

เอกสารประกอบการเรียนรายวิชา เทคโนโลยี (วิทยาการคำนวณ)

ระดับชั้น: มัธยมศึกษาปีที่ 1 สาระที่ 4: เทคโนโลยี มาตรฐาน ว 4.2: เข้าใจและใช้แนวคิดเชิงคำนวณในการแก้ปัญหาที่พบในชีวิตจริงอย่างเป็นขั้นตอนและเป็นระบบ ใช้เทคโนโลยีสารสนเทศและการสื่อสารในการเรียนรู้ การทำงาน และการแก้ปัญหาได้อย่างมีประสิทธิภาพ รู้เท่าทัน และมีจริยธรรม ตัวชี้วัด: ว 4.2 ป.2/2 เขียนโปรแกรมอย่างง่าย โดยใช้ซอฟต์แวร์หรือสื่อ และตรวจหาข้อผิดพลาดของโปรแกรม (บททวนและประยุกต์ใช้ในบริบทชีวิตประจำวันระดับมัธยมศึกษา)

เรื่อง: การเขียนโปรแกรมอย่างง่ายในชีวิตประจำวัน

1. การเขียนโปรแกรมคืออะไร?

หลายคนอาจคิดว่า "การเขียนโปรแกรม" (Programming) ต้องเป็นการพิมพ์โค้ดภาษาอังกฤษยาวๆ ที่ดูซับซ้อนบนหน้าจอคอมพิวเตอร์สีดำเท่านั้น แต่ในความเป็นจริง การเขียนโปรแกรม คือ การเขียนลำดับขั้นตอนคำสั่ง เพื่อให้คอมพิวเตอร์หรือมนุษย์ทำงานตามที่เราต้องการเพื่อแก้ปัญหาใดปัญหาหนึ่งอย่างถูกต้องและแม่นยำ

ลองจินตนาการว่าคอมพิวเตอร์เป็นเหมือนหุ่นยนต์ที่ทำงานได้รวดเร็วและไม่รู้จักเหน็ดเหนื่อย แต่ข้อเสียคือ "มันคิดเองไม่ได้" มันจะทำตามคำสั่งที่เราป้อนให้เท่านั้น หากเราสั่งงานไม่ละเอียดหรือข้ามขั้นตอน คอมพิวเตอร์ก็จะทำงานผิดพลาดทันที

ในชีวิตประจำวัน เราใช้ "แนวคิดการเขียนโปรแกรม" อยู่ตลอดเวลาโดยที่เราไม่รู้ตัว นั่นคือการคิดอย่างเป็น "ขั้นตอน" (Algorithm - อัลกอริทึม) ซึ่งหมายถึงกระบวนการแก้ปัญหาที่สามารถอธิบายออกมาเป็นขั้นตอนที่ชัดเจน ตั้งแต่จุดเริ่มต้น ไปจนถึงผลลัพธ์ที่ต้องการ

2. ตัวอย่าง "โปรแกรม" ในชีวิตประจำวัน

เพื่อให้เห็นภาพชัดเจนขึ้น ลองนึกภาพว่าคุณกำลังเป็น โปรแกรมเมอร์ที่ต้องเขียนคำสั่งให้หุ่นยนต์พ่อบ้านทำงานพื้นฐานในบ้าน หากคำสั่งไม่ละเอียด หุ่นยนต์อาจจะทำผิดพลาดได้ มาดูตัวอย่างการเขียนโปรแกรมในชีวิตจริงกัน:

ตัวอย่างที่ 1: โปรแกรมทอดไข่เจียว (การทำงานแบบเรียงลำดับ - Sequence)

- ขั้นตอนที่ 1: หยิบไข่ไก่ 2 ฟองจากตู้เย็น
- ขั้นตอนที่ 2: ตอกไข่ใส่ชาม (ระวังอย่าให้เปลือกไข่ตกลงไป)
- ขั้นตอนที่ 3: ใส่เครื่องปรุง (ซีอิ้วขาว 1 ช้อนชา)
- ขั้นตอนที่ 4: ตีไข่ให้เข้ากันจนเกิดฟอง
- ขั้นตอนที่ 5: ตั้งกระทะบนเตาและเปิดไฟระดับปานกลาง
- ขั้นตอนที่ 6: ใส่น้ำมันพืชลงในกระทะ รอประมาณ 1 นาทีจนน้ำมันร้อน
- ขั้นตอนที่ 7: เทไข่ลงในกระทะ
- ขั้นตอนที่ 8: ทอดและพลิกด้านจนสุกเหลืองทั้งสองด้าน
- ขั้นตอนที่ 9: ปิดเตา ตักไข่ใส่จาน และพร้อมเสิร์ฟ


ตัวอย่างที่ 2: โปรแกรมเตรียมตัวก่อนออกจากบ้าน (การทำงานแบบมีเงื่อนไข - **Condition**) นอกจากการทำตามลำดับแล้ว การเขียนโปรแกรมยังมีการสร้าง "เงื่อนไข" (If-Else) เพื่อให้ระบบตัดสินใจตามสถานการณ์ที่เปลี่ยนไป:

- **ขั้นตอนที่ 1:** อาบน้ำและแต่งตัวชุดนักเรียน
- **ขั้นตอนที่ 2:** รับประทานอาหารเช้า
- **ขั้นตอนที่ 3:** ตรวจสอบสภาพอากาศนอกหน้าต่าง
- **ขั้นตอนที่ 4:** ถ้า (**If**) ฝนตก แล้ว ให้หยิบร่มใส่กระเป๋า
- **ขั้นตอนที่ 5:** มิฉะนั้น (**Else**) (ถ้าฝนไม่ตก) แล้ว ไม่ต้องหยิบร่ม
- **ขั้นตอนที่ 6:** สวมรองเท้าและเดินทางไปโรงเรียน

นี่คือการเขียนโปรแกรมอย่างง่ายแบบไม่มีคอมพิวเตอร์ (Unplugged Coding) ซึ่งช่วยฝึกให้เราเป็นคนคิดอย่างมีระบบ มีเหตุผล และรอบคอบมากขึ้น

3. การหาข้อผิดพลาดของโปรแกรม (Debugging)

ในโลกของการเขียน โปรแกรม ข้อผิดพลาดต่างๆ เราจะเรียกว่า "**บั๊ก (Bug)**" และกระบวนการค้นหาและแก้ไขข้อผิดพลาดนั้นเรียกว่า "**ดีบั๊ก (Debugging)**"

 **เกร็ดความรู้:** คำว่า *Bug* ที่แปลว่า "แมลง" ถูกนำมาใช้ในวงการคอมพิวเตอร์จริงๆ เมื่อปี ค.ศ. 1947 โดย เกรซ ฮอปเปอร์ (Grace Hopper) นักวิทยาศาสตร์คอมพิวเตอร์ ที่พบว่าเครื่องคอมพิวเตอร์ทำงานผิดพลาดเพราะมี "แมลงเม่า" ตัวเป็นๆ เข้าไปติดอยู่ในแผงวงจรจริงๆ!

ตัวอย่างสถานการณ์: สมมติว่าใน "โปรแกรมทอดไข่เจียว" ด้านบน หุ่นยนต์ทำตามแล้วปรากฏผลลัพธ์ที่ไม่ต้องการ เราต้องกลับมาตรวจสอบ (Debug) ขั้นตอนของเราที่ละบรรทัดว่าผิดพลาดที่จุดไหน:

- **ปัญหา:** ไข่เจียวเค็มเกินไปจนกินไม่ได้
 - **ตรวจหาข้อผิดพลาด (Debug):** ไล่อ่านคำสั่งตั้งแต่ขั้นตอนที่ 1 ลงมา พบว่าที่ขั้นตอนที่ 3 สั่งให้ใส่เครื่องปรุง (ซีอิ๊วขาว 5 ช้อนโต๊ะ) ซึ่งปริมาณมากเกินไป
 - **การแก้ไข:** เปลี่ยนขั้นตอนที่ 3 เป็น (ซีอิ๊วขาว 1 ช้อนชา)
- **ปัญหา:** หุ่นยนต์เทไข่ลงกระทะแล้วน้ำมันกระเด็นเลอะเทอะ หรือไข่ติดกระทะ
 - **ตรวจหาข้อผิดพลาด (Debug):** อาจเกิดจากการข้ามขั้นตอน หรือเรียงลำดับผิด เช่น ดันไปทำขั้นตอนที่ 7 (เทไข่) ก่อนขั้นตอนที่ 5 และ 6 (ตั้งกระทะและรอให้น้ำมันร้อน) กระทะที่ยังไม่ร้อนจะทำให้ไข่ติด และถ้าน้ำมันยังไม่ร้อนจัดแล้วเทน้ำลงไปก็จะกระเด็น
 - **การแก้ไข:** จัดเรียงลำดับคำสั่งใหม่ให้ถูกต้องตามหลักความเป็นจริง

การฝึก Debugging เป็นทักษะที่สำคัญมาก ไม่ใช่แค่ในคอมพิวเตอร์ แต่รวมถึงการแก้ปัญหาในชีวิตจริง เมื่อเกิดปัญหา เราต้องไม่ตื่นตระหนก แต่ค่อยๆ ย้อนกลับไปหาสาเหตุและแก้ไขให้ตรงจุด

4. ซอฟต์แวร์และสื่อสำหรับการเขียนโปรแกรมอย่างง่าย

เมื่อเราเข้าใจหลักการเรียงลำดับขั้นตอนและเงื่อนไขแล้ว เราสามารถนำความรู้ไปประยุกต์ใช้กับซอฟต์แวร์หรือสื่อต่างๆ เพื่อสร้างผลงานจริงได้ เครื่องมือที่เหมาะสมสำหรับผู้เริ่มต้น ได้แก่:

- **Scratch:** โปรแกรมภาษาภาพ (Block-based programming) ที่นักเรียนไม่ต้องพิมพ์โค้ดเอง แต่ใช้การ "ลากแล้ววาง" (Drag and Drop) บล็อกคำสั่งสีสันสดใสมาต่อกันคล้ายจิ๊กซอว์ สามารถใช้สร้างเกมอินเทอร์แอคทีฟ นิทานแอนิเมชัน หรือศิลปะที่เคลื่อนไหวได้
- **Code.org:** เว็บไซต์ยอดนิยมทั่วโลกที่สอนการเขียนโปรแกรมผ่านเกมและตัวละครที่นักเรียนคุ้นเคย (เช่น Minecraft, Angry Birds) โดยมีบทเรียนแบบทีละขั้นตอน (Step-by-step) ช่วยให้เข้าใจการทำซ้ำ (Loop) และเงื่อนไขได้ง่ายขึ้น
- **Micro:bit:** บอร์ดไมโครคอนโทรลเลอร์ขนาดเล็ก ที่ทำให้นักเรียนเห็นผลลัพธ์ของการเขียนโปรแกรมในโลกจริง เช่น สั่งให้ไฟ LED ติด ส่งให้มီးเสียงเตือน หรือทำเป็นเครื่องวัดอุณหภูมิ
- **สื่อ Unplugged:** การเรียนรู้โดยไม่ใช้หน้าจอ เช่น การใช้บัตรคำสั่ง (Command Cards) เล่นเกมกระดานในห้องเรียน เพื่อหาเส้นทางที่สั้นที่สุดให้หุ่นยนต์เดินไปถึงเป้าหมาย

การเรียนรู้สิ่งเหล่านี้ไม่ได้มีเป้าหมายให้นักเรียนทุกคนต้องโตไปเป็นโปรแกรมเมอร์ แต่เพื่อสร้าง **"ทักษะการคิดเชิงคำนวณ" (Computational Thinking)** ซึ่งเป็นทักษะที่จำเป็นในศตวรรษที่ 21 ที่นำไปใช้ได้กับทุกสายอาชีพ

แบบฝึกหัดทบทวนความเข้าใจ

คำชี้แจง: ให้นักเรียนเขียน "โปรแกรม" ลำดับขั้นตอนการ **"ขงนมเย็น (นมชมพู)"** จำนวน 5-8 ขั้นตอน โดยให้นักเรียน **เพิ่มเงื่อนไข (If-Else)** เข้าไปในโปรแกรมด้วย 1 อย่าง (เช่น เงื่อนไขเกี่ยวกับความหวาน หรือการใส่น้ำแข็ง) พร้อมสมมติสถานการณ์ว่าเกิด **ข้อผิดพลาด (Bug)** ขึ้น 1 อย่าง และบอกวิธี **แก้ไข (Debug)**

1. ชื่อโปรแกรม:
2. ขั้นตอนการทำงาน:
 - 1.
 - 2.
 - 3.
 - 4.
 - 5.
 - 6.
 -

○ 7.

8.

3. ข้อผิดพลาดที่พบ (**Bug**): (เช่น ลูกค้าสั่งหวานน้อย แต่ได้หวานมาก หรือน้ำล้นแก้ว)
4. วิธีแก้ไข (**Debug**): (อธิบายว่าจะกลับไปแก้ไขที่ขั้นตอนใด และแก้ไขอย่างไร)

แหล่งอ้างอิง: ปรับปรุงและขยายเนื้อหาตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช 2551 (ฉบับปรับปรุง พ.ศ. 2560) สาระเทคโนโลยี (วิทยาการคำนวณ)