THE BOOK BRAINS

เข้าใจง่าย
ใช้ได้จริง
ตั้งแต่วันนี้!

คู่มือใช้ AI ด้วยภาษาคน
สั่งงานเป็น ได้งานดี **ชีวิตง่ายขึ้นทันที**

อัปเดตล่าสุด
2026
ใช้ได้กับทุก AI
ChatGPT, Gemini,
Claude, Copilot
และอีกมากมาย
★★★★★

PROMPT ENGINEER

ไม่ยาก อย่างที่คิด

ใช้ AI ได้อย่างมืออาชีพ
ภายใน
7วัน

เรียนวันละบท
ทำตามได้จริง!

จากคนใช้ AI ไม่เป็น
สู่คนที่**สั่ง AI ได้**เหมือนมีผู้ช่วยส่วนตัว

- เข้าใจง่าย ไม่ต้องมีพื้นฐาน
- เทคนิคใช้ได้กับทุกงาน ทุกอาชีพ
- ตัวอย่างจริง พร้อม Prompt ใช้ได้ทันที
- อัปเดตเทคนิค **AI ปี 2026**



คุยกับ AI เข้าใจง่าย เหมือนคุยกับคน	สั่งงานชัด ได้ผลลัพธ์ตรงใจ	คิดเป็นระบบ ให้ AI คิดแทนคุณ	ทำงานเร็วขึ้น ประหยัดเวลา ได้มากขึ้น	ผลงานดีขึ้น ระดับมืออาชีพ

ไม่ต้องจำสูตร ไม่ต้องใช้ศัพท์เทคนิคเยอะ
แค่เข้าใจวิธีคิด ก็เขียน Prompt เองได้ทุกงาน!

NOVA SCIENCE

ผู้จัดทำ : ศักดิ์สุตา ฉันทเตยานนท์

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537

ห้ามทำซ้ำ คัดลอก ดัดแปลง เผยแพร่โดยไม่ได้รับอนุญาต

สารบัญ

บทนำ — Prompt Engineer ทำงานอะไร? 7

ภาค 1 — เข้าใจวิธีคิดก่อนเขียน Prompt

บทที่ 1 — ทำไม AI ถึงตอบไม่ตรงที่เราต้องการ 16

บทที่ 2 — จาก “คำถาม” สู่ “โจทย์งาน” 25

บทที่ 3 — Prompt หนึ่งคำสั่งประกอบด้วยอะไรบ้าง 39

บทที่ 4 — ทำอย่างไรให้ AI เข้าใจสิ่งที่เราต้องการจริง ๆ 52

บทที่ 5 — ควบคุมคุณภาพคำตอบตั้งแต่ต้นทาง..... 64

ภาค 2 — ออกแบบ Prompt

บทที่ 6 — เลือกวิธีเขียน Prompt ให้เหมาะกับงาน 78

บทที่ 7 — บทบาท คำสั่ง และบริบท: ใช้อย่างไรให้เกิดผล 92

บทที่ 8 — เมื่อไรควรให้ตัวอย่าง และเมื่อไรไม่ต้องให้ 106

บทที่ 9 — สั่ง AI ให้ตอบในรูปแบบที่กำหนด 119

บทที่ 10 — แบ่งงานยากให้ AI ทำทีละขั้น 133

ภาค 3 — นำ Prompt ไปใช้กับงานจริง

บทที่ 11 — ใช้ AI เขียนงานให้ตรงกลุ่มเป้าหมาย 148
บทที่ 12 — ใช้ AI ช่วยคิดธุรกิจและการตลาด 164
บทที่ 13 — ใช้ AI เป็นผู้ช่วยเรียนรู้ ไม่ใช่แค่หาคำตอบ 177
บทที่ 14 — ใช้ AI วิเคราะห์ข้อมูลอย่างมีเหตุผล 189
บทที่ 15 — ออกแบบ AI ให้เป็นผู้ช่วยส่วนตัว 201

ภาค 4 — เริ่มทำงานแบบ Prompt Engineer

บทที่ 16 — เปลี่ยนปัญหาจริงให้กลายเป็น Prompt 216
บทที่ 17 — ออกแบบ Prompt ที่ซับซ้อนให้เป็นระบบ 233
บทที่ 18 — ต่อ Prompt หลายขั้นให้กลายเป็น Workflow 249
บทที่ 19 — ทดสอบ Prompt อย่างเป็นระบบ 266
บทที่ 20 — วัดอย่างไรว่า Prompt ไหนดีกว่า 280
บทที่ 21 — ป้องกันคำตอบผิดและรู้ว่าเมื่อไรต้องให้คนตรวจ 293

ภาค 5 — จาก Prompt สู่ระบบ AI

บทที่ 22 — จัดการข้อมูลที่ AI ต้องใช้ในแต่ละงาน 307
บทที่ 23 — ให้ AI ค้นข้อมูลจากเอกสารก่อนตอบด้วย RAG 322

บทที่ 24 — ให้ AI เรียกใช้เครื่องมือและระบบภายนอก 334
บทที่ 25 — สร้าง AI Agent ที่คิดและทำงานหลายขั้นตอน..... 346
บทที่ 26 — เลือก AI Model ให้เหมาะกับงาน 359

ภาค 6 — ทำให้ระบบพร้อมใช้งานจริง

บทที่ 27 — ลดต้นทุนและการใช้ Token โดยไม่เสียคุณภาพ 372
บทที่ 28 — ทำให้ AI ตอบเร็วและทำงานได้สม่ำเสมอ 384
บทที่ 29 — จัดการ Prompt หลายเวอร์ชันอย่างเป็นระบบ 395
บทที่ 30 — ตรวจสอบว่า Prompt ใหม่ไม่ได้ทำให้ของเดิมพัง..... 407
บทที่ 31 — รู้ให้ได้ว่า AI ทำอะไรผิดและผิดตรงไหน 419

ภาค 7 — โปรเจกต์จบ

บทที่ 32 — สร้างระบบ AI ตั้งแต่โจทย์จนใช้งานจริง 432

บทนำ — Prompt Engineer ทำงานอะไร?

ทุกวันนี้ใคร ๆ ก็เปิด AI แล้วพิมพ์คำสั่งได้

จึงเกิดคำถามง่าย ๆ ว่า

ถ้าใครก็เขียน Prompt ได้ แล้ว Prompt Engineer ทำอะไร?

คำตอบคือ Prompt Engineer ไม่ได้มีหน้าที่เพียงเขียน Prompt ให้ AI ตอบดีในครั้งเดียว

แต่ต้องคิดต่อว่า

จะออกแบบการทำงานอย่างไรให้ AI ทำงานได้ถูกต้อง สม่ำเสมอ ตรวจสอบได้ และรู้ว่าเมื่อไรควรให้คนเข้ามาตัดสินใจ

ลองนึกถึงบริษัทที่มีข้อความจากลูกค้าเข้ามาหลายร้อยรายการต่อวัน

ถ้ามีพนักงานเปิด AI วางข้อความ พิมพ์คำสั่ง อ่านคำตอบ แก้คำตอบ แล้วทำซ้ำทุกครั้ง นั่นคือการใช้ AI ช่วยงาน

แต่คนยังเป็นศูนย์กลางของทุกขั้นตอน

Prompt Engineer จะถามต่อว่า

- จะส่งข้อมูลอะไรเข้า AI?
- ต้องให้ AI ทำงานตามกติกาอะไร?
- ผลลัพธ์ต้องมีรูปแบบอย่างไร?
- ถ้าข้อมูลไม่ครบจะทำอย่างไร?
- ถ้า AI ตอบผิดจะรู้ได้อย่างไร?
- ขั้นตอนไหนให้ AI ทำเองได้?
- ขั้นตอนไหนต้องให้คนตรวจสอบ?

นี่คือความแตกต่างระหว่าง **การใช้ AI** กับ **การออกแบบการทำงานร่วมกับ AI**

Prompt Engineer ไม่ใช่คนที่เขียน Prompt ยาวที่สุด

ความเข้าใจผิดที่พบบ่อยคือ Prompt Engineer ต้องรู้เทคนิค Prompt จำนวนมาก หรือสามารถเขียนคำสั่งที่ยาวและซับซ้อนได้

แต่ในความเป็นจริง Prompt ที่ยาวไม่ได้หมายความว่าจะดีกว่าเสมอไป

Prompt ที่ดีคือ Prompt ที่ทำให้ AI มีข้อมูลและเงื่อนไขเพียงพอที่จะทำงานตามเป้าหมาย

ดังนั้น สิ่งที่ Prompt Engineer ต้องทำไม่ใช่เพียงการหาคำสั่งที่ “ดูดี”

แต่คือการตอบคำถามว่า

งานนี้ต้องสำเร็จอย่างไร?

จากนั้นจึงค่อยออกแบบวิธีให้ AI ทำงานไปสู่ผลลัพธ์นั้น

ผู้ใช้ AI กับ Prompt Engineer คิดต่างกันอย่างไร?

ผู้ใช้ AI ทั่วไปอาจเริ่มด้วยคำถามว่า

“ฉันจะเขียน Prompt อย่างไร?”

แต่ Prompt Engineer จะคิดต่อว่า

“AI ต้องรู้อะไรจึงจะทำงานนี้ได้?”
“ต้องทำอะไรกับข้อมูล?”
“ผลลัพธ์ที่ดีหน้าตาเป็นอย่างไร?”
“ถ้าข้อมูลไม่พอจะทำอย่างไร?”
“ถ้า AI ทำผิด เราจะรู้ได้อย่างไร?”
“ส่วนไหนควรให้ AI ทำ และส่วนไหนควรให้คนตัดสินใจ?”

นี่คือการเปลี่ยนจาก **การสั่ง AI** ไปสู่ **การออกแบบการทำงานของ AI**

และนี่คือหัวใจของ Prompt Engineering

ต้องเป็นโปรแกรมเมอร์หรือไม่?

ไม่จำเป็นต้องเป็นโปรแกรมเมอร์เพื่อเริ่มเรียนรู้ Prompt Engineering

คุณไม่จำเป็นต้องจำ Prompt หลายร้อยแบบ และไม่จำเป็นต้องรู้ทุกอย่างเกี่ยวกับ AI

สิ่งสำคัญกว่าคือความสามารถในการ

เข้าใจปัญหา → ออกแบบคำสั่ง → จัดการข้อมูล → กำหนดผลลัพธ์ → ทดสอบ → วัดผล → ปรับปรุง

เมื่อทำสิ่งเหล่านี้ได้อย่างเป็นระบบ คุณก็ไม่ได้เป็นเพียงคนที่ “ใช้ AI เป็น”

แต่กำลังเริ่มคิดแบบ Prompt Engineer

หนังสือเล่มนี้จะพาคุณไปถึงไหน?

หนังสือเล่มนี้ไม่ได้เริ่มจากการแจก Prompt สำเร็จรูปจำนวนมาก

เพราะ Prompt ที่ใช้ได้กับงานหนึ่ง อาจไม่เหมาะกับอีกงานหนึ่ง

สิ่งที่สำคัญกว่าคือการเข้าใจ **วิธีคิดเบื้องหลัง Prompt**

คุณจะเริ่มจากการทำความเข้าใจว่า

ทำไม AI ถึงตอบไม่ตรงความต้องการ

จากนั้นจะเรียนรู้การเปลี่ยนความต้องการของมนุษย์ให้กลายเป็น **โจทย์งานที่ชัดเจน** และทำความเข้าใจองค์ประกอบต่าง ๆ ของ Prompt

เมื่อพื้นฐานแน่นแล้ว จะเข้าสู่การออกแบบ Prompt ให้เหมาะกับงานจริง ทั้งการใช้บทบาท บริบท ตัวอย่าง และรูปแบบผลลัพธ์

จากนั้นจึงนำ Prompt ไปใช้กับ

- งานเขียน
- ธุรกิจและการตลาด
- การเรียนรู้
- การวิเคราะห์ข้อมูล
- การสร้างผู้ช่วยส่วนตัว

แต่หนังสือจะไม่หยุดอยู่ที่การเขียน Prompt

คุณจะได้เรียนรู้การทำงานแบบ Prompt Engineer ตั้งแต่การเปลี่ยนปัญหาจริงให้เป็น Prompt การสร้าง Prompt ที่ซับซ้อน การต่อ Prompt เป็น Workflow การทดสอบ การวัดผล และการจัดการความผิดพลาด

จากนั้นจึงก้าวไปสู่การสร้างระบบ AI ที่ใช้

ข้อมูล → RAG → Tools → Agent → Model

รวมถึงเรื่องต้นทุน ความเร็ว ความสม่ำเสมอ การจัดการเวอร์ชัน การทดสอบการถดถอย และการ Debug

ท้ายที่สุด คุณจะนำทุกอย่างมาประกอบกันเป็นโปรเจกต์จริง ตั้งแต่

โจทย์ → ข้อมูล → Prompt → Workflow → ระบบ AI → การทดสอบ → การประเมิน → การใช้งานจริง

นี่คือเส้นทางที่หนังสือเล่มนี้ต้องการพาคุณเดินไป

ก่อนเริ่ม อย่าถามว่า “Prompt ไหนดีที่สุด?”

เมื่อเริ่มใช้ AI หลายคนจะพยายามหา Prompt ที่ดีที่สุด

แต่คำถามที่มีประโยชน์กว่าคือ

“งานนี้ต้องสำเร็จอย่างไร?”

จากนั้นถามต่อว่า

1. ต้องการผลลัพธ์อะไร?
2. ใครจะใช้ผลลัพธ์นั้น?
3. AI ต้องรู้อะไร?
4. มีข้อจำกัดอะไร?
5. ผลลัพธ์ควรมีรูปแบบอย่างไร?
6. จะรู้ได้อย่างไรว่าผลลัพธ์นั้นดีพอ?

คำถามเหล่านี้จะกลายเป็นพื้นฐานของการคิดแบบ Prompt Engineer

เส้นทางของทั้งเล่ม

หนังสือเล่มนี้จะพาคุณค่อย ๆ เปลี่ยนจาก

เข้าใจ AI

↓

วิเคราะห์โจทย์

↓

ออกแบบ Prompt

↓

นำไปใช้จริง

↓

ทดสอบและประเมิน

↓

สร้าง Workflow

↓

ใช้ RAG / Tools / Agent

↓

ปรับปรุงระบบ

↓

สร้างระบบ AI จริง

เมื่อเดินมาถึงช่วงท้าย คุณจะเห็นว่า Prompt Engineering ไม่ได้เกี่ยวกับการหาคำพูดวิเศษเพื่อสั่ง AI

แต่คือการออกแบบ **วิธีทำงานร่วมกันระหว่างคน ข้อมูล AI และระบบ**

สิ่งที่อยากให้จำก่อนเริ่มบทแรก

ตลอดทั้งเล่ม ขอให้จำแนวคิดหนึ่งไว้

Prompt Engineering ไม่ใช่การแข่งขันกันเขียนคำสั่งให้ยาวที่สุด

มันคือการทำให้ AI มีสิ่งที่จำเป็นต่อการทำงานอย่างเพียงพอ ทั้ง

เป้าหมาย ข้อมูล บริบท คำสั่ง เงื่อนไข ข้อจำกัด และเกณฑ์ของผลลัพธ์

เมื่อองค์ประกอบเหล่านี้ชัดเจนขึ้น AI ก็มีโอกาสทำงานตรงกับสิ่งที่เราต้องการมากขึ้น

และเมื่อเราเรียนรู้ไปถึงช่วงท้าย เราจะไม่ได้หยุดอยู่ที่ Prompt หนึ่งชุด

แต่จะค่อย ๆ ก้าวจาก

การใช้ AI → การออกแบบ Prompt → การออกแบบ Workflow → การสร้างระบบ AI

นี่คือเส้นทางของหนังสือเล่มนี้

และเราจะเริ่มจากคำถามพื้นฐานที่สุดก่อนว่า

ทำไม AI ถึงตอบไม่ตรงที่เราต้องการ ทั้งที่เราก็สั่งมันไปแล้ว?

บทที่ 1 — ทำไม AI ถึงตอบไม่ตรงที่เราต้องการ

เคยไหม?

พิมพ์คำสั่งให้ AI ไปอย่างหนึ่ง แต่คำตอบที่ได้กลับไม่ใช่สิ่งที่เราต้องการ

- ขอให้เขียนบทความ แต่เนื้อหากว้างเกินไป
- ขอให้สรุป แต่กลับยาวกว่าเดิม
- ขอให้เขียนให้เป็นธรรมชาติ แต่ยังรู้สึกเหมือนข้อความจาก AI
- ขอให้ช่วยคิดธุรกิจ แต่ได้คำตอบทั่วไปที่ใช้ได้กับแทบทุกคน

เมื่อเจอแบบนี้ หลายคนมักสรุปทันทีว่า

“AI ยังไม่ฉลาดพอ”

แต่ในความเป็นจริง ปัญหาจำนวนมากไม่ได้เกิดจาก AI ไม่เก่ง

ปัญหาเกิดจาก **AI ไม่รู้ว่าเราต้องการอะไรอย่างแท้จริง**

AI ไม่ได้รู้สิ่งที่อยู่ในหัวเรา

เวลาคุยกับคน เราสามารถพูดสั้น ๆ แล้วอีกฝ่ายเข้าใจได้ เพราะเขามีบริบทเกี่ยวกับเราและเรื่องที่กำลังคุยกันอยู่

สมมติคุณกำลังทำเอกสารกับเพื่อน แล้วพูดว่า

“ช่วยทำให้มันดีขึ้นหน่อย”

เพื่อนอาจเข้าใจว่าคุณหมายถึงอะไร เพราะเขารู้ว่าเอกสารนี้ทำขึ้นเพื่ออะไร รู้ว่าคุณต้องการสื่อสารกับใคร และอาจรู้ด้วยว่าคุณชอบรูปแบบไหน

แต่ AI ไม่มีข้อมูลเหล่านั้น หากเราไม่ได้บอก

คำว่า **“ดีขึ้น”** อาจหมายถึง

- กระชับขึ้น
- อ่านง่ายขึ้น
- ดูเป็นมืออาชีพขึ้น
- น่าสนใจขึ้น
- เป็นธรรมชาติมากขึ้น
- เหมาะกับกลุ่มเป้าหมายมากขึ้น

AI จึงต้องตีความจากข้อมูลที่ได้รับ

นี่คือเหตุผลว่าทำไมคำสั่งที่ดูเหมือนชัดเจนสำหรับเรา อาจยังไม่ชัดเจนสำหรับ AI

Prompt ยาวไม่ได้แปลว่าดี

เมื่อ AI ตอบไม่ตรง หลายคนแก้ปัญหาด้วยการเพิ่มข้อความเข้าไปเรื่อย ๆ

จาก

“ช่วยเขียนโพสต์ขายสินค้า”

กลายเป็น

“ช่วยเขียนโพสต์ขายสินค้าให้ละเอียด น่าสนใจ เป็นธรรมชาติ ดูเป็นมืออาชีพ ไม่ขายตรงเกินไป อ่านง่าย ดึงดูดความสนใจ และทำให้คนอยากซื้อ...”

Prompt ยาวขึ้น แต่ไม่ได้หมายความว่า AI เข้าใจงานมากขึ้นเสมอไป

สิ่งที่ควรถามคือ

“ข้อมูลอะไรที่จะช่วยให้ AI ตัดสินใจได้ดีขึ้น?”

ไม่ใช่

“ฉันต้องเพิ่มคำอีกกี่คำ?”

Prompt ที่ดีจึงไม่ได้วัดจากความยาว แต่วัดจากความชัดเจนและความเพียงพอของข้อมูลที่ใช้ทำงาน

สาเหตุที่ AI ตอบไม่ตรง

โดยทั่วไป ปัญหามักเกิดจากองค์ประกอบของงานยังไม่ชัดเจนพอ

1. เป้าหมายไม่ชัด

เช่น

“ช่วยเขียนบทความเรื่อง AI”

AI ไม่รู้ว่าต้องการบทความเพื่ออะไร

อาจเป็นบทความให้ความรู้ บทความขายสินค้า บทความสำหรับผู้บริหาร หรือบทความสำหรับผู้เริ่มต้น

2. ข้อมูลไม่เพียงพอ

เช่น

“ช่วยเขียนอีเมลตอบลูกค้าให้หน่อย”

AI ไม่รู้ว่า

- ลูกค้าถามอะไร
- เกิดปัญหาอะไร
- ลูกค้าเป็นใคร
- บริษัทมีนโยบายอย่างไร
- เราต้องการตอบแบบไหน

- มีข้อจำกัดอะไร
- มีทางเลือกอะไรให้ลูกค้าบ้าง

จึงต้องสร้างคำตอบจากข้อมูลที่มีอยู่

3. บริบทไม่ครบ

ลองเพิ่มข้อมูลว่า

“ลูกค้าขอคืนเงิน แต่สินค้าถูกใช้งานแล้ว เราต้องการปฏิเสธคำขออย่างสุภาพ พร้อมเสนอทางเลือกอื่นให้ลูกค้า”

ทันทีที่มีบริบทเพิ่มขึ้น AI ก็เข้าใจสถานการณ์มากขึ้น และสามารถสร้างคำตอบที่เหมาะกับงานจริงได้มากกว่าเดิม

สังเกตว่าเราไม่ได้ใช้คำสั่งที่ซับซ้อนขึ้นมาก

เราเพียง **ให้ข้อมูลที่ AI จำเป็นต้องรู้**

4. ผลลัพธ์ที่ต้องการไม่ชัด

คำสั่งว่า

“ช่วยสรุปเอกสารนี้”

อาจได้ทั้งสรุปแบบสั้น สรุปแบบละเอียด สรุปเป็นย่อหน้า หรือสรุปเป็นรายการ

ถ้าเราต้องการรูปแบบใดรูปแบบหนึ่ง ก็ควรระบุให้ชัด

5. AI ต้องเดาในสิ่งที่เราไม่ได้บอก

นี่คือปัญหาที่พบได้บ่อยที่สุด

เมื่อข้อมูลหรือเงื่อนไขสำคัญหายไป AI จะต้องเติมส่วนที่ขาดด้วยการตีความ

และเมื่อการตีความของ AI ไม่ตรงกับสิ่งที่เราอยู่ในใจ ผลลัพธ์ก็ไม่ตรงกับความต้องการ

ดังนั้น เป้าหมายของ Prompt Engineering ส่วนหนึ่งคือ

ลดสิ่งที่ AI ต้องคาดเดา

อย่ารีบแก้ด้วยการเพิ่มคำสั่ง

เมื่อ AI ตอบไม่ตรง อย่าเพิ่งเขียน Prompt ใหม่ทั้งหมด

ให้ถามก่อนว่า

“AI ผิดเพราะอะไร?”

ถ้า AI **ไม่เข้าใจงาน**
→ เพิ่มบริบท

ถ้า AI **ทำงานไม่ครบ**
→ เพิ่มขั้นตอน

ถ้า AI **ตอบยาวเกินไป**
→ กำหนดรูปแบบหรือความยาว

ถ้า AI **ทำเกินคำสั่ง**

→ เพิ่มขอบเขต

ถ้า AI **สรุปเกินข้อมูล**

→ กำหนดให้แยกข้อเท็จจริงกับสมมติฐาน

การแก้ Prompt ที่ดีจึงไม่ใช่การสุ่มเปลี่ยนคำไปเรื่อย ๆ แต่คือการวิเคราะห์สาเหตุของปัญหาแล้วแก้ตรงจุด

คิดแบบ Prompt Engineer

คนทั่วไปอาจคิดว่า

“ทำอย่างไรให้ AI ตอบดีขึ้น?”

แต่ Prompt Engineer จะถามละเอียดกว่านั้นว่า

“AI ไม่ดีตรงไหน?”

จากนั้นจึงหาว่าปัญหาอยู่ที่

เป้าหมาย → ข้อมูล → บริบท → คำสั่ง → ข้อจำกัด → ผลลัพธ์

หรือส่วนอื่นของกระบวนการ

นี่คือการเปลี่ยนจาก **การแก้คำ** ไปสู่ **การแก้ปัญหา**

ลองทำตามทีละขั้น

เลือกงานหนึ่งอย่างที่คุณใช้ AI อยู่จริง เช่น

- เขียนบทความ
- สรุปเอกสาร
- เขียนอีเมล
- วิเคราะห์ข้อมูล
- คิดไอเดียธุรกิจ

แล้วใช้ Prompt สั้น ๆ ก่อน

“ช่วยทำงานนี้ให้หน่อย”

จากนั้นดูผลลัพธ์และถามตัวเอง 4 ข้อ

1. **AI เข้าใจเป้าหมายหรือไม่?**
2. **AI มีข้อมูลเพียงพอหรือไม่?**
3. **AI ต้องเดาอะไรเองหรือไม่?**
4. **ผลลัพธ์ที่ได้มีปัญหาตรงไหน?**

จากนั้นแก้ **เฉพาะจุดที่มีปัญหา**

อย่าเพิ่มทุกอย่างพร้อมกัน เพราะถ้าเปลี่ยนหลายองค์ประกอบในครั้งเดียว คุณจะไม่รู้ว่าอะไรเป็นสาเหตุให้ผลลัพธ์ดีขึ้นหรือแย่ลง

แบบฝึกหัดท้ายบท

เลือก Prompt ที่คุณใช้จริงหนึ่งคำสั่ง

ตัวอย่าง

“ช่วยเขียนโพสต์ขายสินค้า”

จากนั้นลองหาว่า AI ต้องเดาอะไรบ้าง

เช่น

- สินค้าคืออะไร?
- ลูกค้าคือใคร?
- ต้องการขายหรือสร้างการรับรู้?
- ใช้ช่องทางไหน?
- ต้องการน้ำเสียงแบบใด?
- ต้องการความยาวเท่าไร?
- มีอะไรที่ห้ามพูด?

จากนั้นเลือกเพิ่มเฉพาะข้อมูลที่จำเป็นที่สุด แล้วทดลองอีกครั้ง

สิ่งที่ต้องสังเกตไม่ใช่แค่ว่า **Prompt ยาวขึ้นหรือไม่**

แต่คือ

“AI ต้องเดาน้อยลงตรงไหน?”

บทที่ 2 — จาก “คำถาม” สู่ “โจทย์งาน”

คนส่วนใหญ่เริ่มใช้ AI ด้วยการคิดว่า

“ฉันควรพิมพ์อะไรดี?”

แต่คนที่เริ่มคิดแบบ Prompt Engineer จะเปลี่ยนคำถามเป็น

“งานนี้ต้องสำเร็จอย่างไร?”

ความแตกต่างดูเหมือนเล็ก แต่จริง ๆ แล้วเปลี่ยนวิธีทำงานกับ AI ทั้งหมด

เพราะ Prompt ไม่ควรเป็นจุดเริ่มต้นของการคิด แต่ควรเป็น **ผลลัพธ์จากการวิเคราะห์งาน**

จากผู้ใช้ AI ไปสู่คนที่ออกแบบงานให้ AI

ลองดูคำสั่งง่าย ๆ

“ช่วยเขียนรายงาน”

คำสั่งนี้ไม่ได้ผิด แต่ยังไม่บอกว่าเรากำลังพยายามแก้ปัญหาอะไร

AI ยังไม่รู้ว่า

- รายงานเรื่องอะไร
- เขียนให้ใคร

- ใช้เพื่ออะไร
- มีข้อมูลอะไรให้ใช้
- ต้องการผลลัพธ์แบบไหน

ถ้าเรารีบเขียน Prompt ทันที เราอาจกำลังพยายามแก้ปัญหาที่ตัวเองยังอธิบายไม่ได้

ดังนั้น ก่อนเขียน Prompt ให้เริ่มจาก **ปัญหา** ก่อน

ตัวอย่างเช่น

“ฉันต้องนำเสนอผลการดำเนินงานให้ผู้บริหาร และมีเวลาพูดเพียง 10 นาที”

ตอนนี้เราเริ่มเห็นโจทย์แล้ว

จากนั้นจึงเปลี่ยนเป็นงานที่ AI ช่วยได้ เช่น

“ช่วยสรุปผลการดำเนินงานให้เหมาะกับการนำเสนอผู้บริหารภายใน 10 นาที โดยเน้นผลลัพธ์ ความเสี่ยง และประเด็นที่ต้องตัดสินใจ”

เราจึงได้ลำดับว่า

ปัญหา → งาน → Prompt

ไม่ใช่

Prompt → แล้วค่อยคิดว่ากำลังแก้อะไร

นี่คือหลักสำคัญของบทนี้: **Prompt เป็นผลจากการวิเคราะห์งาน**

คำถามของมนุษย์ไม่ใช่โจทย์ของ AI เสมอไป

มนุษย์มักพูดถึงสิ่งที่ต้องการด้วยภาษาสั้น ๆ เพราะเรามีบริบทอยู่ในหัว

เช่น

“ช่วยวิเคราะห์ลูกค้าให้หน่อย”

สำหรับคนพูดอาจมีความหมายชัดเจน เพราะรู้อยู่แล้วว่าหมายถึงลูกค้ากลุ่มไหน ต้องการวิเคราะห์เรื่องอะไร และจะนำผลไปใช้ทำอะไร

แต่ AI ไม่ได้มีบริบททั้งหมดนั้น

ดังนั้น หน้าที่ของ Prompt Engineer คือเปลี่ยนความต้องการที่ยังคลุมเครือให้กลายเป็น **โจทย์งานที่สามารถลงมือทำได้**

ลองถามตัวเองว่า

1. **ต้องการผลลัพธ์อะไร?**
2. **ใครจะเป็นคนใช้ผลลัพธ์นั้น?**
3. **AI ต้องรู้อะไร?**
4. **มีข้อจำกัดอะไร?**
5. **ผลลัพธ์ควรมีรูปแบบอย่างไร?**
6. **จะรู้ได้อย่างไรว่าผลลัพธ์นั้นดีพอ?**

คำถามทั้งหกข้อนี้เป็นพื้นฐานสำคัญของการคิดแบบ Prompt Engineer

ขั้นตอนเปลี่ยนคำถามให้เป็นโจทย์งาน

ขั้นที่ 1 — ระบุปัญหาที่ต้องการแก้

อย่าเริ่มจากคำว่า “เขียน Prompt”

ให้เริ่มจาก

“ตอนนี้มีปัญหาอะไร?”

ตัวอย่าง

“ทีมบริการลูกค้าต้องอ่านข้อความจำนวนมาก ทำให้ใช้เวลาคัดแยกปัญหานาน”

นี่คือปัญหา ไม่ใช่ Prompt

ขั้นที่ 2 — ระบุงานที่ต้องทำ

เมื่อรู้ปัญหาแล้ว ให้ถามต่อว่า

“AI ต้องเข้ามาช่วยทำงานส่วนไหน?”

จากตัวอย่างเดิม อาจเป็น

“ให้ AI ช่วยจัดประเภทข้อความลูกค้าตามประเภทปัญหา”

ตอนนี้งานเริ่มชัดขึ้นแล้ว

ปัญหา: ใช้เวลาคัดแยกข้อความลูกค้านาน
งาน: จัดประเภทข้อความลูกค้า

ขั้นที่ 3 — ระบุข้อมูลที่ AI ต้องใช้

ต่อไปถามว่า

“AI ต้องเห็นข้อมูลอะไรจึงจะทำงานนี้ได้?”

เช่น

- ข้อความจากลูกค้า
- รายการหมวดหมู่ปัญหา
- คำอธิบายของแต่ละหมวดหมู่
- กฎหรือเงื่อนไขในการจัดประเภท

หลักสำคัญคือ **ข้อมูลที่มากที่สุดไม่จำเป็นต้องเป็นข้อมูลที่ดีที่สุด**

ควรให้ข้อมูลที่มีผลต่อการตัดสินใจของ AI และตรวจสอบได้

ขั้นที่ 4 — กำหนดผลลัพธ์ที่ต้องการ

คำว่า “วิเคราะห์” หรือ “สรุป” กว้างเกินไป

ลองเปลี่ยนจาก

“วิเคราะห์ความคิดเห็นลูกค้า”

เป็น

“จัดกลุ่มความคิดเห็นลูกค้าตามประเด็นปัญหา พร้อมสรุป 3 ปัญหาที่พบมากที่สุดและจำนวนครั้งที่พบ”

ตอนนี้ AI รู้แล้วว่าไม่ได้มีหน้าที่เพียง “อ่านแล้ววิเคราะห์” แต่ต้องส่งมอบผลลัพธ์ที่มีโครงสร้างชัดเจน

ขั้นที่ 5 — กำหนดข้อจำกัด

งานเดียวกันอาจให้ผลลัพธ์แตกต่างกันอย่างมาก หากข้อจำกัดไม่เหมือนกัน

ตัวอย่างเช่น

“สรุปให้ผู้บริหารอ่านภายใน 2 นาที”

ย่อมแตกต่างจาก

“สรุปให้ทีมปฏิบัติการใช้เป็นคู่มือทำงาน”

ดังนั้นต้องถามว่า

- มีความยาวจำกัดหรือไม่
- ใช้ภาษาแบบไหน
- ห้ามสร้างข้อมูลเพิ่มเติมหรือไม่
- ต้องใช้เฉพาะข้อมูลที่ให้หรือไม่
- ต้องจัดลำดับความสำคัญหรือไม่
- มีข้อกำหนดเฉพาะขององค์กรหรือไม่

ข้อจำกัดเหล่านี้ช่วยลดสิ่งที่ AI ต้องคาดเดา