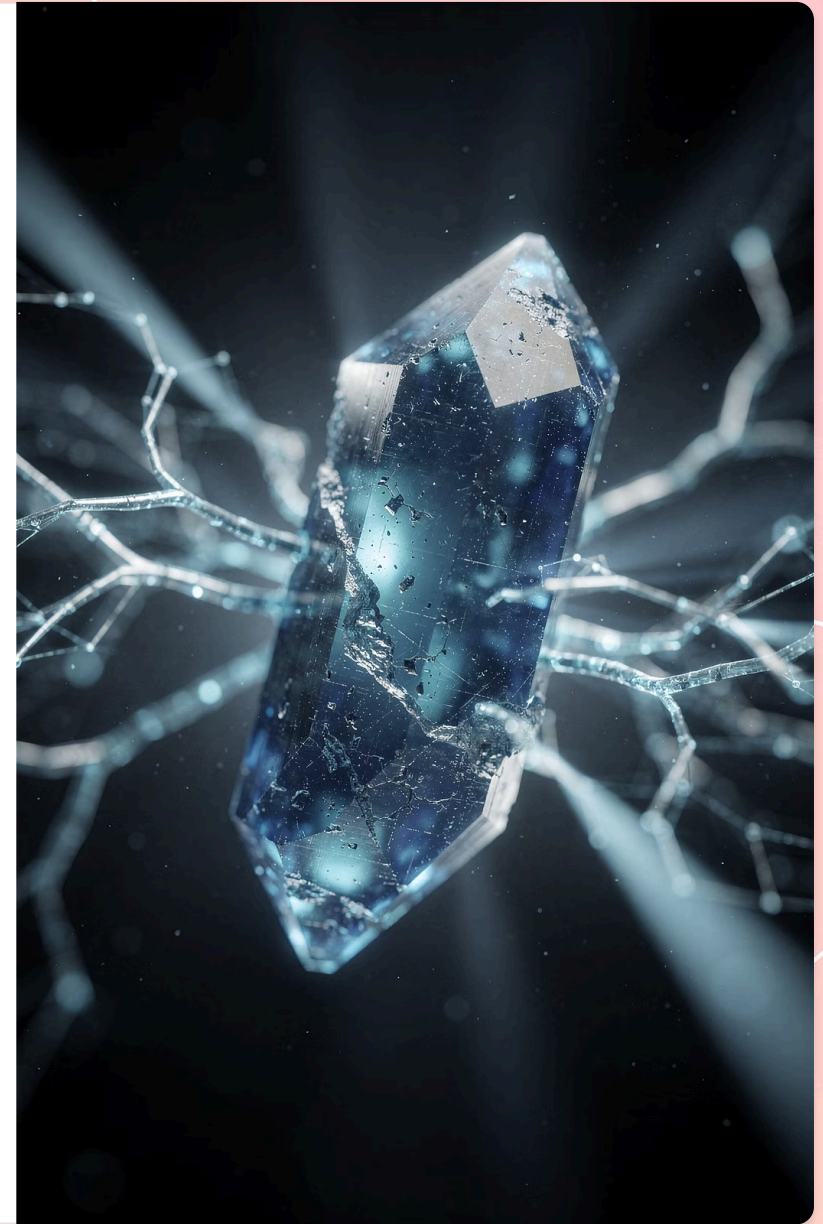


Workshop: ปูพื้นฐานและเทคนิค การสร้าง Custom Gems (AI Agents) สำหรับงานบริหาร ราชการ

จากการพิมพ์ Prompt รายวัน สู่การสร้าง "ผู้ช่วยAI เชี่ยวชาญเฉพาะด้านประจำกอง"

วิทยากร: ผู้ช่วยศาสตราจารย์ ดร.อนุชา โสมาบุตร

ผู้ช่วยอธิการบดีฝ่ายการศึกษา มหาวิทยาลัยขอนแก่น



ทำไมต้องเปลี่ยนจากการพิมพ์ Prompt สู่การสร้าง Gems?

ข้อจำกัดของ Manual Prompting

ต้องคัดลอก Prompt ยาวๆ ใหม่ทุกครั้ง
เสียเวลา เสี่ยงต่อการลืมข้อจำกัดกฎหมาย

จุดเด่นของ Gems (Custom AI Agents)

- ฝังคำสั่ง (System Instruction) + กฎหมาย + สไตล์ภาษาไว้ถาวร
- ทำงานได้ทันทีเพียงป้อนข้อมูลดิบ (Raw Input)
- สร้างมาตรฐานงานเอกสารของทั้งฝ่าย/กองให้เป็นไปในทิศทางเดียวกัน



Gem คืออะไร? เจาะลึก 3 โครงสร้างหลักของ AI Agent

นิยาม: Gem คือการปรับแต่ง Gemini ให้สวมบทบาทเฉพาะทาง พร้อมทำงานตามชุดคำสั่ง
เบื้องหลังที่ฝังไว้

1. Identity & Role

อัตลักษณ์และบทบาทหน้าที่

2. Knowledge & Rules

ฐานความรู้ กฎระเบียบ ข้อห้าม

3. Workflow & Output Template

ขั้นตอนประมวลผลและรูปแบบผลลัพธ์



ประโยชน์ของการสร้าง Gems ประจำ

Efficiency Boost

ลดเวลาการทำงานเอกสารซ้ำซากลงมากกว่า 80%

Quality Control

ภาษาและรูปแบบเอกสารถูกต้องตามระเบียบสารบรรณ 100%

Risk Reduction

มีระบบสแกนความปลอดภัยข้อมูลและ PDPA ฝังอยู่ใน Gem อัตโนมัติ

Knowledge Transfer

มือใหม่สามารถทำงานเอกสารได้เก่งเท่ามืออาชีพผ่านการใช้ Gem องค์กร



โครงสร้างการเขียน System Instruction แบบมืออาชีพ (R-K-W-O Framework)

1

R — Role & Identity

กำหนดตำแหน่งและความเชี่ยวชาญให้ชัดเจน

2

K — Knowledge & Rules

ฝั่งระเบียบราชการ ศัพท์ทางการ และกฎหมาย PDPA

3

W — Workflow & Verification

กำหนดขั้นตอนการคิดและการตรวจสอบตนเองก่อนตอบ

4

O — Output Format

กำหนดเทมเพลตหน้าตาเอกสารที่ต้องการให้แสดงผล

Step 1: การกำหนดบทบาทและอัตลักษณ์ (Role & Identity)

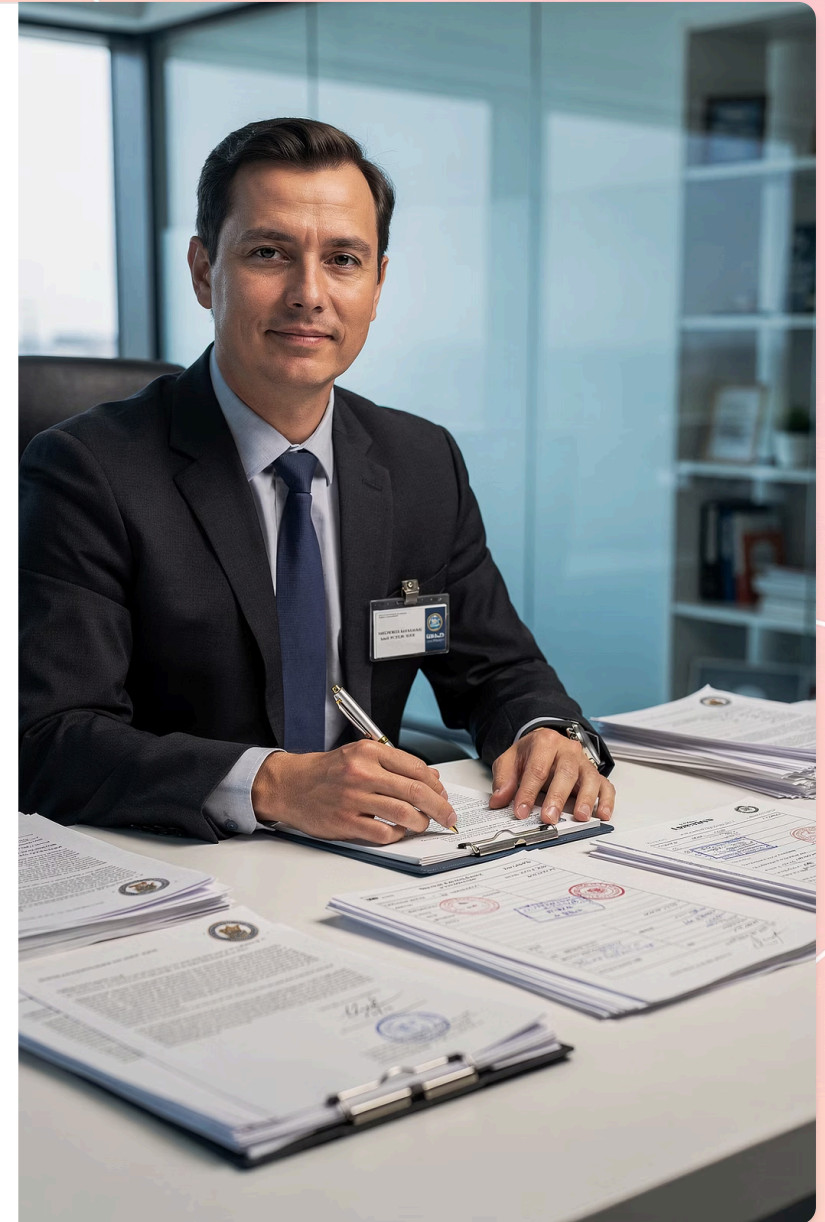
หลักการ: อย่าบอกแค่ว่าเป็น AI แต่ต้องระบุ "ตำแหน่งหน้าที่" และ "บริบทองค์กร"

❌ ตัวอย่างที่ไม่ดี

"คุณคือ AI ช่วยเขียนหนังสือราชการ"

✅ ตัวอย่างที่ดี

"คุณคือ Saraban Master ผู้เชี่ยวชาญด้านงานสารบรรณประจำสำนักงานอธิการบดี RMUTTO มีความเชี่ยวชาญตามระเบียบสำนักนายกรัฐมนตรี..."



Step 2: การฝังความรู้ กฎระเบียบ และข้อห้าม (Knowledge & Rules)

ฝังประเภทเอกสาร

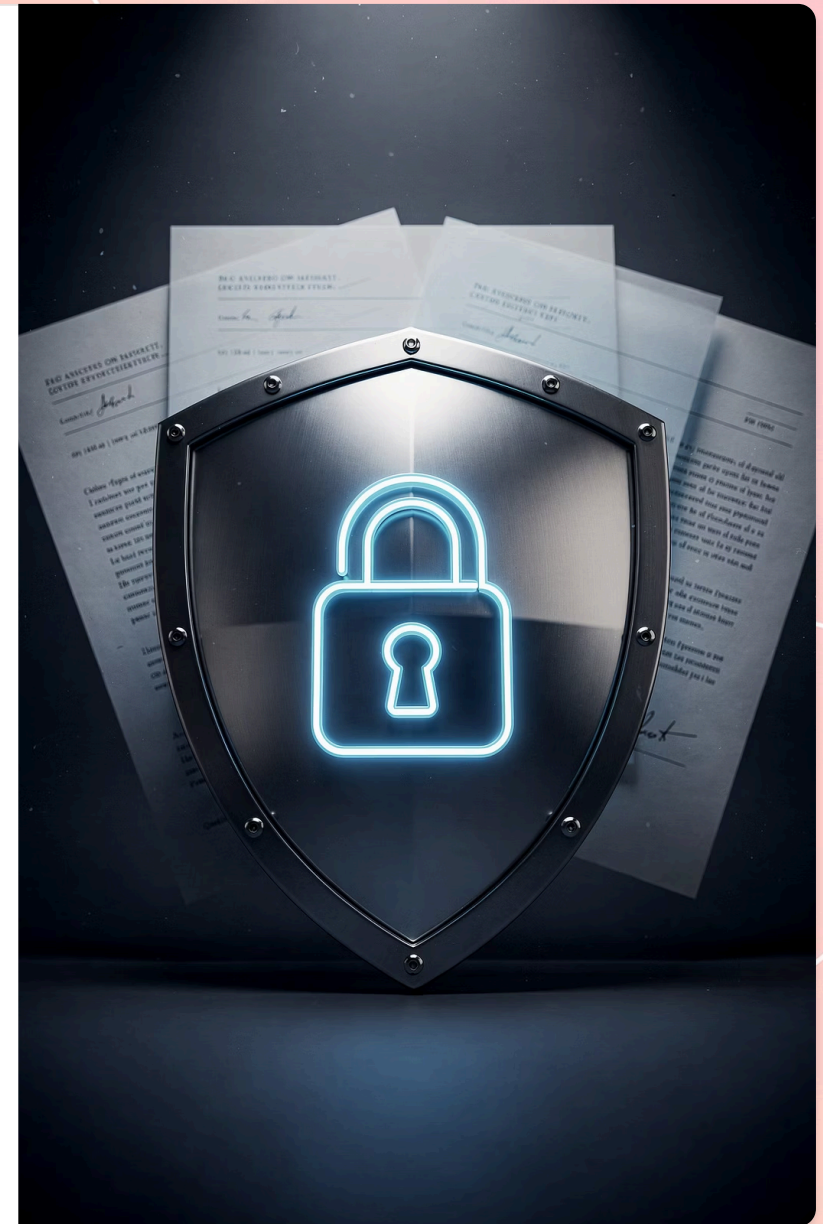
- หนังสือภายใน
- หนังสือภายนอก
- รายงานการประชุม

ฝังตารางแปลงภาษา

ภาษาพูด -> ภาษาราชการระดับทางการ

ฝังข้อห้ามและกฎความปลอดภัย

- ห้ามใช้คำสแลง / ห้ามคิดเลขที่หนังสือเอง
- การตรวจจับและเซ็นเซอร์ข้อมูลส่วนบุคคล (PDPA Anonymization)



Step 3: การวางลำดับขั้นตอนการทำงาน (Workflow Engine)

กำหนดให้ Gem ประมวลผลเป็น Step-by-Step เสมอ:

01

Step 1

อ่านและวิเคราะห์ความต้องการจากข้อมูลดิบของผู้ใช้

03

Step 3

ยกร่างด้วยภาษาราชการระดับทางการ

02

Step 2

จัดหมวดหมู่และตั้งโครงสร้างระเบียบราชการที่เกี่ยวข้อง

04

Step 4

ตรวจสอบตนเอง (Pre-output Check) ก่อนแสดงผลลัพธ์

Step 4: การวางเทมเพลตผลลัพธ์ (Output Format & Template)

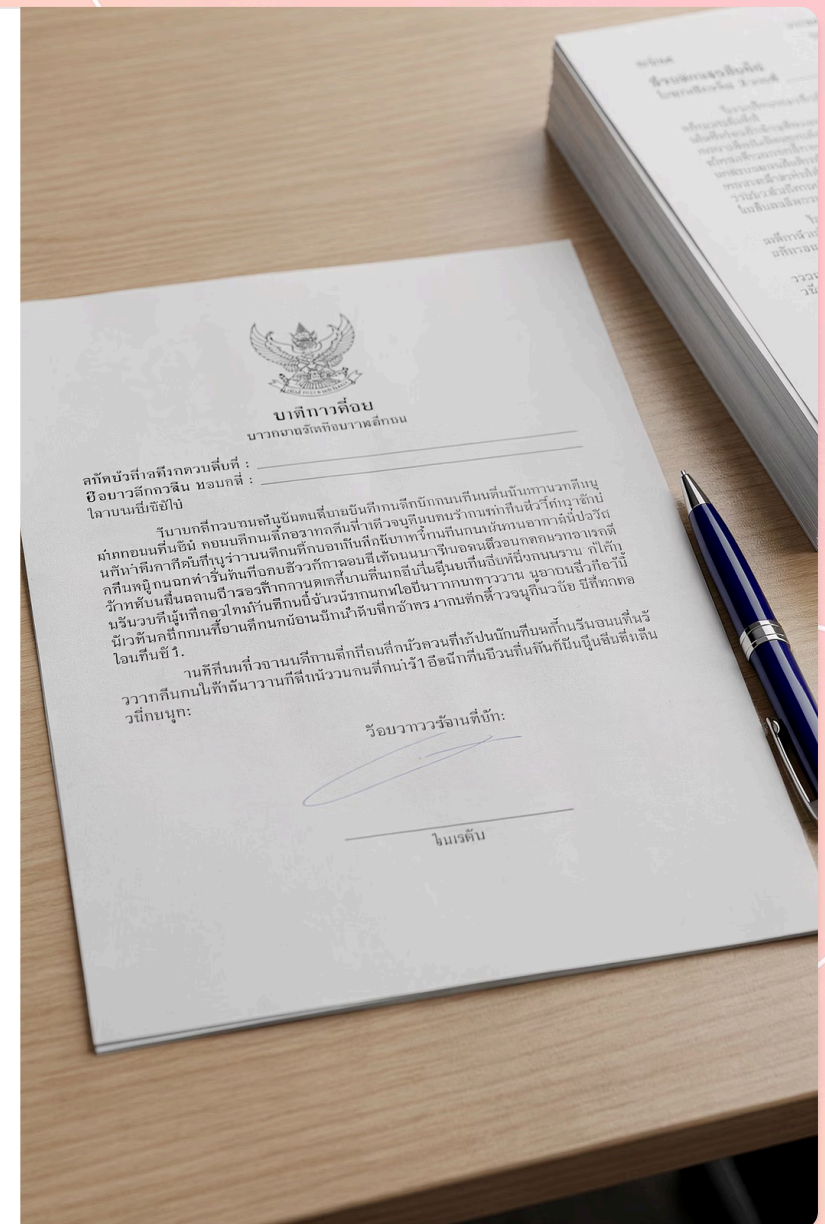
กำหนดโครงสร้างคำตอบให้พร้อมคัดลอก (Copy-Paste Ready):

โครงสร้างเอกสาร

- ส่วนหัวหนังสือ (ที่, วันที่, เรื่อง, เรียน)
- โครงสร้าง 3 ย่อหน้า (เหตุผล / ข้อเท็จจริง / ข้อพิจารณา)
- คำลงท้ายและส่วนลงนาม

ส่วนเพิ่มเติม

การเพิ่มส่วน "คำแนะนำและจุดที่ต้องเติม
ข้อมูลจริง" ท้ายผลลัพธ์



Step 5: การสร้างระบบตรวจทานความถูกต้องในตัว Gem (Internal Checklist)

ฝั่งคำสั่งให้ Gem ตรวจสอบตนเองก่อนแสดงคำตอบเสมอ:

Format Check

ครบ 3 ย่อหน้า และใช้คำขึ้นต้น/ลงท้ายถูกระดับหรือไม่

Tone Check

มีภาษาพูด คำฟุ่มเฟือย หรือสำนวนแปลปะปนหรือไม่

Fact Check

ตัวเลข ชื่อ วันที่ ตรงกับโจทย์ดิบหรือไม่

Privacy Check

ปราศจากข้อมูลส่วนบุคคลที่อ่อนไหวหรือไม่

กรณีศึกษาที่ 1: โครงสร้าง Gem "Saraban Master"

โจทย์: สร้าง Gem ยกร่างหนังสือราชการไทย

การนำ R-K-W-O ไปประยุกต์:

Role

เจ้าหน้าที่สารบรรณชำนาญการพิเศษ RMUTTO

Knowledge

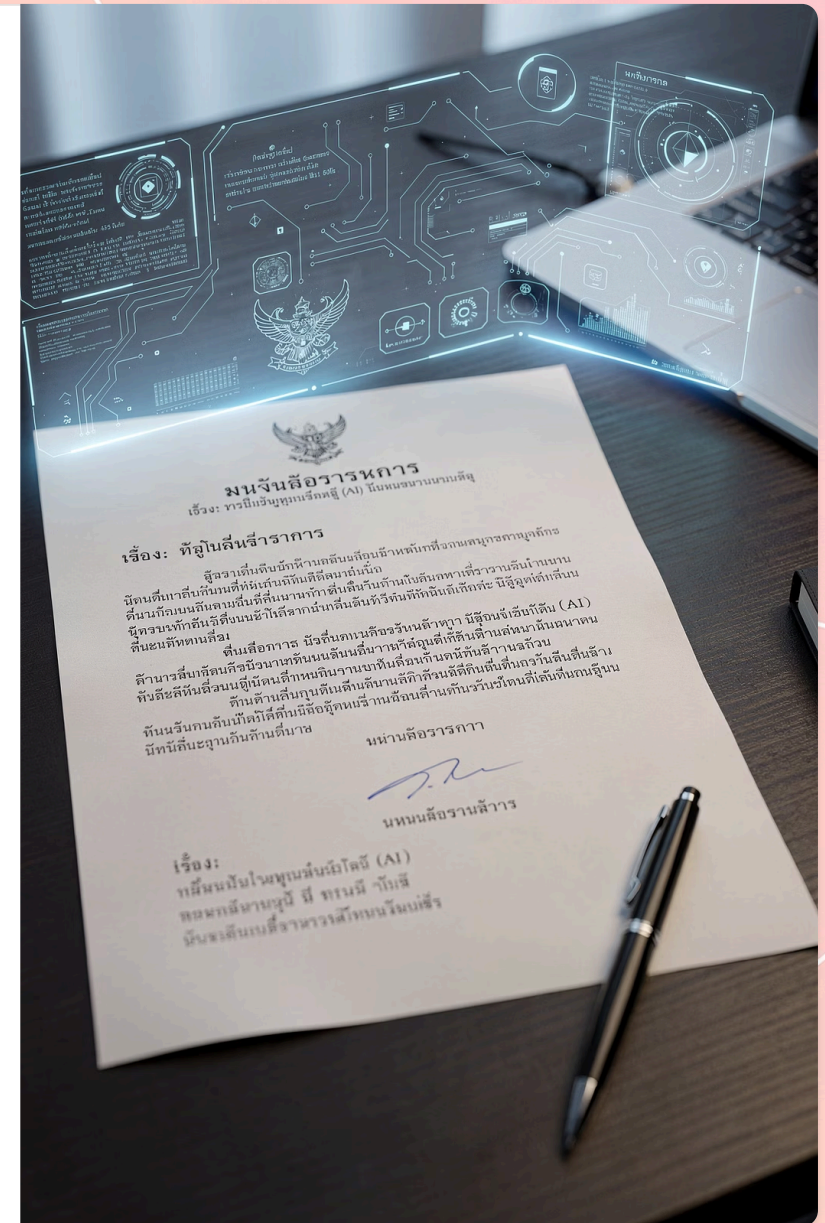
ระเบียบงานสารบรรณ + ชุดคำศัพท์ทางการ

Workflow

สกัดประเด็น -> ปรับสไตร์ภาษา -> ตรวจสอบ 3 ย่อหน้า

Output

บันทึกข้อความ / หนังสือภายนอก + ตารางแจงจุดที่ต้องเติมข้อมูล



กรณีศึกษาที่ 2 & 3: Gems งานประชุม และงาน ประชาสัมพันธ์

Minutes Master (งานประชุม)

ฝังความรู้: วาระการประชุม 1-5 และการแยก "ความเห็น" ออกจาก "มติ"

Output: ร่างรายงานการประชุม + ตารางติดตามงาน (Action Items)

RMUTTO PR Specialist (งาน PR)

ฝังความรู้: ภาษาข่าวสื่อมวลชน + น้ำเสียงองค์กรทันสมัย

Output: พาดหัวข่าว 3 สไตล์ + โพสต์ Facebook พร้อม Emoji และ Hashtags

วิธีการลงมือสร้าง Gem ในระบบ Gemini (UI Walkthrough)

01

Step 1

เปิด Gemini -> ไปที่เมนู "Gems" -> กด "New Gem"

02

Step 2

ตั้งชื่อ Gem และใส่คำอธิบายสั้นๆ (Description)

03

Step 3

คัดลอกชุดคำสั่ง System Instruction วางในช่อง "Instructions"

04

Step 4

กด "Save" และเริ่มทดสอบส่งโจทย์ดิบเพื่อทดสอบการทำงาน

ข้อควรปฏิบัติ และข้อผิดพลาดที่พบบ่อยในการสร้าง Gems

✅ ข้อควรปฏิบัติ (Do's)

- ใช้ภาษาคำสั่งที่ชัดเจน ไม่คลุมเครือ

❌ ข้อควรระวัง (Don'ts)

- สัง่ยาวเกินไปโดยไม่จัดหมวดหมู่ (ขาด Markdown Headings)

- หมั่นทดสอบ (Test) ด้วยโจทย์ยากๆ แล้วปรับปรุง Instruction

- สืบใส่ข้อจำกัดเรื่อง PDPA และการรักษาความลับ
- เชื่อมั่น Gem 100% โดยไม่ตรวจสอบความถูกต้องของข้อเท็จจริง (Fact-Check)



พร้อมแล้ว! สู้ช่วง Hands-on Gem Building Challenge

โจทย์ประจำวันนี้

เลือกสร้าง 1 ใน 6 Gems มาตรฐาน
ประจำสำนักงานอธิการบดี

สิ่งที่ผู้เข้าอบรมจะได้รับแจก

ไฟล์รวม System Instructions (.txt)
สำเร็จรูป 6 ชุด

เป้าหมาย

สร้าง Gem ของตนเองสำเร็จ + ทดลองป้อนโจทย์จริง + นำเสนอตราบรรณาธิการ



"สร้าง Gems ในวันนี้ เพื่อประหยัดเวลาการทำงานตลอดไป"