

(ฉบับนักพัฒนา)

Markdown

เขียนเอกสารอย่างมืออาชีพด้วยภาษาที่เรียบง่าย



อนุชิต ชาญ

สำนักพิมพ์ทิวลิป

Markdown ฉบับนักพัฒนา

เขียนเอกสารอย่างมืออาชีพด้วยภาษาที่เรียบง่าย

อนุชิต ชโลธร

คำนำ

ในโลกของนักพัฒนา ไม่ว่าจะเป็นการเขียนโค้ด การสื่อสารกับทีม หรือการส่งมอบงานให้ลูกค้า “เอกสาร” คือสิ่งที่หลีกเลี่ยงไม่ได้ และ Markdown ก็กลายเป็นภาษายอดนิยมที่ใช้กันอย่างแพร่หลายในงานเหล่านี้ เพราะมันเขียนง่าย อ่านง่าย และรองรับการแสดงผลในหลายแพลตฟอร์ม

ไม่ว่าคุณเคยเห็นไฟล์ README.md ใน GitHub, บันทึก API สำหรับทีม, หรือสไลด์ที่เขียนด้วย Marp — ทั้งหมดนี้ล้วนขับเคลื่อนด้วย Markdown

หนังสือเล่มนี้ถูกเขียนขึ้นเพื่อให้นักพัฒนาเข้าใจ Markdown อย่างเป็นระบบ โดยไม่จำเป็นต้องมีพื้นฐานมาก่อน คุณจะได้เรียนรู้ตั้งแต่โครงสร้างพื้นฐาน การจัดรูปแบบ การจัดการโค้ด การใช้งาน Markdown ในโปรเจกต์จริง รวมไปถึงการนำไปใช้กับ GitHub, Static Site Generator และระบบ DevOps

เป้าหมายของหนังสือนี้ไม่ใช่แค่ให้คุณ “ใช้ Markdown เป็น” แต่เพื่อให้คุณสามารถ **สื่อสารด้วยเอกสารได้ดีขึ้น** ในฐานะนักพัฒนามืออาชีพ

สารบัญ

แนะนำ Markdown	1
Markdown คืออะไร	1
ทำไม Markdown ถึงเป็นที่นิยมในหมู่นักพัฒนา	1
เปรียบเทียบ Markdown กับภาษา Markup อื่น ๆ	2
พื้นฐานของ Markdown	4
การติดตั้งและเครื่องมือที่ใช้เขียน	4
โครงสร้างทั่วไปของ Markdown	4
การจัดรูปแบบข้อความ	6
Heading (หัวข้อ)	6
ตัวหนา (Bold) และตัวเอียง (Italic)	6
Blockquote (ข้อความอ้างอิง)	7
รายการแบบลำดับ / ไม่ลำดับ (List)	7
รายการไม่ลำดับ (Unordered List)	7
รายการลำดับ (Ordered List)	8
การใส่ลิงก์และรูปภาพ	8
ลิงก์	8
รูปภาพ	9
การจัดการโค้ดใน Markdown	10
การใช้ inline code	10

การใช้ code block พร้อมการไฮไลต์ภาษา	10
การใส่คำอธิบายโค้ด	12
คำอธิบายก่อนหรือหลังโค้ด	12
คำอธิบายภายในโค้ด	12
เทคนิคการใส่โค้ดใน Markdown	13
การจัดระเบียบเอกสาร	14
การใช้ตาราง (Tables)	14
การใส่เส้นคั่น (Horizontal Rules)	15
การใส่หมายเหตุและเชิงอรรถ (Footnotes)	15
การแบ่งเอกสารด้วย Section และ Anchor	16
Markdown กับ GitHub	19
GitHub Flavored Markdown (GFM) คืออะไร	19
โครงสร้าง Repository	20
การเขียน README.md	21
การสร้าง CHANGELOG.md, CONTRIBUTING.md	22
CHANGELOG.md	23
CONTRIBUTING.md	23
การสร้าง Issue Templates และ Pull Request Templates	23
ตัวอย่าง .github/ISSUE_TEMPLATE/bug_report.md	24
ตัวอย่าง .github/PULL_REQUEST_TEMPLATE.md	24
เคล็ดลับการทำโปรเจกต์ให้น่าร่วมพัฒนา	25

Markdown กับการทำเอกสาร	26
ใช้ Markdown เขียนเอกสาร API	26
การใช้ Markdown กับ OpenAPI / Swagger	27
การใช้ Mermaid.js วาดไดอะแกรมใน Markdown	28
การสร้างสไลด์ด้วย Marp (Markdown Presentation)	30
Syntax ของ Marp	30
การเพิ่มธีมและ Custom Style	31
การ Export เป็น PDF / HTML	33
การสร้างเอกสารแบบ PDF / HTML ด้วย Pandoc	33
การใช้ Markdown กับเครื่องมือต่างๆ	34
Markdown กับ Static Site Generator	36
Jekyll	36
จุดเด่น	37
เหมาะกับ	37
Hugo	37
จุดเด่น	38
เหมาะกับ	38
Docusaurus	39
จุดเด่น	39
ฟีเจอร์พิเศษ	40
เหมาะกับ	40

สรุปเปรียบเทียบ	40
การสร้างเอกสาร DevDoc หรือ Knowledge Base	41
DevDoc (เอกสารสำหรับนักพัฒนา)	41
Knowledge Base (คลังความรู้)	41
เคล็ดลับการสร้างเอกสารด้วย SSG	42
การใช้ Markdown ในโลกของ DevOps และ AI	44
ใช้ Markdown ใน CI/CD Pipeline	44
การเขียน README เพื่อใช้กับ GitHub Actions	45
ตัวอย่าง README สำหรับ GitHub Actions	45
การเขียน Prompt และบันทึกผลลัพธ์การทดลอง AI	46
การจัดการไฟล์ Markdown	49
การจัดโครงสร้างไฟล์ Markdown	49
การตั้งชื่อไฟล์และลิงก์ที่เป็นมิตรกับ Git	50
การดูแล Markdown ให้ใช้งานได้นาน	51

แนะนำ Markdown

Markdown คืออะไร

Markdown คือภาษามาร์กอัป (markup language) ที่ออกแบบมาให้ **อ่านง่ายและเขียนง่าย** โดยใช้สัญลักษณ์ธรรมดาจากแป้นพิมพ์ เช่น #, *, หรือ - เพื่อจัดรูปแบบข้อความ เช่น หัวข้อ ตัวหนา ตัวเอียง ลิงก์ ตาราง หรือโค้ด

Markdown คิดค้นโดย **John Gruber** ร่วมกับ **Aaron Swartz** ในปี 2004 โดยมีเป้าหมายให้คนทั่วไปสามารถเขียนเอกสารที่อ่านได้สะดวกบน plain text และสามารถแปลงเป็น HTML ได้โดยอัตโนมัติ

Markdown เป็นภาษาที่ “ไม่ต้องเรียนก็พอจะเดาได้” เพราะมีความเรียบง่าย ไม่มีโครงสร้างซับซ้อน จึงได้รับความนิยมอย่างรวดเร็วในวงการนักเขียนบล็อกเกอร์ และโดยเฉพาะนักพัฒนา

ทำไม Markdown ถึงเป็นที่นิยมในหมู่นักพัฒนา

นักพัฒนาจำเป็นต้องเขียนเอกสารจำนวนมาก เช่น README, API doc, CHANGELOG, Wiki, หรือแม้กระทั่งสไลด์ฟรีเซนต์งาน

Markdown ตอบโจทย์นักพัฒนาในหลายมิติ

- **เขียนง่ายและอ่านง่าย** บน text editor ใด ๆ โดยไม่ต้องพึ่งเครื่องมือพิเศษ
- **แสดงผลได้ดี** บนแพลตฟอร์มยอดนิยม เช่น GitHub, GitLab, VS Code

- เป็นมิตรกับ **Git** เพราะเป็น plain text จึง diff และ merge ได้ง่าย
- รองรับการแสดงผลอัตโนมัติ บน Static Site Generator เช่น Jekyll, Docusaurus
- มีไลบรารีและเครื่องมือเสริมมากมาย ทั้ง CLI และ UI เช่น Pandoc, Typora, Obsidian, Marp

ในโลกของการทำงานแบบ DevOps, Docs-as-Code และ Remote Work การเขียนเอกสารด้วย Markdown กลายเป็นทักษะพื้นฐานที่ “จำเป็นต้องมี” สำหรับนักพัฒนา

เปรียบเทียบ Markdown กับภาษา Markup อื่น ๆ

คุณสมบัติ	Markdown	HTML	LaTeX	reStructuredText
ความง่ายในการเขียน	ง่ายมาก อ่านออก แม้ไม่รู้ syntax	ซับซ้อน ต้องจำ tag	ซับซ้อน มี syntax เฉพาะ	ค่อนข้างซับซ้อน
ความสามารถในการแสดงผล	ดีพอใช้ แสดงผลได้ทั่วไป	ยืดหยุ่นสูง ควบคุมได้ละเอียด	เหมาะกับเอกสารวิชาการ	เหมาะกับเอกสารเทคนิค
ใช้งานในงานนักพัฒนา	เหมาะสม ใช้งานจริงอย่างแพร่หลาย	ใช้ในบางกรณี เช่นอีเมลหรือเว็บ	ใช้เฉพาะทาง (วิชาการ, คณิต)	ใช้กับโปรเจกต์ Python บางส่วน
รองรับใน Dev Tools	รองรับเกือบทุกแพลตฟอร์ม	รองรับบางส่วน เช่น HTML preview	แทบไม่รองรับ	รองรับบางแพลตฟอร์มเฉพาะทาง

จากตารางเปรียบเทียบจะเห็นว่า Markdown โดดเด่นเรื่องความเรียบง่ายในการเขียนและเหมาะสมกับงานของนักพัฒนา ทั้งยังได้รับการสนับสนุนจาก