

WRITER2RESEARCH

สอน AI ให้ทำงานแบบเรา

คู่มือใช้งาน Skill ฉบับลงมือทำ
ค้นหา · ติดตั้ง · เขียนเอง

ครูหลิน

สอน AI ให้ทำงานแบบเรา

คู่มือใช้งาน Skill ฉบับลงมือทำ

ครูหลิน · Writer2Research

พิมพ์ครั้งที่ 1 · กันยายน 2569

writer2research.com · LINE @writer2research

เนื้อหาในเล่มอ้างอิงเอกสารทางการที่รวบรวมไว้ในภาคผนวก จ
ตัวละครในเรื่องเล่าทุกตัวเป็นตัวละครสมมติ

คำนำ

ครูเริ่มเขียนเล่มนี้เพราะคำถามเดียวที่ได้ยินซ้ำในห้องอบรมทุกครั้ง

"ทำไม AI มันจำไม่ได้คะ"

คนถามมักเป็นพยาบาลที่ใช้ AI มาสองสามเดือนแล้ว เก่งพอที่จะสั่งงานได้ผลดี แต่ยังต้องพิมพ์คำสั่งยาวสามย่อหน้าใหม่ทุกเช้า พอวันไหนพิมพ์ไม่ครบ ผลลัพธ์ก็เพี้ยน พอเปลี่ยนคนทำ มาตรฐานก็เปลี่ยนตาม

คำตอบสั้นมาก AI จำไม่ได้เพราะเราไม่เคยเขียนไว้ที่ไหนเลย เราเก็บวิธีทำงานไว้ในหัว แล้วเล่าให้มันฟังใหม่ทุกวัน

Skill คือคำตอบยาว

มันไม่ใช่ของใหม่ที่ต้องเรียนตั้งแต่ต้น ไม่ใช่โปรแกรมที่ต้องเขียนโค้ดเป็น มันคือไฟล์เดอร์หนึ่งชุดที่มีไฟล์ข้อความอยู่ข้างใน บอก AI ว่างานแบบนี้ทำอย่างไร เขียนครั้งเดียว ใช้ได้ทุกวัน ส่งต่อให้ทีมได้ ย้ายไปใช้กับ AI ค่ายอื่นได้

เล่มนี้เขียนสำหรับคนที่ไม่เขียนโค้ด ทุกบทลงมือทำได้จริง มีเรื่องเล่าจากหน้างานประกอบ มีแบบฝึกหัดท้ายบทที่ทำเสร็จภายในยี่สิบนาที และมีแม่แบบพร้อมใช้แปดชุดอยู่ท้ายเล่ม

ครูตั้งใจไว้ข้อเดียว อ่านจบแล้วต้องมี Skill ของตัวเองอย่างน้อยหนึ่งตัวที่ใช้ได้จริง ไม่ใช่หนึ่งตัวที่สวยแต่ไม่เคยถูกเรียกใช้

ขอให้สนุกกับการสอน AI ให้ทำงานแบบเรานะคะ

ครูหลิน

กันยายน 2569

วิธีใช้หนังสือเล่มนี้

เล่มนี้แบ่งเป็นหกภาค อ่านเรียงจากต้นจนจบได้ และหยิบอ่านเฉพาะภาคที่ต้องใช้ก็ได้ค่ะ

หกภาคของเล่มนี้

ภาค	บท	เนื้อหา	เหมาะกับใคร
1	1-4	เข้าใจว่า Skill คืออะไรและทำงานอย่างไร	ทุกคน อ่านก่อนเสมอ
2	5-8	ภูมิทัศน์ถึงวงการ Claude ChatGPT Gemini	คนที่ใช้ AI หลายค่าย
3	9-12	ค้นหา ทัดตั้ง และตรวจสอบความปลอดภัย	คนที่อยากใช้ของที่มีคนทำไว้แล้ว
4	13-17	เขียน Skill ของตัวเอง	คนที่พร้อมลงมือ อ่านให้ครบทุกบท
5	18-21	ใช้จริงในงานวิจัย สถิติ คลินิก และงานสอน	พยาบาลและนักวิจัย
6	22-24	ขยายผลสู่ทีมและวางแผนต่อ	หัวหน้าหน่วยงานและคนที่จะสอนต่อ

ถ้ามีเวลาน้อย

อ่านสื่บทนี้ก่อน แล้วค่อยกลับมาเก็บที่เหลือ

- บทที่ 2 รู้ว่า Skill คืออะไร

- บทที่ 14 เขียน description ให้ AI หยิบถูกจังหวะ ซึ่งเป็นบทที่สำคัญที่สุดของเล่ม
- บทที่ 15 เขียนเนื้อคู่มือ
- ภาคผนวก ก คัดลอกแม่แบบไปแก้ไขได้ทันที

สัญลักษณ์ที่ใช้ในเล่ม

กล่องสีแบบ

กล่อง	ความหมาย
ควรรู้	ข้อมูลเสริมที่ช่วยให้เข้าใจลึกขึ้น ข้ามได้ถ้ารีบ
ระวัง	ข้อผิดพลาดที่คนทำบ่อย หรือความเสี่ยงที่ต้องรู้ก่อน
เรื่องจากหน้างาน	เรื่องเล่าตัวอย่าง ใช้ตัวละครสมมติ
ลงมือทำ	แบบฝึกหัดท้ายบท ทำเสร็จภายในสัปดาห์สัปดาห์

ทุกบทปิดด้วยกล่อง "เก็บกลับบ้าน" สรุปสามถึงห้าข้อ และกล่อง "ลงมือทำ" สามข้อ ถ้าทำครบทุกบท จะได้ Skill ของตัวเองอย่างน้อยหนึ่งตัวโดยไม่ต้องนั่งเขียนรวดเดียว

สิ่งที่ต้องเตรียม

- เครื่องที่เปิด AI ได้หนึ่งเครื่อง จะเป็นค่ายไหนก็ได้
- โปรแกรมพิมพ์ข้อความธรรมดาที่มีอยู่แล้วในเครื่อง เช่น Notepad หรือ TextEdit
- งานที่ทำซ้ำบ่อยที่สุดของตัวเองหนึ่งงาน ไว้ใช้เป็นโจทย์ทดลองเล่ม
- โฟลเดอร์ว่างหนึ่งโฟลเดอร์ ตั้งชื่อว่า **my-skills** ไว้เก็บของที่เขียนเอง

ข้อมูล ณ กันยายน 2569

รายละเอียดของแต่ละค่ายเปลี่ยนบ่อย ตัวเลขและคำสิ่งทั้งหมดในเล่มมาจากเอกสารทางการที่รวบรวมไว้ในภาคผนวก จ ก่อนนำไปอ้างอิงในงานวิชาการ ให้เปิดตรวจกับต้นทางอีกครั้งนะคะ

สารบัญ

คำนำ	3
วิธีใช้หนังสือเล่มนี้	4
ภาค หนึ่ง เข้าใจ Skill	9
1 คำสั่งที่เราพิมพ์ซ้ำทุกเช้า	10
2 Skill คือโฟลเดอร์ ไม่ใช่เวทมนตร์	18
3 กายวิภาคของ SKILL.md	26
4 AI ไม่ได้อ่านทั้งหมด แต่อ่านเป็นชั้น	36
ภาค สอง ภูมิทัศน์ทั้งวงการ	45
5 Skill กลายเป็นมาตรฐานกลางของทั้งวงการ	46
6 Skill บน Claude ทุกช่องทาง	55
7 ฝั่ง ChatGPT: GPTs, Projects และ Codex	63
8 ฝั่ง Gemini: Gems และ NotebookLM	71
ภาค สาม ค้นหาและติดตั้ง	79
9 ห้าที่ที่ใช้ค้นหา Skill	80
10 อ่านหน้ารายละเอียดให้เป็นก่อนกดติดตั้ง	90
11 ติดตั้ง จัดการ และถอนให้เป็น	98
12 ติดตั้ง Skill เท่ากับติดตั้งซอฟต์แวร์	109
ภาค สี่ เขียน Skill ของตัวเอง	118

13	เลือกงานที่คํมจะกํเป็น Skill	119
14	เขียน description ให้ AI หยิบถูกจังหวะ	128
15	เขียนเนื้อคํมือให้ AI กํตามได้	138
16	ไฟล์ประกอบ: references, assets และ scripts	149
17	ทดสอบ แก่ และดูแลให้อยู่ยาว	158
ภาค ห้า ใช้จริงในงานสุขภาพและวิจัย		168
18	Skill สำหรับงานเขียนวิทยานิพนธ์และบทความ	169
19	Skill สำหรับงานสถิติและการรายงานผล	178
20	Skill สำหรับบททวนวรรณกรรมและจริยธรรมการวิจัย	187
21	Skill สำหรับงานคลินิกและงานสอน	196
ภาค หก ขยายผล		205
22	Skill กับ MCP กับ Plugin ประกอบร่างเป็นระบบงาน	206
23	ทำให้ทั้งทีมใช้ Skill ชุดเดียวกัน	216
24	เก้าสิบวันข้างหน้า	225
ภาคผนวก ก — แม่แบบ SKILL.md แปรชุดพร้อมใช้		234
ภาคผนวก ข — คลังคำสั่งและปุ่มที่ต้องรู้		251
ภาคผนวก ค — เช็กลิสต์ที่พิมพ์ออกมาปะข้างจอได้		259
ภาคผนวก ง — อภิธานศัพท์		264
ภาคผนวก จ — แหล่งอ้างอิงและที่ไปต่อ		268

ภาค หนึ่ง

เข้าใจ Skill

บทที่ 1-4

Skill ไม่ใช่ของลึกลับ มันคือโพลเดอร์ที่มีไฟล์ข้อความอยู่ข้างใน
ภาคนี้ปูพื้นให้ครบก่อนลงมือ

- 1 คำสั่งที่เราพิมพ์ซ้ำๆทุกเช้า
- 2 Skill คือโพลเดอร์ ไม่ใช่เวทมนตร์
- 3 กายวิภาคของ SKILL.md
- 4 AI ไม่ได้อ่านทั้งหมด แต่อ่านเป็นชั้น

บทที่ 1

คำสั่งที่เราพิมพ์ซ้ำทุกเช้า

คำสั่งยาวสามย่อหน้าที่คุณพิมพ์เมื่อวาน วันนี้หายไปแล้ว พรุ่งนี้ต้องพิมพ์ใหม่อีกรอบ ปัญหาไม่ได้อยู่ที่ AI และไม่ได้อยู่ที่คุณ

เช้าวันจันทร์ที่หน้าจอลืม

เจ็ดโมงครึ่ง กาแฟยังร้อน หน้าจอสว่างขึ้น ช่องแชทว่างเปล่า

คุณเริ่มพิมพ์ "ช่วยสรุปบทความวิจัยฉบับนี้ เขียนภาษาไทยทั้งทางการ แยกหัวข้อวัตถุประสงค์ วิธีการ ผลการวิจัย ข้อจำกัด ใช้คำว่ากลุ่มตัวอย่าง ไม่ใช่คำว่าผู้เข้าร่วม ตัวเลขทศนิยมสองตำแหน่ง อ้างอิงแบบ APA 7th ห้ามแต่งตัวเลขที่ไม่มีในต้นฉบับ"

พิมพ์เสร็จ อ่านทวน เติมอีกสองบรรทัดที่ลืม แล้วกด Enter

ผลออกมาดี คุณพอใจ ปิดเครื่อง กลับบ้าน

วันพุธคุณเปิดแชทใหม่ แล้วต้องพิมพ์ทั้งหมดนั้นอีกครั้ง คราวนี้จำได้ไม่หมด ลืมบรรทัดเรื่องทศนิยม ผลที่ได้เลยเป็นทศนิยมสามตำแหน่งบ้าง สี่ตำแหน่งบ้าง คุณพิมพ์แก้ตามหลัง เสียเวลาไปอีกเจ็ดนาที

เจ็ดนาทีต่อครั้ง สัปดาห์ละสี่ครั้ง คุณสับสนสัปดาห์ ลองคิดดูค่ะ

สื่ออาการที่เจอเหมือนกันทั้งประเทศ

ครูสอนเรื่องนี้มาหลายรุ่น ทั้งพยาบาล อาจารย์ นักศึกษาปริญญาโท และคนทำงานวิจัยในโรงพยาบาล อาการที่เล่าให้ฟังแทบไม่ต่างกันเลย มีอยู่สี่แบบ

หนึ่ง พิมพ์คำสั่งยาวซ้ำทุกวัน

คำสั่งดี ๆ ไม่ได้เกิดในครั้งเดียว คุณลองแล้วลองอีก ตัดคำนี้ เติมคำนั้น จนได้ชุดคำที่ใช้ได้ผล แล้วชุดคำนั้นก็จมหายไปกับแชทที่ปิดไป ครั้งหน้าคุณเริ่มจากศูนย์ใหม่

สอง ผลลัพธ์ไม่คงที่

วันนี้ได้สรุปหำย่อหน้าเรียบร้อย พรุ่งนี้ได้บทความยาวสองหน้าพร้อมหัวข้อที่ไม่ได้ขอ ไม่ใช่เพราะ AI อารมณ์ไม่ดี แต่เพราะคำสั่งที่คุณพิมพ์สองวันนั้นไม่เหมือนกัน คุณจำได้ไม่ครบ มันจึงเดาส่วนที่ขาดให้เอง

ที่น่าเจ็บใจคือส่วนที่คุณลืมมักเป็นส่วนที่คุณคิดนานที่สุด เพราะมันเป็นข้อยกเว้นเล็ก ๆ ที่ใช้เวลากว่าที่จะค้นพบ พอลืม ก็ต้องค้นพบใหม่

สาม ต้องอธิบายมาตรฐานหน่วยงานใหม่ทุกรอบ

หน่วยงานคุณมีข้อตกลงของตัวเอง ตารางต้องมี n และร้อยละในช่องเดียวกัน หัวข้อบทที่ 4 ต้องเรียงตามวัตถุประสงค์ ชื่อโรงพยาบาลต้องเขียนเต็มครั้งแรก แล้วย่อ ข้อตกลงพวกนี้ไม่มีอยู่ในหัว AI มันรู้เพราะคุณบอก และมันลืมทันทีที่ปิดแชท

สี่ ส่งต่อวิธีทำงานให้ทีมไม่ได้

น้องในทีมถามว่า "พี่ทำยังไงถึงได้ผลแบบนั้น" คุณตอบว่า "เดี๋ยวส่งคำสั่งให้" แล้วก็ไล่หาในแชทเก่า หาไม่เจอ สุดท้ายพิมพ์ใหม่ให้แบบย่อ ๆ น้องเอาไปใช้ ได้ผลไม่เหมือนของคุณ

หน่วยงานหนึ่งจึงมีวิธีทำงานอยู่หลายเวอร์ชันพร้อมกัน ของพี่คนหนึ่งอย่างหนึ่งของน้องอีกคนอีกอย่าง ไม่มีใครผิด แต่รายงานที่ส่งขึ้นไปหน้าตาไม่เหมือนกันสักฉบับ



ถูกรอบเริ่มจากศูนย์ เพราะไม่มีที่ไหนเก็บวิธีทำงานไว้เลย

ภาพที่ 1 วงจรพิมพ์ซ้ำที่ไม่มีวันจบ

เรื่องจากหน้างาน

พี่สนใจเป็นพยาบาลวิชาชีพ ดูแลงานคุณภาพของหอผู้ป่วยใน ทุกสิ้นเดือน ต้องเขียนสรุปอุบัติการณ์ส่งคณะกรรมการ เธอใช้ AI ช่วยเรียบเรียงมาปึกว่าแล้ว

เดือนแรกเธอใช้เวลาทั้งบ่ายกว่าจะได้คำสั่งที่ถูกต้อง เดือนถัดมาเธอเปิดแชทเก่าไปท้อปมาวาง เดือนที่สามแชทเก่าหาไม่เจอ เธอพิมพ์ใหม่จากความจำ

ปีนี้พี่สนใจย้ายไปหอผู้ป่วยอื่น คนที่มารับงานต่อได้ไฟล์รายงานเก่าไปหมด แต่ไม่ได้วิธีทำงานสักบรรทัดเดียว

ต้นเหตุร่วมมีข้อเดียว

สื่อการช่างบนดูเหมือนคนละเรื่อง จริง ๆ มันคือเรื่องเดียวกัน

เราเก็บ "วิธีทำงาน" ไว้ในบทสนทนา

บทสนทนาเป็นของชั่วคราวโดยธรรมชาติ มันเกิดขึ้นเพื่อจบงานตรงหน้า แล้วก็จบไป เหมือนคุยกันในลิฟต์ ตอนคุยเข้าใจตรงกันดี ออกจากลิฟต์แล้วไม่เหลืออะไร

ลองแยกให้ชัด งานที่คุณทำกับ AI มีสองก้อนเสมอ ก้อนแรกคือ **เนื้อหา** ไฟล์บทความ ตัวเลขจากแบบสอบถาม ร่างที่เขียนค้างไว้ ก้อนนี้เปลี่ยนทุกครั้ง ก้อนที่สองคือ **วิธีการ** รูปแบบที่ต้องการ ศัพท์ที่ใช้ ข้อห้าม ลำดับขั้นตอน ก้อนนี้แทบไม่เปลี่ยนเลย

เราเอาสองก้อนนี้ยัดลงช่องเดียวกันทุกครั้ง แล้วโยนทิ้งพร้อมกันทุกครั้ง

เนื้อหาเปลี่ยนทุกวัน วิธีการแทบไม่เปลี่ยนเลย แต่เราทิ้งมันพร้อมกันทุกครั้ง

สิ่งที่เก็บวิธีการ

ถ้าวิธีการคือของที่ควรเก็บ คำถามต่อไปคือเก็บไว้ที่ไหน คนทำงานส่วนใหญ่เลือกอยู่สี่ที่ ลองดูทีละแบบ

แบบที่หนึ่ง จำเอาเอง

ไม่ต้องเตรียมอะไร ใช้ได้ทันที ข้อเสียคือความจำมนุษย์มีขีดจำกัด คุณจำคำสั่งสิบบรรทัดได้ครบวันนี้ อีกสามสัปดาห์เหลือหกบรรทัด และคุณส่งความจำให้คนอื่นไม่ได้

แบบที่สอง โน้ตส่วนตัว

ดีขึ้นมาก คุณจดไว้ในสมุด ในแอปโน้ต หรือในไฟล์เวิร์ด เปิดดูได้ทุกเมื่อ ไม่ลืม ข้อเสียคือคุณต้องเป็นคนหยิบมาเองทุกครั้ง เปิดไฟล์ ลากคลุม ก๊อปปี้ สลับหน้าจอ วาง แล้วค่อยพิมพ์เนื้อหาต่อ ห้าขั้นตอนก่อนเริ่มทำงานจริง

แบบที่สาม prompt ยาว ๆ ที่ถือปวง

นี่คือที่คนเก่งส่วนใหญ่หยุดอยู่ คุณมีคลัง prompt ของตัวเอง prompt คือคำสั่งที่เราพิมพ์ให้ AI ทำงาน คุณเก็บไว้เป็นชุด ๆ ตั้งชื่อไว้ด้วย ใช้แล้วได้ผลดี

ข้อเสียเริ่มโผล่ตอนคลังใหญ่ขึ้น พอมีสามสิบชุด คุณจำไม่ได้ว่าชุดไหนใช้กับงานไหน หาชุดที่ถูกนานกว่าพิมพ์ใหม่ และเมื่อแก่ชุดหนึ่ง เพื่อนที่ถือไปเมื่อเดือนก่อนก็ยังใช้ของเก่าอยู่ดี

แบบที่สี่ ไฟล์คู่มือที่ AI เปิดอ่านเองได้

แบบนี้ต่างออกไปตรงคนหยิบ

คุณเขียนวิธีทำงานลงไฟล์ วางไว้ในที่ที่ AI มองเห็น เขียนกำกับไว้ด้วยว่าไฟล์นี้ใช้ตอนไหน จากนั้นเมื่อคุณสั่งงานที่ตรงกับที่เขียนไว้ AI เปิดไฟล์นั้นอ่านเอง แล้วทำตาม โดยคุณไม่ต้องถือปอะไรเลย

คุณพิมพ์ "สรุปบทความนี้ตามรูปแบบหน่วยงาน" แปะคำ ที่เหลืออยู่ในไฟล์

ความต่างระหว่างแบบที่สามกับแบบที่สี่จึงไม่ใช่เนื้อหา เนื้อหาแทบเหมือนกัน ต่างกันที่ใครเป็นคนเดินไปหยิบคู่มือมาเปิด トラバิดที่คนหยิบคือคุณ คุณก็ยังลืมได้ ยังหาไม่เจอได้ ยังส่งเวอร์ชันผิดให้เพื่อนได้

เปรียบเทียบที่เก็บวิธีทำงานสี่แบบ

ที่เก็บ	อยู่ได้นานไหม	ส่งต่อทีมได้ไหม	ใครเป็นคนหยิบ
จำเอาเอง	ไม่กี่สัปดาห์	ไม่ได้	เราเอง และจำได้ไม่ครบ
โน้ตส่วนตัว	นาน	ส่งไฟล์ให้ได้	เราเอง ทุกครั้ง
prompt ที่ถือปวง	นาน	ส่งข้อความให้ได้	เราเอง และเวอร์ชันแตกต่างกัน
ไฟล์คู่มือที่ AI อ่านเอง	นาน แท้ทีเดียวจบ	แชร์ทั้งทีมได้	AI หยิบเอง

จำเอาเอง		หายเมื่อลืม
โน้ตส่วนตัว		ต้องเปิดหาเอง
prompt ยาวที่ถือปวง		ต้องวางใหม่ทุกครั้ง
ไฟล์คู่มือที่ AI เปิดเอง		อยู่ตลอด ส่งต่อได้

ยิ่งแก่งยาว วิธีทำงานยังอยู่กัน และยิ่งส่งต่อให้คนอื่นได้

ภาพที่ 2 สี่ที่เก็บวิธีทำงาน เทียบความทนทาน

ย้ายวิธีทำงานออกจากแชท

ทางออกของทั้งสี่อาการจึงเป็นเรื่องเดียว ย้ายวิธีทำงานออกจากบทสนทนา ไปไว้ในไฟล์ที่อยู่กับที่

คำว่าอยู่กับที่สำคัญกว่าที่คิด แชทเป็นสายน้ำ ไหลไปแล้วไหลเลย ไฟล์เป็นก้อนหิน คุณวางไว้ตรงไหน พรุ่งนี้มันยังอยู่ตรงนั้น ปีหน้าก็ยังอยู่ และถ้าคุณย้ายที่ทำงาน มันย้ายตามคุณไปได้

พอวิธีทำงานอยู่ในไฟล์ อาการที่หนึ่งหาย เพราะไม่ต้องพิมพ์ซ้ำ อาการที่สองหาย เพราะทุกครั้งอ่านจากต้นฉบับเดียวกัน ผลจึงคงที่ อาการที่สามหาย เพราะ

มาตรฐานหน่วยงานเขียนไว้แล้ว อาการที่สับสน เพราะไฟล์ส่งต่อได้ แก่ที่เดียวทุกคนได้ของใหม่พร้อมกัน

ของที่กำหนดที่นี่มีชื่อเรียกว่า **Skill** ชื่อเต็มคือ Agent Skills ตอนนีขอให้อ่านไว้แค่ ว่ามันคือที่เก็บวิธีการทำงานแบบที่สื่ รายละเอียดว่าหน้าตาเป็นอย่างไร ข้างในมีอะไรเก็บไว้คุยกันตอนเปิดโพลเดอร์ดูจริง ๆ

ทำไมต้องเป็นไฟล์ ไม่ใช่ช่องตั้งค่าในโปรแกรม

ไฟล์เป็นของที่คุณเป็นเจ้าของ ก๊อปปี้ได้ ส่งอีเมลได้ ใส่โดรฟ์ให้ทีมได้ เปิดอ่านด้วยตาเปล่าได้ และแก้ไขได้โดยไม่ต้องรอใครมาเปิดสิทธิ์ให้

อีกข้อที่สำคัญกว่านั้น รูปแบบไฟล์แบบนี้เป็นมาตรฐานเปิด เว็บกลางคือ agentskills.io เครื่องมือหลายค่ายประกาศรองรับ ไฟล์ที่คุณเขียนวันนี้จึงไม่ผูกติดกับโปรแกรมเดียวไปตลอด

อย่าเพิ่งไปไล่ลบคลัง prompt เก่า

คลัง prompt ที่คุณสะสมมาไม่ใช่ของเสียเปล่า มันคือวัตถุดิบชั้นดี คำสั่งที่ผ่านการลองผิดลองถูกมาแล้วคือสิ่งที่ควรเอาไปเขียนลงไฟล์เป็นอย่างแรก

สิ่งที่เปลี่ยนคือที่อยู่ของมัน ไม่ใช่เนื้อหาของมัน เก็บไฟล์เก่าไว้ก่อน จนกว่าคุณจะไม่ใจว่าย้ายครบแล้ว

ก่อนพลิกไปหน้าถัดไป

งานที่คุณทำเก่งอยู่แล้ว มีวิธีของมันอยู่ วิธีนั้นอยู่ในหัวคุณ อยู่ในคำสั่งที่คุณพิมพ์ซ้ำมาทั้งปี มันมีตัวตนจริง เพียงแต่ยังไม่เคยถูกเขียนลงที่ไหน

ขั้นแรกไม่ใช่การเรียนรู้เขียนโค้ด ไม่ใช่การหาเครื่องมือใหม่ ขั้นแรกคือเปิดโพลเดอร์หนึ่งขึ้นมาดู แล้วจะเห็นว่าข้างในไม่มีอะไรน่ากลัวเลย

เก็บกลับบ้าน

- งานกับ AI มีสองก้อนเสมอ เนื่องจากที่เปลี่ยนทุกวัน กับวิธีทำงานที่แทบไม่เปลี่ยน
- อาการพิมพ์ซ้ำ ผลไม่คงที่ อธิบายใหม่ถูกรอบ และส่งต่อทีมไม่ได้ มาจากต้นเหตุเดียว คือเก็บวิธีทำงานไว้ในแชท
- ที่เก็บวิธีทำงานมีสี่แบบ ต่างกันที่ความทนทานและใครเป็นคนหยิบมาใช้
- แบบที่ทนที่สุดคือไฟล์คู่มือที่ AI เปิดอ่านเอง เพราะคุณไม่ต้องหยิบเอง
- ของที่ทำหน้าที่นี้เรียกว่า Skill

ลงมือทำท้ายบท

1. เปิดประวัติแชทของคุณ ย้อนไปสามสัปดาห์ หากคำสั่งที่พิมพ์ซ้ำในความหมายเดียวกันตั้งแต่สามครั้งขึ้นไป จดไว้หนึ่งชุด
2. ชัดเส้นแบ่งคำสั่งชุดนั้นเป็นสองส่วน ส่วนที่เป็นเนื้อหาของวันนั้น กับส่วนที่เป็นวิธีทำงานที่ใช้ได้ทุกครั้ง เขียนส่วนที่สองลงไฟล์เปล่าไฟล์หนึ่ง ตั้งชื่อไฟล์ตามงาน
3. ถามตัวเองสามข้อ งานนี้ต้องได้ผลหน้าตาแบบไหน มีศัพท์หรือรูปแบบอะไรที่หน่วยงานบังคับ และมีอะไรที่ห้ามทำเด็ดขาด เติมคำตอบทั้งสามลงไฟล์เดียวกัน เก็บไว้ใช้ในบทถัดไป

บทที่ 2

Skill คือโฟลเดอร์ ไม่ใช่เวกมนตร์

ชื่อฟังดูใหญ่ พอเปิดเข้าไปดูข้างในมีไฟล์ข้อความไฟล์เดียว เขียนด้วยภาษาคน อ่านออกตั้งแต่บรรทัดแรก

เปิดโฟลเดอร์ดูสักครั้ง

คนส่วนใหญ่ได้ยินคำว่า Skill แล้วนึกถึงของที่ต้องติดตั้ง ต้องมีสิทธิ์ ต้องเรียกช่างมาลง

ความจริงคือ Skill หรือชื่อเต็มว่า Agent Skills คือ **โฟลเดอร์** โฟลเดอร์แบบเดียวกับที่คุณสร้างบนหน้าจอกูทวิน คลิกขวา สร้างโฟลเดอร์ใหม่ ตั้งชื่อ เสร็จ

ในโฟลเดอร์นั้นต้องมีไฟล์ชื่อ **SKILL.md** หนึ่งไฟล์ นั่นคือหัวใจ ข้างในบรรจุคำสั่ง ความรู้เฉพาะทางของงานนั้น และชี้ไปยังไฟล์ประกอบถ้ามี AI จะโหลดมาใช้เมื่อเจองานที่ตรงกับที่เขียนไว้

นามสกุล **.md** คือไฟล์ Markdown เป็นไฟล์ข้อความธรรมดา เปิดด้วยโปรแกรมแก้ไขข้อความอะไรก็ได้ ไม่ต้องชื่อ ไม่ต้องลง ไม่มีรหัสลับ คุณอ่านมันได้ด้วยตาเปล่า คนในทีมก็อ่านได้

โฟลเดอร์เต็มรูปแบบมีสี่ส่วน ส่วนแรกบังคับ อีกสามส่วนไม่บังคับ

- **SKILL.md** — บังคับ ไฟล์เดียวที่ขาดไม่ได้
- **scripts/** — ไม่บังคับ
- **references/** — ไม่บังคับ
- **assets/** — ไม่บังคับ

ข้างใน SKILL.md มีสองส่วน

เปิดไฟล์ขึ้นมา คุณจะเห็นสองส่วนแยกกันชัดเจน

ส่วนบนคือ **frontmatter** เขียนด้วยรูปแบบ YAML คั่นหัวท้ายด้วยขีดสามขีด ส่วนนี้คือป้ายชื่อของ Skill สั้น ๆ ไม่ที่บรรทัด

ส่วนล่างคือเนื้อคู่มือ เขียนเป็น Markdown ตรงนี้คือที่ที่คุณเล่าวิธีทำงาน ทั้งหมด เขียนยาวได้ ใส่หัวข้อได้ ใส่รายการได้ เขียนเหมือนสอนน้องใหม่ในหน่วยงาน

```
---
name: summarize-research-th
description: สรุปบทความวิจัยเป็นภาษาไทยทั้งทางการตามรูปแบบของหน่วยงาน ใช้เมื่อผู้
ใช้ส่งไฟล์บทความวิจัยมาแล้วขอสรุป ย่อ หรือขอลอดหัวขั้ววัตถุประสงค์ วิธีการ ผลการ
วิจัย และข้อจำกัด
---
```

name คือชื่อเรียก

name มีกติกาที่ต้องทำตาม ยาวไม่เกิน 64 ตัวอักษร ใช้ตัวพิมพ์เล็ก a ถึง z ตัวเลข และขีดกลางเท่านั้น ห้ามมี XML tag และห้ามใช้คำสงวนสองคำคือ "anthropic" กับ "claude"

description คือหัวใจของหัวใจ

description คือคำอธิบายว่า Skill นี้ทำอะไร และ **เมื่อไรที่ควรใช้** ต้องมีทั้ง สองอย่าง ยาวไม่เกิน 1,024 ตัวอักษร ห้ามเว้นว่าง ห้ามมี XML tag

ทำไมต้องบอกด้วยว่าใช้เมื่อไร เพราะ AI เลือก Skill จาก **description** เท่านั้น มันไม่ได้เปิดอ่านข้างในก่อนตัดสินใจ

คิดง่าย ๆ ว่าเป็นคือสันเพิ่มบนชั้นวาง ในห้องมีแฟ้มสามสิบเล่ม คุณอ่านแค่สันแฟ้มแล้วถึงเล่มที่ต้องการออกมา สันแฟ้มที่เขียนว่า "งานเอกสาร" ช่วยอะไรไม่ได้เลย สันแฟ้มที่เขียนว่า "สรุปบทความวิจัยภาษาไทย ใช้ตอนได้ไฟล์ pdf งานวิจัยมา" ถึงถูกเล่มตั้งแต่ครั้งแรก

เขียน description ให้คนอ่านเข้าใจ ไม่ใช่ให้ตัวเองเข้าใจ

ข้อผิดพลาดที่เจอบ่อยที่สุดคือเขียนสั้นเกิน เช่น "ช่วยเรื่องงานวิจัย" กว้างแบบนี้ Skill จะถูกหยิบมาใช้ผิดงาน หรือไม่ถูกหยิบเลย

อีกข้อคือลืมนบอกว่าใช้เมื่อไร บอกแต่จะทำอะไร AI เลยไม่รู้ว่าจะสถานการณ์ไหนควรเปิดแฟ้มเล่มนี้ ใส่คำที่ผู้ใช้จะพิมพ์จริงลงไปด้วย เช่น สรุปบทความ ถอดหัวข้อ ทำตารางสังเคราะห์

สามโฟลเดอร์ที่ค่อยเติมทีหลัง

สามโฟลเดอร์ที่เหลือไม่บังคับ มีไว้ตอนงานโตขึ้น แต่ละอันมีหน้าที่ต่างกันชัดเจน

scripts/ เก็บสคริปต์ โปรแกรมสั้น ๆ ที่ทำงานซ้ำ ๆ ให้แม่นกว่าคน เช่น แปลงไฟล์ นับข้อมูล จัดรูปแบบตาราง คุณไม่ต้องเขียนเองตั้งแต่วันแรก และหลายคนไม่เคยต้องมีเลย

references/ เก็บเอกสารอ้างอิงที่ยาวเกินกว่าจะยัดลง SKILL.md เช่น คู่มือรูปแบบการเขียนของหน่วยงานสิบหน้า รายการศัพท์บัญญัติ เกณฑ์การให้คะแนน คู่มือหลักจะชี้ว่าให้เปิดไฟล์ไหนตอนไหน

assets/ เก็บของที่เอาไปใช้ต่อ ไม่ใช่ของที่เอาไว้อ่าน เช่น ไฟล์แม่แบบเอกสาร โลโก้ ฟอนต์ ตารางเปล่าที่รอเติมข้อมูล

โครงสร้างโฟลเดอร์ Skill

```

apa7-table/
├── SKILL.md
├── references/
├── apa7-rules.md
├── scripts/
├── check-decimals.py
├── assets/
└── table-example.md

```

บังคับต้องมี
คือคำอธิบายบวกขึ้นตอนทั้งหมด

เอกสารอ้างอิงยาว ๆ
เปิดเมื่อคู่มือสั่งให้เปิด

สคริปต์ที่ AI สั่งรันได้
ตัวโค้ดไม่กินพื้นที่

เกมเพลตและตัวอย่างไฟล์ที่ต้องทำตาม

มีแค่ SKILL.md ไฟล์เดียวก็ใช้ได้แล้ว สามโฟลเดอร์ที่เหลือเพิ่มทีหลังได้

ภาพที่ 3 กายวิภาคโฟลเดอร์ Skill และหน้าที่ของแต่ละส่วน

ทำไมต้องแยกเป็นโฟลเดอร์ ไม่เขียนรวมไฟล์เดียวให้อบ

เพราะไฟล์ที่ไม่ถูกเปิด ไม่กินที่ในความจำของ AI เลย

คุณจึงเก็บคู่มือหน่วยงานสืบหน้าไว้ใน references/ ได้โดยไม่เปลือง AI
จะเปิดก็ต่อเมื่อคู่มือหลักสั่งให้เปิด งานที่ไม่ต้องใช้ ก็ไม่ต้องแบก

สั่งงานปากเปล่า กับ คู่มือที่วางไว้บนโต๊ะ

ความต่างระหว่าง prompt กับ Skill อธิบายด้วยภาพในห่อผู้ช่วยได้ตรงที่สุด

prompt คือการสั่งงานปากเปล่า คุณเดินไปบอกน้องว่า "ช่วยจัดเวรเดือนหน้าให้หน่อย คนเก่าสองคน อย่าให้ใครขึ้นดึกติดกันเกินสามคืน" น้องทำได้ ทำได้ดีด้วย แต่คำพูดนั้นลอยอยู่ในอากาศ เดือนหน้าคุณต้องพูดใหม่ น้องคนอื่นที่ไม่ได้ยินก็ไม่รู้

Skill คือคู่มือปฏิบัติงาน SOP ที่วางไว้บนโต๊ะ เขียนไว้ว่างานจัดเวรทำอย่างไร เจอนไขมีอะไรบ้าง ห้ามอะไร ใครมารับเวรก็หยิบอ่านได้ คนที่เขียนไม่ต้องอยู่ตรงนั้น และเมื่อเจอนไขเปลี่ยน คุณแก้ที่คู่มือเล่มเดียว ทุกคนได้ของใหม่พร้อมกัน

prompt กับ Skill ต่างกันตรงไหน

หัวข้อ	prompt	Skill
รูปแบบ	ข้อความที่พิมพ์ในแชท	โฟลเดอร์ที่มี SKILL.md
อายุ	จบพร้อมแชท	อยู่จนกว่าจะลบ
ใครหยิบมาใช้	เราพิมพ์เองทุกครั้ง	AI หยิบเองเมื่อตรงงาน
แก้ไข	แก้ละที่ที่ก๊อปไป	แก้ที่ไฟล์เดียว
ส่งต่อ	ส่งข้อความให้เพื่อน	ส่งโฟลเดอร์ให้ทั้งทีม
ความยาวที่รับไหว	ยาวมากแล้วอ่านยาก	แนบเอกสารเพิ่มได้

Prompt

- ส่งงานปากเปล่าหน้าเคาน์เตอร์
- อยู่แค่ในแชทนี้
- ต้องพิมพ์ใหม่ทุกครั้ง
- ส่งต่อให้ทีมไม่ได้
- ผลลัพธ์แล้วแต่ว่าวันนั้นพิมพ์ครบไหม

Skill

- คู่มือปฏิบัติงานที่วางไว้บนโต๊ะ
- อยู่เป็นไฟล์ในโฟลเดอร์
- AI หยิบมาใช้เอง
- ส่งต่อได้เหมือนส่งไฟล์
- ใครมาเข้าเวรก็ได้ผลเหมือนกัน

ต่างกันที่ว่าวิธีทำงานถูกเก็บไว้ที่ไหน ไม่ใช่ที่ว่าเราสั่งเก่งแค่ไหน

ภาพที่ 4 Prompt กับ Skill ต่างกันตรงไหน

prompt คือคำพูดที่ลอยอยู่ในอากาศ Skill คือคู่มือที่วางอยู่บนโต๊ะ

AI หยิบมาใช้เอง โดยเราไม่ต้องสั่ง

ตรงนี้คือจุดที่คนใหม่มักไม่เชื่อจนกว่าจะได้ลอง

คุณไม่ต้องพิมพ์ว่า "ใช้ Skill ชื่อนี้" คุณพิมพ์งานที่ต้องการตามปกติ ระบบมีรายชื่อ Skill ทั้งหมดพร้อมคำอธิบายอยู่ในมืออยู่แล้ว มันเทียบสิ่งที่คุณขอกับคำอธิบายเหล่านั้น เจอเล่มที่ตรง ก็เปิดคู่มือเล่มนั้นอ่านเต็ม ๆ แล้วทำตาม

ลำดับจึงเป็นสามจังหวะ รู้ว่ามีอะไรอยู่บ้างจากคำอธิบาย เลือกเล่มที่ตรงแล้วอ่านคู่มือ ลงมือทำตามคู่มือและเปิดไฟล์ประกอบเท่าที่ต้องใช้

เพราะแบบนี้ **description** ถึงสำคัญกว่าที่หลายคนคิด คู่มือข้างในดีแค่ไหน ถ้าสันแฟ้มเขียนไม่ชัด ก็ไม่มีใครดึงออกมา

เรื่องจากหน้างาน

อาจารย์วิไลสอนระเบียบวิธีวิจัยมาสิบสองปี เธอมีข้อกำหนดการเขียนบทที่ 4 ของสาขาตัวเองอยู่ในหัว ตารางต้องมี n และร้อยละในช่องเดียวกัน ค่า p ต้องรายงานแบบไหน คำไหนห้ามใช้

เมื่อก่อนเธอพิมพ์อธิบายทั้งหมดนี้ให้ AI ฟังทุกครั้งที่นักศึกษาส่งงานมาให้ตรวจ วันละสามสี่รอบ

วันหนึ่งเธอเปิดไฟล์เปล่า เขียนข้อกำหนดพวกนั้นลงไป ตั้งชื่อไฟล์ว่า SKILL.md ใส่ไว้ในโฟลเดอร์หนึ่ง เขียนคำอธิบายไว้ว่าใช้ตอนตรวจบทที่ 4

วันถัดมาเธอพิมพ์แค่ "ช่วยดูบทที่ 4 ของนักศึกษาคนนี้ให้หน่อย" แล้วผลที่ได้ตรงมาตรงฐานของเธอทุกข้อ ทั้งที่เธอไม่ได้เอ่ยถึงไฟล์นั้นเลยสักคำ

Skill ไม่ใช่อะไร

ตัดความเข้าใจผิดสามข้อทั้งตั้งแต่ตอนนี้

ไม่ใช่โมเดลใหม่ Skill ไม่ได้ทำให้ AI ฉลาดขึ้น ความสามารถพื้นฐานเท่าเดิมทุกอย่าง สิ่งที่เปลี่ยนคือมันรับรู้บริบทของงานคุณ รู้ว่าหน่วยงานคุณใช้รูปแบบไหน

เหมือนพนักงานใหม่ที่เก่งอยู่แล้ว พอได้อ่านคู่มือหน่วยงานก็ทำงานได้ตรงขึ้น ไม่ใช่เพราะฉลาดขึ้น แต่เพราะรู้จักกา

ไม่ใช่หลักการที่ต้องเขียนโค้ด คุณไม่ต้องรู้ภาษาโปรแกรม ไม่ต้องคอมไพล์ ไม่ต้องเข้าใจคำว่า API เครื่องมือที่ใช้คือโปรแกรมพิมพ์ข้อความ ทักษะที่ใช้คือการอธิบายงานให้คนอื่นเข้าใจ ซึ่งพยาบาลและอาจารย์ทำเป็นอยู่แล้ว

ไม่ใช่ MCP MCP server เป็นองค์ประกอบคนละชนิด อยู่คนละชั้นกับ Skill มันเป็นส่วนที่ต้องตั้งค่าและทำงานอยู่เบื้องหลัง ส่วน Skill เป็นไฟล์ข้อความที่วางไว้เฉย ๆ ไม่มีอะไรต้องรัน ไม่มีอะไรต้องเชื่อมต่อ สองอย่างนี้อยู่ร่วมกันได้ แต่ไม่ใช่ของสิ่งเดียวกัน

มีแค่ SKILL.md ไฟล์เดียวก็ใช้ได้แล้ว

ข้อนี้ขอย้ำ เพราะเป็นข้อที่ทำให้คนไม่กล้าเริ่ม

คุณไม่ต้องมี `scripts/` คุณไม่ต้องมี `references/` คุณไม่ต้องมี `assets/` โฟลเดอร์ที่มีไฟล์เดียวก็เป็น Skill ที่สมบูรณ์แล้ว ใช้งานได้จริง ไม่ได้ด้อยกว่าของใครในทางใดเลย

Skill ตัวแรกของคนส่วนใหญ่ยาวไม่ถึงหน้ากระดาษ มีป้ายชื่อสองบรรทัด กับคู่มืออีกยี่สิบบรรทัด แค่นั้น แล้วค่อยเติมทีหลังเมื่อเจอปัญหาจริงว่าขาดอะไร

เขียนไว้เกินความจำเป็นตั้งแต่วันแรก มีแต่จะทำให้เลิกละกลางทาง

เก็บกลับบ้าน

- Skill คือโฟลเดอร์ที่มีไฟล์ SKILL.md เป็นหัวใจ ไม่ใช่โปรแกรมที่ต้องติดตั้ง
- SKILL.md มีสองส่วน frontmatter ที่เป็นป้ายชื่อ กับเนื้อคู่มือที่เขียนเป็นภาษาคน
- frontmatter บังคับสองช่อง name กับ description และ description ต้องบอกทั้งว่าทำอะไรและใช้เมื่อไร
- AI เลือก Skill จาก description เท่านั้น มันไม่ได้เปิดอ่านข้างในก่อนตัดสินใจ
- Skill ไม่ใช่โมเดลใหม่ ไม่ใช่ปลั๊กอินที่ต้องเขียนโค้ด และไม่ใช่ MCP
- มีแค่ SKILL.md ไฟล์เดียวก็เป็น Skill ที่ใช้งานได้จริง

ลงมือทำท้ายบท

1. สร้างโฟลเดอร์เปล่าหนึ่งอัน ตั้งชื่อเป็นภาษาอังกฤษตัวพิมพ์เล็กคั่นด้วยขีดกลาง แล้วสร้างไฟล์เปล่าชื่อ SKILL.md ไว้ข้างใน เท่านั้นโครงสร้างก็ครบแล้ว
2. เขียน description ของ Skill ที่คุณอยากได้ ให้ครบสองส่วนในประโยคเดียว ส่วนแรกบอกว่าทำอะไร ส่วนที่สองขึ้นต้นว่า "ใช้เมื่อ" แล้วใส่คำที่คุณจะพิมพ์จริงตอนสั่งงานลงไปด้วย
3. เอา description ที่เขียนไปให้เพื่อนร่วมงานอ่าน โดยไม่อธิบายอะไรเพิ่ม ถ้าเพื่อนบอกได้ทันทีว่าควรหยิบใช้ตอนไหน แปลว่าใช้ได้ ถ้าเพื่อนต้องถามกลับ ให้แก้แล้วลองใหม่

บทที่ 3

กายวิภาคของ SKILL.md

ไฟล์เดียว บรรทัดไม่กี่สิบบรรทัด แต่มีกติกาที่ถ้าพลาดแล้วระบบไม่รู้จักรหัสนี้เลย เราจะผ่านมันดูทีละส่วนเหมือนดูแผ่นกายวิภาค

ผ่าไฟล์ออกเป็นสองท่อน

วางไฟล์ SKILL.md ลงบนโต๊ะแล้วผ่าตามยาว คุณจะได้อีกสองท่อน ไม่มีท่อนที่สาม

ท่อนบนคือ **frontmatter** อยู่บนสุดเสมอ ห้ามมีอะไรอยู่เหนือมัน แม้แต่บรรทัดว่างบรรทัดเดียว มันเขียนด้วยรูปแบบ YAML ซึ่งฟังดูน่ากลัว แต่น้ำตาจริง ๆ คือ "ชื่อช่อง ทวิภาค เว้นวรรค แล้วค่า" เท่านั้น ท่อนล่างคือเนื้อคู่มือ เขียนเป็น Markdown ธรรมดา

เส้นแบ่งสองท่อนคือขีดสามขีด `---` ต้องมีสองชุด ชุดหนึ่งอยู่บรรทัดแรกสุด อีกชุดปิดท้าย frontmatter

SKILL.md

name: apa7-table-formatter

description: จัดตารางผลสถิติ

ตามแบบ APA 7th ใช้เมื่อ...

ตารางผลสถิติ APA 7th

ขั้นตอน

1. ระบุสถิติและระดับข้อมูล
2. เลือกแม่แบบจาก references/
3. รายงาน effect size และ CI

เส้นคั่นบน เปิดส่วนป้ายชื่อ

ไม่เกิน 64 ตัวอักษร a-z ตัวเลข ชัดกลาง

ไม่เกิน 1,024 ตัวอักษร

บอกทำอะไรและใช้เมื่อไร

เส้นคั่นล่าง ปิดส่วนป้ายชื่อ

เนื้อคู่มือ เขียนเป็นขั้นตอนสั่งทำ

สองก่อนเท่านั้น ก่อนบนคือป้ายชื่อ ก่อนล่างคือคู่มือ

ภาพที่ 5 ฝ่าไฟล์ SKILL.md ที่ละบรรทัด

ขีดสามขีดต้องเป็นขีดสามขีดจริง ๆ

โปรแกรมพิมพ์งานบางตัวชอบ "ช่วย" เปลี่ยนขีดสั้นสามอันเป็นเส้นยาวเส้นเดียว พอกลายเป็นเส้นยาว ระบบอ่านไม่ออก แล้ว Skill ทั้งตัวก็เจ็บบหายไป นี่คือเหตุผลที่เราไม่พิมพ์ SKILL.md ใน Word

ช่อง name – ป้ายชื่อที่มีกติกาเข้ม

ช่องแรกคือ **name** กติกาสั้นแต่เข้ม จำสี่ข้อนี้ให้ขึ้นใจ

1. ยาวไม่เกิน 64 ตัวอักษร
2. ใช้ได้เฉพาะตัวพิมพ์เล็ก a ถึง z ตัวเลข และขีดกลาง
3. ห้ามมี XML tag
4. ห้ามใช้คำสวงสองคำ คือ "anthropic" และ "claude"

ข้อสองคือชื่อที่คนไทยพลาดบ่อยที่สุด เราอยากตั้งชื่อเป็นภาษาไทยให้ตัวเองอ่านง่าย ตั้งไม่ได้ค่ะ เว้นวรรคก็ไม่ได้ ชิดล่างก็ไม่ได้ ส่วนข้อสี่ไม่มีใครเดาได้เอง ตั้งชื่อว่า `claude-nursing-helper` ไม่ผ่าน ทั้งที่ฟังดูไม่มีพิษภัย

ชื่อที่ใช้ได้ กับ ชื่อที่ใช้ไม่ได้

ชื่อ	ผ่านไหม	เพราะอะไร
<code>nursing-care-plan-th</code>	ผ่าน	ตัวเล็ก ชิดกลาง สื่อความชัด
<code>apa7-table-formatter</code>	ผ่าน	มีตัวเลขได้ อ่านแล้วรู้ว่าทำอะไร
สรุปงานวิจัย	ไม่ผ่าน	เป็นภาษาไทย
Nursing Care Plan	ไม่ผ่าน	มีตัวพิมพ์ใหญ่และเว้นวรรค
<code>nursing_care_plan</code>	ไม่ผ่าน	ใช้ชิดล่าง ต้องเป็นชิดกลาง
<code>claude-research-helper</code>	ไม่ผ่าน	มีคำสงวน claude
helper	ผ่านแต่แย่	กว้างจนไม่รู้ว่าช่วยอะไร

แถวสุดท้ายสำคัญ ผ่านกติกาไม่ได้แปลว่าเป็นชื่อที่ดี ชื่อที่ดีคือชื่อที่คุณเห็นอีกหกเดือนข้างหน้าแล้วยังรู้กันที่ว่ามันทำอะไร สูตรง่าย ๆ คือ งาน บวกสิ่งที่ทำ เช่น `thesis-ch4-reviewer`

ชื่อไฟล์เดอร์ควรตรงกับ name

ไม่ใช่กติกาบังคับ แต่เป็นนิสัยที่ช่วยชีวิตคุณตอนมี Skill สืบตัว ไฟลเดอร์ชื่อ `apa7-table-formatter` ข้างในมี SKILL.md ที่

```
name: apa7-table-formatter
```

แค่นี้ก็หาเจอโดยไม่ต้องเปิดไฟล์ที่ละอัน