
เล่นไม่ได้

ช่วยแก้ตามลำดับนี้:
1. ตรวจสอบว่า preview server ยังทำงานไหม
2. อ่าน error ที่ขึ้นทั้งหมด
3. ตรวจสอบไฟล์ HyperFrames หลัก
   เช่น index.html
4. ตรวจสอบ path ของรูป เสียง วิดีโอ
5. แก้ให้ preview กลับมาเล่นได้ก่อน
6. ยังไม่ต้อง render MP4

```

■ เมื่อข้อความไทยสั้นจ่อ

```

prompt · ข้อความสั้นจ่อ

> จาก preview ล่าสุด ข้อความภาษาไทย
สั้นจ่อหรืออ่านยาก

ช่วยแก้:
- ตัดประโยคให้สั้นลง
- แบ่งบรรทัดให้เหมาะกับมือถือ
- เพิ่มระยะขอบซ้ายขวา
- ตรวจสอบขนาดฟอนต์
- ตรวจสอบทุก scene ที่มีข้อความไทย

```

ถ้า Claude แก้แล้วยังยาว ให้สั่งลดคำ ไม่ใช่ลดฟอนต์อย่างเดียว

■ เมื่อ animation หรือ transition ทำให้ฟัง

```
prompt · effect ทำให้ฟัง
```

> หลังเพิ่ม animation/transition แล้ว preview มีปัญหา

ช่วยตรวจว่า animation หรือ component ไหนเป็นต้นเหตุ

ให้คุณ:

1. ระบุส่วนที่น่าจะทำให้ฟัง
2. ลดความซับซ้อนก่อน
3. ถ้ายังฟัง ให้ถอด component ออก
4. คืน preview ให้เล่นได้ก่อน
5. ยังไม่ต้อง render MP4

ความเสถียรสำคัญกว่าลูกเล่น

■ เมื่อเสียงไม่ตรงกับภาพ

```
prompt · เสียงไม่ตรง
```

> เสียงบรรยายใน preview ยังไม่ตรงกับภาพ

ช่วยแก้:

1. ตรวจสอบความยาวไฟล์เสียง
2. ตรวจสอบเวลาเริ่มของเสียง
3. ตรวจสอบเสียงโดนตัดท้ายไหม
4. ถ้าเสียงยาวเกิน ให้ย่อสคริปต์ก่อน
5. ถ้าเสียงสั้นไป ให้ปรับจังหวะ scene
6. อัปเดต preview ให้ตรวจใหม่

■ เมื่อ Gemini TTS ใช้ไม่ได้

```

prompt · Gemini TTS error

> Gemini TTS ยังสร้างเสียงไม่ได้

ช่วยแก้แบบปลอดภัย:
1. อ่าน error ล่าสุด
2. ตรวจสอบ official docs ล่าสุดของ Gemini TTS
3. ตรวจสอบชื่อ model ที่ใช้ตรง official docs ใหม่
4. ตรวจสอบ API key หรือ environment variable ที่ต้องใช้
5. คัมแสดง API key กลับมาในคำตอบ
6. ถ้าขาด setup ให้บอกทีละขั้นสั้นๆ
7. ถ้าแก้เองได้ ให้แก้และทดสอบสร้างไฟล์เสียงใหม่

```

■ เมื่อ final render MP4 ไม่ผ่าน

```

prompt · render fail

> preview เล่นได้แล้ว แต่ final render MP4 ไม่สำเร็จ

ช่วยแก้เฉพาะปัญหาคับ render:
1. อ่าน error ตอน render
2. ตรวจสอบปัญหาจาก asset, audio, codec, path, หรือ dependency
3. อย่าเปลี่ยนดีไซม์วิดีโอ ถ้าไม่จำเป็น
4. แก้ให้ render ผ่าน
5. ตรวจสอบว่า MP4 เปิดได้
6. บอก path ไฟล์ final

```

สำคัญ: ถ้า preview ดีแล้ว อย่าให้ Claude ปรับดีไซม์ใหม่บ่อยๆ แต่แก้ render ให้ผ่าน

■ เมื่อ Claude แก้วนไปมา

บางครั้ง Claude อาจแก้หลายรอบแล้วยังไม่จบ ให้บังคับให้สรุปสถานะก่อน

```
prompt · บังคับให้สรุปสถานะ

> หยุดแก้ก่อน แล้วช่วยสรุปสถานะปัจจุบัน
แบบสั้นๆ

ตอบเป็นหัวข้อ:
1. ตอนนี้จะไรทำงานแล้ว
2. อะไรยังพัง
3. error ล่าสุดคืออะไร
4. สาเหตุที่เป็นไปได้ 1-3 ข้อ
5. แผนแก้ที่สั้นที่สุดคืออะไร
6. จะเริ่มแก้จากไฟล์หรือจุดไหน

หลังสรุปแล้ว รอฉันยืนยันก่อนค่อยแก้ต่อ
```

■ เมื่ออยากย้อนกลับเวอร์ชันก่อน

```
prompt · ย้อนเวอร์ชัน

> เวอร์ชันล่าสุดแย่ง ฉันอยากย้อนกลับ
ไปเวอร์ชันก่อนที่ preview ยังดูดี

ช่วย:
1. หาไฟล์/สถานะก่อนหน้าที่น่าจะดีที่สุด
2. สรุปว่าคุณจะย้อนอะไรกลับมาบ้าง
3. อ্যালไฟล์สำคัญถ้าไม่จำเป็น
4. คืน preview ให้เหมือนเวอร์ชันที่
เสถียรก่อนหน้า
```

■ Prompt ปิดงานแก้ปัญหา

หลังแก้เสร็จ ให้บันทึกปัญหาไว้ การบันทึก error ช่วยให้อบต๋อไปแก้เร็วขึ้น

```
prompt · บันทึก error

> ช่วยบันทึกปัญหาที่เพิ่งแก้ลง
PROJECT_BRIEF.md

ขอแบบสั้น:
- ปัญหาที่เจอ
- สาเหตุ
- วิธีแก้
- ไฟล์ที่เกี่ยวข้อง
- ถ้าเจออีกควรใช้ prompt อะไร
- สถานะล่าสุด preview และ MP4
```

■ สรุปลำดับแก้ error

- 1 preview ให้เล่นได้ก่อน
- 2 แก้จาก error ล่าสุด
- 3 แก้ทีละจุด
- 4 อย่า render MP4 ระหว่างลองผิดลองถูก
- 5 final render เฉพาะเมื่อ preview ผ่านแล้ว

KEY TAKEAWAY

อย่างกลัว error — error เป็นเรื่องปกติของ workflow ทุกแบบ สิ่งสำคัญคือสั่งให้ Claude แก้เป็นระบบ

→ [undo ไป](#)
 บทที่ 11 · Workflow สั้นที่สุดตั้งแต่เริ่มจนได้วิดีโอ

READ ON

บทที่ สิบเอ็ด • CHAPTER ELEVEN

11

Workflow สั้นที่สุด ตั้งแต่เริ่มจนได้วิดีโอ

บทนี้คือ **สูตรใช้งานจริง**แบบรวบรัด ถ้าคุณอ่านถึงเล่มแล้ว กลับมาเปิดบทนี้เวลาจะทำงานจริง

หลักใหญ่มี 5 ขั้นตอน: เปิด Claude → Claude อ่าน source → preview → เพิ่มเสียง → final render

✘ อย่าจำทุก prompt ในเล่ม

✘ อย่า render ทุกครั้งที่แก้

✘ อย่าข้าม preview ไป render เลย

■ 5 ขั้นตอนหลักของทั้งเล่ม

- 1 เปิด Claude Code ในโฟลเดอร์โปรเจกต์
- 2 ให้ Claude อ่าน official source ถ้าจำเป็น
- 3 สร้าง/แก้ด้วย HTML preview
- 4 เพิ่มเสียงและตรวจใน preview
- 5 พอใจแล้วค่อย final render MP4



สรุปทั้งเล่มในภาพเดียว · ใช้ prompt ภาษาไทย ให้ Claude ทำงานเทคนิค ใช้ preview จนพอใจ แล้วค่อย final render MP4

ทำไม 5 ขั้นนี้ถึงพอ

01

ลด CLI

ใช้ command line แคตตอนเปิด — ที่เหลือเป็น prompt ภาษาไทย

02

ตรวจ source

สลดการเดาของ AI — Claude อ่าน official ก่อนแก้

03

preview ก่อน

เร็วกว่า render หลายเท่า ลูปแก้ทันใจ

■ Checklist ก่อนเริ่ม

ก่อนสั่งท้าวิติโอ ตรวจสอบ 5 อย่างนี้

- เปิด Claude Code ได้ด้วย `claude`
- อยู่ในโฟลเดอร์โปรเจกต์ที่ถูกต้อง
- HyperFrames skills พร้อมใช้
- มี `PROJECT_BRIEF.md` หรือ context ของโปรเจกต์
- มีไอเดียวิดิโอคร่าวๆ

ถ้าไม่แน่ใจ ให้ใช้ prompt นี้

```
prompt · ตรวจสอบความพร้อม
```

```
> ช่วยตรวจสอบความพร้อมของโปรเจกต์นี้ก่อน  
เริ่มท้าวิติโอ
```

```
ตรวจสอบให้หน่อยว่า:
```

1. อยู่ในโฟลเดอร์โปรเจกต์ถูกต้องไหม
2. HyperFrames skills พร้อมใช้ไหม
3. มี `PROJECT_BRIEF.md` หรือ context เดี๋ยว
4. `preview workflow` พร้อมไหม
5. ถ้าขาดอะไร ให้แก้หรือบอกฉันสั้นที่สุด

```
ยังไม่ต้องสร้างวิดิโอ
```



ถ้าตรวจสอบแล้วครบ

ส่ง Master Prompt ในหน้าถัดไปได้เลย — ถ้าขาดอะไร ให้ Claude แก้เองก่อน อย่าเพิ่งสร้างวิดิโอจนกว่า checklist 5 ข้อจะผ่านครบ

■ Master Prompt สำหรับเริ่มวิดีโอใหม่

นี่คือ prompt หลักที่ควรใช้บ่อยที่สุด

```
prompt · Master

> ฉันต้องการให้คุณเป็นผู้ช่วยสร้างวิดีโอ
ด้วย HyperFrames ในโปรเจกต์นี้

กติกา:
- จับไม่รู้อีโหน่อีเหนอ
- ใช้ prompt ภาษาไทยเป็นหลัก
- ใช้ official docs/repo เป็นฐาน
- ใช้ HyperFrames skills ที่เหมาะสม
- ใช้ CLI เองเท่าที่จำเป็น
- ทำ HTML preview ก่อน
- ห้าม render MP4 จนกว่าฉันจะสั่ง
  "final render"
- ถ้ามี error ให้แก้เองก่อน

งานที่ต้องทำ:
สร้างวิดีโอ [แนวตั้ง/แนวนอน] ความยาว
[จำนวน] วินาที เรื่อง "[หัวข้อ]"

รายละเอียด:
- กลุ่มคนดู: [กลุ่ม]
- โทนภาพ: [โทน]
- ประเด็นหลัก: [ประเด็น]
- call to action: "[CTA]"

เริ่มจากสร้าง HTML preview ให้ฉันกด play
ดูก่อน
```

■ Prompt สำหรับแก้งานหลังดู preview

```
prompt · feedback

> หลังดู preview แล้ว ขอแก้จุดนี้:

1. [จุดที่ 1]
2. [จุดที่ 2]
3. [จุดที่ 3]

ให้คุณแก้ไขไฟล์ที่จำเป็น แล้วอัปเดต
HTML preview ให้จับคู่ใหม่
ยังไม่ต้อง render MP4
```

■ Prompt สำหรับเพิ่ม catalog

```
prompt · catalog

> ช่วยใช้ official HyperFrames catalog
เพื่อปรับวิดีโอ preview ล่าสุด

ฉันต้องการ:
- [เช่น caption อ่านง่ายขึ้น]
- [เช่น transition นุ่มขึ้น]

เลือกไม่เกิน 2-3 component/block
ใช้เฉพาะ official catalog
ตัดตั่งและใส่เอง
อัปเดต HTML preview ให้ดู
```

■ Prompt สำหรับเพิ่มเสียง Gemini TTS

```

prompt · TTS

> ช่วยเพิ่มเสียงบรรยายภาษาไทยให้วิดีโอ
preview ล่าสุดด้วย Gemini TTS

ให้ทำตามนี้:
1. อ่าน official Gemini TTS docs ล่าสุด
2. ตรวจสอบ model TTS ที่ใช้ได้จริง
3. ถ้าชื่อ model ที่จับคู่ไม่ตรง docs
   ให้บอกตรงๆ และเลือก official ที่เหมาะ
4. เขียนสคริปต์เสียงไทยให้พอดี
5. สร้างไฟล์เสียงโดยเก็บ API key
   อย่างปลอดภัย
6. ใส่เสียงในวิดีโอ และเปิด preview
   ให้ตรวจสอบ sync
7. ยังไม่ต้อง render จนกว่าจะสั่ง
   "final render พร้อมเสียง"

ห้ามแสดง API key กลับมาในคำตอบ
    
```

■ Prompt สำหรับ final render

```

prompt · final

> ตอนนี้ฉันพอใจกับ preview ล่าสุดแล้ว

ช่วย final render เป็น MP4
ตั้งชื่อไฟล์ไม่ทับกับของเดิม เช่น
output-final-v1.mp4
ถ้ามี error ตอน render ให้แจ้ง
ตรวจสอบว่า MP4 เปิดได้ ถ้ามีเสียงให้ตรวจสอบ
ว่าเสียงอยู่ในไฟล์จริง

ตอนนี้อนุญาตให้ render MP4 ได้
    
```

■ Workflow แบบ 10 นาที

ถ้าต้องทำเร็ว ใช้แบบนี้

- 1 เปิดโฟลเดอร์โปรเจกต์
- 2 พิมพ์ `cLaude`
- 3 ส่ง Master Prompt
- 4 ดู preview
- 5 ส่ง feedback 1-2 รอบ
- 6 เพิ่มเสียงถ้าจำเป็น
- 7 ดู preview พร้อมเสียง
- 8 สั่ง final render
- 9 เปิด MP4 ตรวจสอบ
- 10 บันทึกสรุปลง `PROJECT_BRIEF.md`



ถ้าเวลาไม่พอ

ตัดเสียงหรือ catalog ออกก่อน — ทำวีดิโอเรื่อยๆ ที่ชัด ดีกว่าวีดิโอแรกที่ render ไม่ทัน

ทางเลือก 3 อย่างที่ใช้อยู่

①

ข้าม catalog

ถ้าวีดิโอเรื่อยๆ ก็พอ ไม่ต้องเพิ่ม component — Claude ใช้ default ทำได้

②

ข้ามเสียง

ถ้าคลิป quote สั้น 8 วิ บางครั้งไม่ต้องมีเสียงบรรยายก็ทำงานได้

③

ลด feedback

รอบแรกพอดูได้แล้วข้าม iterate ไป final ได้เลย

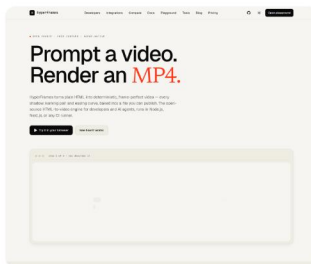
■ กฎ 7 ข้อของเล่มนี้

- 1 ใช้ prompt ภาษาไทยเป็นหลัก
- 2 ให้ Claude อ่าน official source ก่อนเมื่อต้องติดตั้งหรือใช้ model
- 3 ให้ Claude ทำงานเทคนิคเองให้มากที่สุด
- 4 preview ก่อน render เสมอ
- 5 render MP4 เฉพาะตอน final หรือ milestone
- 6 ถ้า error ให้ Claude อ่านและแก้เองก่อน
- 7 บันทึกสถานะลง `PROJECT_BRIEF.md` เพื่อความต่อเนื่อง

จบเล่ม

คุณไม่จำเป็นต้องจำทุก prompt ในเล่ม จำวิธีคิดนี้:

บอกผลลัพธ์ให้ชัด · ให้ Claude ทำงานเทคนิค · ดู preview จนพอใจ · ค่อย render MP4 final



HyperFrames เป็นเครื่องมือหลัก · คุณทำหน้าที่ส่งเป้าหมายและตรวจ preview ไม่ใช่เขียนระบบวิดีโอเอง

ภาคผนวก A · CHAPTER A



Prompt สั้นพร้อมใช้

ภาคผนวกนี้รวม **prompt สั้น** สำหรับ copy ไปใช้ทันที

แต่ละ prompt ทำหน้าที่ชัดเจน เปลี่ยนเฉพาะข้อความใน [...]

✘ ไม่ต้องจำ prompt ทุกตัว

✘ ไม่ต้องอ่านเรียงจากต้น

✘ ใช้เป็น reference เปิดเฉพาะตอนต้องการ

■ ตรวจสอบโปรเจกต์

```

prompt · ตรวจสอบพร้อม
> ช่วยตรวจสอบความพร้อมของโปรเจกต์นี้ก่อน
เริ่มงาน

ตรวจสอบว่าอยู่ในไฟเตอร์ถูกต้องไหม
มี HyperFrames skills พร้อมไหม
และ preview workflow พร้อมไหม
ถ้าขาดอะไร ให้แก้หรือบอกฉันสิ่งที่ขาด
ยังไม่ต้องสร้างวิดีโอ
    
```

■ อ่าน official HyperFrames repo

```

prompt · อ่าน repo
> ก่อนทำงาน HyperFrames ให้คุณอ่าน
official repo นี้ก่อน:
https://github.com/heygen-com/hyperframes

จากนั้นสรุปเฉพาะสิ่งที่จำเป็นต่อโปรเจกต์นี้
และใช้วิธี official เท่านั้น
    
```

■ ติดตั้ง HyperFrames skills

```
prompt · ติดตั้ง skills  
  
> ช่วยติดตั้ง HyperFrames skills ตามวิธี  
จาก official HyperFrames repo  
  
ให้คุณรับคำสั่งเองถ้าจำเป็น ใช้วิธี  
official เท่านั้น  
ถ้าติด error ให้แก้เองก่อน แล้วสรุปสั้นๆ
```

■ สร้างวิดีโอแบบ preview-first

```
prompt · สร้างวิดีโอ  
  
> ใช้ /hyperframes สร้างวิดีโอ  
[แนวตั้ง/แนวนอน] ความยาว [จำนวน] วินาที  
เรื่อง "[หัวข้อ]"  
  
กลุ่มคนดู: [กลุ่ม]  
โทนภาพ: [โทน]  
ประเด็นหลัก: [ประเด็น]  
CTA: "[CTA]"  
  
ให้สร้าง HTML preview ให้ฉันกด play  
ดูก่อน ยังไม่ต้อง render MP4 จนกว่าฉัน  
จะสั่ง "final render"
```

■ แก้จาก feedback

```
prompt · feedback  
  
> หลังดู preview แล้ว ขอแก้:  
1. [จุดที่ 1]  
2. [จุดที่ 2]  
3. [จุดที่ 3]  
  
แก้ไฟล์ที่จำเป็น แล้วอัปเดต HTML preview  
ให้ดูใหม่ ยังไม่ต้อง render MP4
```

■ ใช้ catalog

```
prompt · catalog  
  
> ช่วยเลือก block/component จาก official  
HyperFrames catalog ที่เหมาะกับวิดีโอนี้  
  
เป้าหมายคือ [caption อ่านง่ายขึ้น /  
transition นุ่มขึ้น / effect บางๆ]  
เลือกไม่เกิน 3 รายการ ใส่นิพจน์  
แล้วอัปเดต HTML preview  
ยังไม่ต้อง render MP4
```

■ เพิ่มเสียง Gemini TTS

```
prompt · Gemini TTS

> ช่วยเพิ่มเสียงบรรยายภาษาไทยด้วย
Gemini Flash TTS ผ่าน Google Cloud/gcloud

ก่อนทำงานให้อ่าน official docs ล่าสุดของ
Google Cloud Free Trial, gcloud CLI
และ Gemini-TTS

ให้คุณ:

1. พาจับตั้งค่า Google Cloud
   แบบมือใหม่ ถ้ายังไม่พร้อม
2. ตรวจสอบชื่อ Flash TTS model ที่
   official docs สองครั้งจริง
3. ใช้ gcloud/Application Default
   Credentials หรือวิธี official
   ที่ปลอดภัย
4. เขียนสคริปต์เสียงให้พอดีกับวิดีโอ
5. สร้างไฟล์เสียง ใ้ใน preview
   และตรวจ sync
6. ยังไม่ต้อง render MP4 จนกว่าฉัน
   จะสั่ง final render พร้อมเสียง

ห้ามแสดง token / credential / API key
ในคำตอบ
```

■ แก้ error

```
prompt · แก้ error

> งานนี้ยังไม่สำเร็จ ช่วยแก้ไขให้เอง
แบบเป็นระบบ

อ่าน error ล่าสุด ทลสาเหตุ แก้เฉพาะ
จุดที่จำเป็น แล้วทำให้ HTML preview
กลับมาเล่นได้ก่อน
ยังไม่ต้อง render MP4 จนกว่าฉันจะสั่ง
final render
ถ้าจำเป็นต้องให้ฉันทำอะไร ให้บอกทีละขั้น
แบบคนไม่รู้โค้ด
```

■ Final render

```
prompt · final

> ตอนนี้นัดพรีวิว preview ล่าสุดแล้ว

ช่วย final render เป็น MP4
ตั้งชื่อไฟล์ไม่ให้ทับของเดิม
ถ้ามี error ตอน render ให้แก้เอง
ตรวจว่า MP4 เปิดได้
แล้วบอก path ไฟล์ final

ตอนนี้อนุญาตให้ render MP4 ได้
```

■ บันทึกสถานะ

```
prompt · บันทึก
> ช่วยอัปเดต PROJECT_BRIEF.md

บันทึก:
- หัวข้อวิดีโอ
- style ที่ใช้
- component ที่เพิ่ม
- ไฟล์เสียง
- path preview
- path MP4 final
- ปัญหาที่เจอ
- สิ่งที่ต้องปรับต่อ
```

Prompt แต่ละตัวมาจากบทไหน

01-02

ตรวจ · อ่าน repo

ดูเพิ่มในบทที่ 3

03

ติดตั้ง skills

ดูเพิ่มในบทที่ 4

04

สร้างวิดีโอ

ดูเพิ่มในบทที่ 5

05-06

feedback · catalog

ดูเพิ่มในบทที่ 6, 7

07

Gemini TTS

ดูเพิ่มในบทที่ 8

08-09

error · final

ดูเพิ่มในบทที่ 10, 11

ภาคผนวก A · จบ

เปิดหน้านี้เป็น **reference** เวลาทำงานจริง – ไม่ต้องจำ ใช้แล้วก็ปิด

ภาคผนวก B · CHAPTER B

B

คำศัพท์ แบบภาษาคนทั่วไป

คำศัพท์เฉพาะที่ใช้ในเล่ม อธิบายแบบ **ภาษาคนทั่วไป** ไม่ต้องมีพื้นฐานเทคนิค

ทุกคำมาจาก official source หรือใช้จริงใน workflow ของเล่ม — เปิดดูตอนสงสัยได้เลย

■ เครื่องมือหลักในเล่ม

Claude Code	AI assistant ที่ทำงานในโฟลเดอร์โปรเจกต์ได้ เช่น สร้างไฟล์ แก้ไฟล์ รันคำสั่ง ตรวจสอบ error และช่วยจัดการ workflow
HyperFrames	เครื่องมือสร้างวิดีโอจาก HTML/CSS/JS แล้ว preview ใน browser หรือ render เป็น MP4 — ในเล่มนี้คุณไม่ต้องเขียน HTML เอง ให้ Claude เขียนให้
CLI	ช่องพิมพ์คำสั่งในคอมพิวเตอร์ เช่น Terminal หรือ PowerShell — ในเล่มนี้ใช้ให้น้อยที่สุด หลักๆ ใช้ติดตั้งและเปิด Claude Code
Prompt	ข้อความที่คุณส่ง AI — prompt ที่ดีควรบอกผลลัพธ์ที่ต้องการ เจาะจง และให้ AI รับผิดชอบขั้นตอนเทคนิค
Skill	ชุดคำแนะนำเฉพาะทางที่ทำให้ Claude Code ทำงานบางอย่างได้ถูกวิธีขึ้น เช่น HyperFrames skills

■ ขั้นตอนทำวิดีโอ

Preview	การเปิดดูวิดีโอใน browser ก่อน render เป็น MP4 — เรียกว่า render มาก เป็นวงจรถัดไปตอนแรก
Render	การแปลงงานเป็นไฟล์วิดีโอ MP4 จริง — ควรทำตอน final หรือ milestone ไม่ใช่ทุกครั้งที่แก้
Final render	การ render MP4 ตอนท้าย หลังดู preview จนพอใจแล้ว — สั่ง Claude ชัดๆ ว่า "ตอนนี้อนุญาตให้ render MP4 ได้"
MP4	ไฟล์วิดีโอที่เอาไปส่งงาน อัปโหลด หรือแชร์ได้
Caption	ข้อความบนจอหรือคำบรรยายในวิดีโอ — วิดีโอสั้น caption ต้องใหญ่ สั้น และอ่านกัน
Transition	จังหวะเปลี่ยนจาก scene หนึ่งไปอีก scene หนึ่ง — ควรใช้พอดี้ ไม่ให้กลบเนื้อหา

■ Catalog และเสียง

Catalog	คลัง block/component สำเร็จรูปของ HyperFrames เช่น caption, transition, chart, effect
Component / Block	ของสำเร็จรูปที่เอามาใส่ในวิดีโอได้ เช่น caption style หรือ transition
TTS	Text-to-speech แปลว่าการแปลงข้อความเป็นเสียงพูด — ในเล่มนี้ใช้ Gemini TTS โดยให้ Claude ตรวจสอบ official docs ก่อนเลือก model
gcloud	เครื่องมือ command line ของ Google Cloud ใช้ login เลือก project ตั้งค่า authentication และเรียกบริการ Google Cloud
Google Cloud Free Trial	โปรแกรมทดลองใช้ Google Cloud สำหรับผู้ใช้ใหม่ที่เข้าเงื่อนไข มีเครดิตฟรีตามเงื่อนไข official — ต้องใส่บัตรเครดิตเพื่อยืนยันตัวตน และคอยดู billing

■ Security & Misc

API key	กุญแจสำหรับเรียกใช้บริการ API เช่น Gemini TTS — ห้ามแชร์สาธารณะ ห้ามให้ Claude แสดง key กลับมาในคำตอบ
.env	ไฟล์สำหรับเก็บค่าลับหรือค่าตั้งค่า เช่น API key — อย่าแฮร์ไฟล์นี้ถ้ามี key จริงอยู่ข้างใน
Dependency	ของที่โปรเจกต์ต้องใช้เพื่อทำงาน เช่น package, tool หรือ library — ถ้าขาด ให้ Claude ติดตั้งหรือบอกขั้นตอนแบบสั้นๆ
Error	ข้อความแจ้งปัญหา — ไม่ต้องกลัว ให้ Claude อ่าน error และแก้เป็นระบบ
PROJECT_BRIEF.md	ไฟล์บันทึกเป้าหมายโปรเจกต์ ให้ Claude อ่านทุก session — บันทึก style วิดีโอ, model TTS ที่ใช้, path preview/MP4 final

END OF BOOK

ขอบคุณที่อ่านจนจบ — ลองเปิด Claude Code แล้วเริ่มทำวิดีโอแรกของคุณวันนี้