

## Computational Thinking

<https://www.youtube.com/watch?v=mUXo-S7gzds>

### ทักษะการคิดเชิงคำนวณ

Computational Thinking เป็นการคิดที่คล้ายกับความคิดของหุ่นยนต์ หรือการเขียนโปรแกรม มีประโยชน์อย่างมากมาย โดยเน้นไปที่ทักษะการคิดอย่างมีวิจารณญาณ การคิดอย่างเป็นเหตุเป็นผล หรืออาจจะเรียกว่าเป็นทักษะการแก้ปัญหาที่ลึกซึ้งมากขึ้น โดยจะมองปัญหา และแก้ปัญหอย่างเป็นระบบ โดยแนวทางการแก้ปัญหานั้นมนุษย์และคอมพิวเตอร์สามารถเข้าใจได้เหมือนกัน โดยถือเป็นกระบวนการขั้นสูงของการแก้ปัญหา computational Thinking แบ่งองค์ประกอบได้เป็น 4 วิธีคิด

- Decomposition
- Pattern recognition
- Abstraction
- Algorithm

Decomposition เป็นการแตกปัญหาที่ใหญ่หรือยุ่งยาก ให้เป็นปัญหาย่อยๆ เพื่อให้สามารถแก้ไขได้ง่าย

Pattern Recognition เป็นการมองเห็นปัญหา หรือรูปแบบของปัญหาที่แตกต่างกันแต่มีสิ่งที่มีความคล้ายคลึงกัน (ในรูปเป็นการมองเห็นรูปแบบของการที่ททำให้ยานพาหนะแต่ละแบบเคลื่อนที่ได้ด้วยความเร็วต้องมีส่วนของความโค้ง)

Abstraction เป็นการเลือกเฉพาะสิ่งที่สำคัญ เพื่อเห็น โครงสร้างโดยรวม

Algorithmic Thinking เป็นวิธีคิดที่เป็นลำดับขั้นตอน เช่น ทำสิ่งใดสิ่งหนึ่ง เราสามารถสร้างเป็นเช็ทของวิธีการ และคนอื่นสามารถใช้วิธีการของเราในการแก้ปัญหานั้นๆ ได้เหมือนกัน

มาดูตัวอย่างของปัญหาที่ใช้การคิดแบบ computational thinking

สมมติว่ารถของ max เกิดยางแบนขึ้นมา แต่โชคดีที่เค้าเป็นคนที่มีความคิดแบบ computational thinking เขาจะแก้ปัญหานี้อย่างไร

รถของ max ไม่สามารถเคลื่อนย้ายได้ ซึ่งรถเค้าตอนนี้มีปัญหาเกิดขึ้นสองอย่างคือ น้ำมันใกล้จะหมด และยางแบนไป 1 ล้อ

นั่นคือการคิดแบบ decomposition (แยกได้ว่ามีปัญหอะไรบ้าง) จากนั้น max เคยมีประสบการณ์เจอปัญหาลักษณะใกล้เคียงกันนี้มาก่อน เช่น รถน้ำมันหมด หรือยางแบน เค้าคิดว่าเค้าควรจะเปลี่ยนยางที่แบนก่อน นั่นคือการคิดแบบ pattern recognition (เคยเจอปัญหาในรูปแบบนี้มาก่อน คาดเดาได้ว่าควรแก้ด้วยวิธีการ

แบบใด) รถค่าน้ำมันเหลือน้อย แต่ก็ยังเพียงพอที่จะขับไปยังปั้มที่ใกล้ที่สุด แต่เค้าก็ตัดสินใจที่จะเปลี่ยนยาง ล้อที่แบนก่อน นั่นคือการคิดแบบ abstraction (เลือกทำในสิ่งที่สำคัญ และสามารถจำลองความคิดได้ว่าจาก ระยะทางที่อยู่ถึงปั้ม รถจะไปถึงได้หรือไม่) สุดท้ายการคิดแบบ algorithm คือการเปลี่ยนยางล้อ ต้องใช้ แม่แรงเพื่อยกรถขึ้น จากนั้นจึงค่อยเปลี่ยนล้อที่แบนออก แล้วเอาล้ออะไหล่เปลี่ยนแทน (คิดอย่างเป็นขั้นตอน ต้องทำอะไรบ้าง)

ทั้งหมดนี้คือ computational thinking ที่ไม่ใช่เฉพาะ โปรแกรมเมอร์เท่านั้นที่ต้องใช้วิธีคิดแบบนี้ แต่ในความเป็นจริง พวกเราทุกคนก็จำเป็นต้องคิดแบบนี้ไม่ว่าจะประกอบอาชีพใดๆ ก็ตาม

ทำไม computational thinking จึงสำคัญ ตามกฎของมัวร์ กล่าวถึงประสิทธิภาพของคอมพิวเตอร์ ที่จะสามารถประมวลผลได้อย่างรวดเร็วเพิ่มขึ้นเรื่อยๆ และมนุษย์ก็มีความจำเป็นต้องใช้คอมพิวเตอร์นี้เพื่อ อำนาจความสะดวกต่างๆ และช่วยแก้ปัญหาให้ได้อย่างมีประสิทธิภาพที่สุด เรามีเครื่องคอมพิวเตอร์และ อุปกรณ์คอมพิวเตอร์ต่างๆ เหล่านั้นเพื่อการใช้งาน แต่เราก็จำเป็นอย่างยิ่งที่จะต้องรู้ว่าทำอะไร เราถึงจะสั่ง ให้อุปกรณ์ต่างๆ เหล่านั้นทำงานหรือแก้ปัญหาตามความต้องการของเราได้ นั่นคือต้องใช้ทักษะการคิดแบบ computational thinking นั่นเอง