

ใบความรู้ที่ 1.1 เรื่อง ขั้นตอนการพัฒนาโปรแกรมและโปรแกรมจำลองแบบออนไลน์และออฟไลน์ หน่วยที่ 4
แผนการจัดการเรียนรู้ที่ 4 เรื่อง โปรแกรมเมอร์น้อย
รายวิชา เทคโนโลยี1 รหัส ว21103 ภาคเรียนที่ 1 ชั้นมัธยมศึกษาปีที่ 1

- จุดประสงค์**
1. อธิบายขั้นตอนการพัฒนาโปรแกรมและภาษาคอมพิวเตอร์ได้
 2. รู้จักและสามารถใช้โปรแกรมจำลองแบบออนไลน์และออฟไลน์ (Scratch, Kodu, Code.org) เบื้องต้นได้

ขั้นตอนการพัฒนาโปรแกรม

ในการเขียนโปรแกรมเพื่อแก้ปัญหาต่างๆ จำเป็นต้องมีการวางแผน และออกแบบโปรแกรมไว้ล่วงหน้า โดยกำหนดขั้นตอนวิธีการทำงานให้ชัดเจน ซึ่งกระบวนการวิเคราะห์ และออกแบบโปรแกรมเรียกว่า วัฏจักรการพัฒนาระบบงาน System Development Life Cycle (SDLC) ซึ่งมีกระบวนการทำงานเริ่มต้นจากการวิเคราะห์ปัญหาไปจนถึงการนำโปรแกรมไปใช้งาน และปรับปรุงพัฒนาระบบให้ดีขึ้น มีขั้นตอนของวัฏจักรการพัฒนาระบบงาน ดังต่อไปนี้

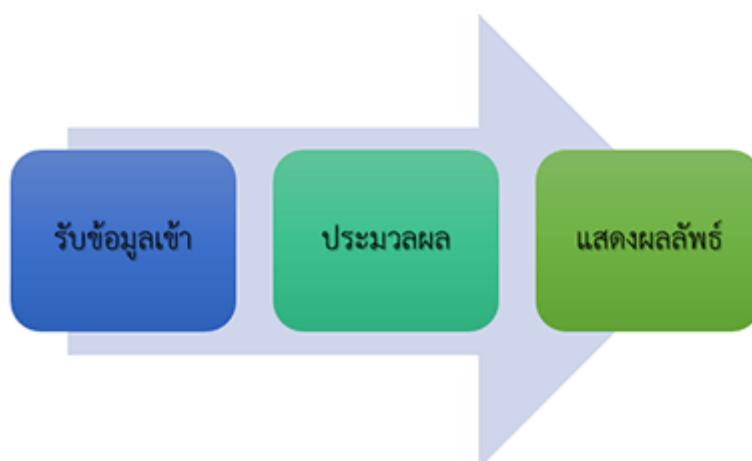
ขั้นตอนหรือวิธีการพัฒนาโปรแกรมประกอบด้วย 6 ขั้นตอนดังต่อไปนี้

1. การวิเคราะห์ปัญหา
2. การออกแบบโปรแกรม
3. การเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์
4. การทดสอบและแก้ไขโปรแกรม
5. การทำเอกสารประกอบโปรแกรม
6. การบำรุงรักษาโปรแกรม

1. การวิเคราะห์ปัญหา (Problem Analysis)

การวิเคราะห์ปัญหา ประกอบด้วยขั้นตอนต่างๆ ดังนี้

1. กำหนดวัตถุประสงค์ของงาน เพื่อพิจารณาว่าโปรแกรมต้องทำการประมวลผลอะไรบ้าง
2. พิจารณาข้อมูลนำเข้า (Input) เพื่อให้ทราบว่าต้องนำข้อมูลอะไรเข้าคอมพิวเตอร์ ข้อมูลมีคุณสมบัติเป็นอย่างไร ตลอดจนลักษณะและรูปแบบของข้อมูลที่จะนำเข้า
3. พิจารณาการประมวลผล (Process) เพื่อให้ทราบว่าโปรแกรมมีขั้นตอนการประมวลผลอย่างไรและมีเงื่อนไขการประมวลผลอะไรบ้าง
4. พิจารณาข้อมูลเสนอแนะ (Output) เพื่อให้ทราบว่าเมื่อมีข้อมูลอะไรที่จะแสดง ตลอดจนรูปแบบและสื่อที่จะใช้ใน การแสดงผล เช่นการแสดงผลทางจอภาพ การแสดงผลทางเครื่องพิมพ์



กระบวนการทำงานของคอมพิวเตอร์

2. การออกแบบโปรแกรม (Design)

การออกแบบขั้นตอนการทำงานของโปรแกรมเป็นขั้นตอนที่ใช้เป็นแนวทางในการลงรหัสโปรแกรม ผู้ออกแบบขั้นตอนการทำงานของโปรแกรมอาจใช้เครื่องมือต่างๆ ช่วยในการออกแบบ อาทิเช่น คำสั่งจำลอง (Pseudocode) หรือ ผังงาน (Flow chart) การออกแบบโปรแกรมนั้นไม่ต้องพะวงกับรูปแบบคำสั่งภาษาคอมพิวเตอร์ แต่ให้มุ่งความสนใจไปที่ลำดับขั้นตอนในการประมวลผลของโปรแกรมนั้น

3. การเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ (Programming)

การเขียนโปรแกรมเป็นการนำเอาผลลัพธ์ของการออกแบบโปรแกรม มาเปลี่ยนเป็นโปรแกรมภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง ผู้เขียนโปรแกรมจะต้องให้ความสนใจต่อรูปแบบคำสั่งและกฎเกณฑ์ของภาษาที่ใช้เพื่อให้การประมวลผลเป็นไปตามผลลัพธ์ที่ได้ออกแบบไว้ นอกจากนี้ผู้เขียนโปรแกรมควรแทรกคำอธิบายการทำงานต่างๆ ลงในโปรแกรมเพื่อให้โปรแกรมนั้นมีความกระจ่างชัดและง่ายต่อการตรวจสอบและโปรแกรมนี้อย่างยิ่งใช้เป็นส่วนหนึ่งของเอกสารประกอบ

4. การทดสอบและแก้ไขโปรแกรม

การทดสอบโปรแกรมเป็นการนำโปรแกรมที่ลงรหัสแล้วเข้าคอมพิวเตอร์ เพื่อตรวจสอบรูปแบบกฎเกณฑ์ของภาษา และผลการทำงานของโปรแกรมนั้น ถ้าพบว่ายังไม่ถูกต้องก็แก้ไขให้ถูกต้องต่อไป ขั้นตอนการทดสอบและแก้ไขโปรแกรม อาจแบ่งได้เป็น 3 ขั้นตอนต่อไปนี้

1. สร้างแฟ้มเก็บโปรแกรมซึ่งส่วนใหญ่นิยมนำโปรแกรมเข้าผ่านทางแป้นพิมพ์โดยใช้โปรแกรมประมวลผลคำ

2. ใช้ตัวแปลภาษาคอมพิวเตอร์แปลโปรแกรมที่สร้างขึ้นเป็นภาษาเครื่อง โดยระหว่างการแปลจะมีการตรวจสอบความถูกต้องของรูปแบบและกฎเกณฑ์ในการใช้ภาษา ถ้าคำสั่งใดมีรูปแบบไม่ถูกต้องก็จะแสดงข้อผิดพลาดออกมาเพื่อให้ผู้เขียนนำไปแก้ไขต่อไป ถ้าไม่มีข้อผิดพลาด เราจะได้โปรแกรมภาษาเครื่องที่สามารถให้คอมพิวเตอร์ประมวลผลได้

3. ตรวจสอบความถูกต้องของการประมวลผลของโปรแกรม โปรแกรมที่ถูกต้องตามรูปแบบและกฎเกณฑ์ของภาษา แต่อาจให้ผลลัพธ์ของการประมวลผลไม่ถูกต้องก็ได้ ดังนั้นผู้เขียนโปรแกรมจำเป็นต้องตรวจสอบว่าโปรแกรมประมวลผลถูกต้องตามต้องการหรือไม่ วิธีการหนึ่งก็คือ สมมติข้อมูลตัวแทนจากข้อมูลจริงนำไปให้โปรแกรมประมวลผลแล้วตรวจสอบผลลัพธ์ว่าถูกต้องหรือไม่ ถ้าพบว่าไม่ถูกต้องก็ต้องดำเนินการแก้ไขโปรแกรมต่อไป การสมมติข้อมูลตัวแทนเพื่อการทดสอบเป็นสิ่งที่มีความสำคัญเป็นอย่างมาก ลักษณะของข้อมูลตัวแทนที่ดีควรจะสมมติทั้งข้อมูลที่ถูกต้องและข้อมูลที่ผิดพลาด เพื่อทดสอบว่าโปรแกรมที่พัฒนาขึ้นสามารถครอบคลุมการปฏิบัติงานในเงื่อนไขต่างๆ ได้ครบถ้วน นอกจากนี้อาจตรวจสอบการทำงานของโปรแกรมด้วยการสมมติตัวเองเป็นคอมพิวเตอร์ที่จะประมวลผล แล้วทำตามคำสั่งที่ละคำสั่งของโปรแกรมนั้นๆ วิธีการนี้อาจทำได้ยากถ้าโปรแกรมมีขนาดใหญ่ หรือมีการประมวลผลที่ซับซ้อน

5. การทำเอกสารประกอบโปรแกรม

การทำเอกสารประกอบโปรแกรมเป็นงานที่สำคัญของการพัฒนาโปรแกรม เอกสารประกอบโปรแกรมช่วยให้ผู้ใช้โปรแกรมเข้าใจวัตถุประสงค์ ข้อมูลที่จะต้องใช้กับโปรแกรม ตลอดจนผลลัพธ์ที่จะได้จากโปรแกรม การทำโปรแกรมทุกโปรแกรมจึงควรต้องทำเอกสารกำกับ เพื่อใช้สำหรับการอ้างอิงเมื่อจะใช้งานโปรแกรมและเมื่อต้องการแก้ไขปรับปรุงโปรแกรม เอกสารประกอบโปรแกรมที่จัดทำ ควรประกอบด้วยหัวข้อต่อไปนี้

1. วัตถุประสงค์
2. ประเภทและชนิดของคอมพิวเตอร์และอุปกรณ์ที่ใช้ในโปรแกรม
3. วิธีการใช้โปรแกรม
4. แนวคิดเกี่ยวกับการออกแบบโปรแกรม
5. รายละเอียดโปรแกรม
6. ข้อมูลตัวแทนที่ใช้ทดสอบ
7. ผลลัพธ์ของการทดสอบ

6. การบำรุงรักษาโปรแกรม

เมื่อโปรแกรมผ่านการตรวจสอบตามขั้นตอนเรียบร้อยแล้ว และถูกนำมาให้ผู้ใช้ได้ใช้งาน ในช่วงแรกผู้ใช้ อาจจะยังไม่คุ้นเคยก็อาจทำให้เกิดปัญหาขึ้นมาบ้าง ดังนั้นจึงต้องมีผู้คอยควบคุมดูแลและคอยตรวจสอบการทำงาน การบำรุงรักษาโปรแกรมจึงเป็นขั้นตอนที่ผู้เขียนโปรแกรมต้องคอยเฝ้าดูและหาข้อผิดพลาดของโปรแกรมในระหว่างที่ผู้ใช้ใช้งานโปรแกรม และปรับปรุงโปรแกรมเมื่อเกิดข้อผิดพลาดขึ้น หรือในการใช้งานโปรแกรมไปนานๆ ผู้ใช้อาจต้องการเปลี่ยนแปลงการทำงาน ของระบบงานเดิมเพื่อให้เหมาะกับเหตุการณ์ นักเขียนโปรแกรมก็จะต้องคอยปรับปรุงแก้ไขโปรแกรมตามความต้องการของ ผู้ใช้ที่เปลี่ยนแปลงไปนั่นเอง



ที่มา : http://www.krurock.com/Programming/lesson01/lesson01_4.php

ภาษาคอมพิวเตอร์

มนุษย์ ใช้ภาษาในการสื่อสารมาตั้งแต่สมัยโบราณ การใช้ภาษาเป็นเรื่องที่มนุษย์พยายามถ่ายทอดความคิดและความรู้สึกต่าง ๆ เพื่อการโต้ตอบและสื่อความหมาย ภาษาที่มนุษย์ใช้ติดต่อสื่อสารในชีวิตประจำวัน เช่น ภาษาไทย ภาษาอังกฤษ หรือภาษาจีน ต่างเรียกว่า “ภาษาธรรมชาติ” (Natural Language) เพราะมีการศึกษา ได้ยิน ได้ฟัง กันมาตั้งแต่เกิดการใช้งานคอมพิวเตอร์ ซึ่งเป็นเครื่องมือทางอิเล็กทรอนิกส์ให้ทำงานตามที่ต้องการ จำเป็นต้องมีการกำหนดภาษา สำหรับใช้ติดต่อสั่งงานกับคอมพิวเตอร์

ภาษาคอมพิวเตอร์จะเป็น ”ภาษาประดิษฐ์” (Artificial Language) ที่มนุษย์คิดสร้างมาเอง เป็นภาษาที่มีจุดมุ่งหมายเฉพาะ มีกฎเกณฑ์ที่ตายตัวและจำกัด คืออยู่ในกรอบให้ใช้คำและไวยากรณ์ที่กำหนดและมีการตีความหมายที่ชัดเจน จึงจัดภาษาคอมพิวเตอร์เป็นภาษาที่มีรูปแบบเป็นทางการ (Formal Language) ต่างกับภาษาธรรมชาติที่มีขอบเขตกว้างมาก ไม่มีรูปแบบตายตัวที่แน่นอน กฎเกณฑ์ของภาษาจะขึ้นกับหลักไวยากรณ์และการยอมรับของกลุ่มผู้ใช้นั้น ๆ

ภาษาคอมพิวเตอร์อาจแบ่งได้เป็น 3 ระดับ คือ ภาษาเครื่อง (Machine Language) ภาษาระดับต่ำ (Low Level Language) และภาษาระดับสูง (High Level Language)

1 ภาษาเครื่อง (Machine Language)

การเขียนโปรแกรมเพื่อสั่งให้คอมพิวเตอร์ทำงานในยุคแรก ๆ จะต้องเขียนด้วยภาษาซึ่งเป็นที่ยอมรับของเครื่องคอมพิวเตอร์ที่เรียกว่า ”ภาษาเครื่อง” ภาษานี้ประกอบด้วยตัวเลขล้วน ทำให้เครื่องคอมพิวเตอร์สามารถทำงานได้ทันที ผู้ที่จะเขียนโปรแกรมภาษาเครื่องได้ ต้องสามารถจำรหัสแทนคำสั่งต่าง ๆ ได้ และในการคำนวณต้องสามารถจำได้ว่าจำนวนต่าง ๆ ที่ใช้ในการคำนวณนั้นถูกเก็บไว้ที่ตำแหน่งใด ดังนั้นโอกาสที่จะเกิดความผิดพลาดในการเขียนโปรแกรมจึงมีมาก นอกจากนี้เครื่องคอมพิวเตอร์แต่ละระบบมีภาษาเครื่องที่แตกต่างกันออก ทำให้เกิดความไม่สะดวกเมื่อมีการเปลี่ยนเครื่องคอมพิวเตอร์เพราะจะต้องเขียน โปรแกรมใหม่ทั้งหมด

2 ภาษาระดับต่ำ (Low Level Language)

เนื่อง จากภาษาเครื่องเป็นภาษาที่มีความยุ่งยากในการเขียนดังได้กล่าวมาแล้ว จึงไม่มีผู้นิยมและมีการใช้น้อย ดังนั้นได้มีการพัฒนาภาษาคอมพิวเตอร์ขึ้นอีกระดับหนึ่ง โดยการใช้ตัวอักษรภาษาอังกฤษเป็นรหัสแทนการทำงาน การใช้และการตั้งชื่อตัวแปรแทนตำแหน่งที่ใช้เก็บจำนวนต่าง ๆ ซึ่งเป็นค่าของตัวแปรนั้น ๆ การใช้สัญลักษณ์ช่วยให้การเขียนโปรแกรมนี้เรียกว่า ”ภาษาระดับต่ำ” ภาษาระดับต่ำเป็นภาษาที่มีความหมายใกล้เคียงกับภาษาเครื่อง มากบางครั้งจึงเรียกภาษานี้ว่า ”ภาษาอิงเครื่อง” (Machine – Oriented Language) ตัวอย่างของภาษาระดับต่ำ ได้แก่ ภาษาแอสเซมบลี เป็นภาษาที่ใช้คำในอักษรภาษาอังกฤษเป็นคำสั่งให้เครื่องทำงาน เช่น ADD หมายถึง บวก SUB หมายถึง ลบ เป็นต้น การใช้คำเหล่านี้ช่วยให้การเขียนโปรแกรมง่ายขึ้นกว่าการใช้ภาษาเครื่องซึ่ง เป็นตัวเลขล้วน ดังตารางแสดงตัวอย่างของภาษาระดับต่ำและภาษาเครื่องที่สั่งให้มีการบวกจำนวน ที่เก็บอยู่ในหน่วยความจำ

ภาษาระดับต่ำ	ภาษาเครื่อง	รหัสเลขฐานสิบหก
MOV AL,05	10110000 00000101	B0 05
MOV BL,08	10110011 00001000	B3 08
ADD AL,BL	00000000 11011000	00 D8
MOV CL,AL	10001000 11000001	88 C1

ตารางที่ 5.1 แสดงความสัมพันธ์ของคำสั่งในภาษาระดับต่ำและภาษาเครื่อง

จาก ตารางบรรทัดแรก 10110000 00000101 เป็นคำสั่งให้นำจำนวน 5 (หรือเขียนในรูปของเลขฐานสองเป็น 00000101) ไปเก็บในรีจิสเตอร์ชื่อ AL โดยส่วนแรก 10110000 คือรหัสคำสั่ง MOV ซึ่งเป็นการเคลื่อนย้ายข้อมูลจำนวนมาเก็บไว้ในรีจิสเตอร์ AL

บรรทัดที่ สอง 10110011 00001000 เป็นคำสั่งให้นำจำนวน 8 (หรือเขียนในรูปของเลขฐานสองเป็น

00001000) ไปเก็บในรีจิสเตอร์ชื่อ BL โดยส่วนแรก 10110011 คือรหัสคำสั่ง MOV ซึ่งเป็นการเคลื่อนย้ายข้อมูลจำนวนมาเก็บไว้ในรีจิสเตอร์ BL

บรรทัดที่สาม เป็นคำสั่งการบวกระหว่างรีจิสเตอร์ AL กับ BL หรือนำ 5 บวก 8 ผลลัพธ์เก็บในรีจิสเตอร์ AL

บรรทัดที่สี่ เป็นการนำผลลัพธ์จากรีจิสเตอร์ชื่อ AL ไปเก็บไว้ในรีจิสเตอร์ชื่อ CL

การใช้โปรแกรมที่เขียนด้วยภาษาแอสเซมบลีนั้น เครื่องคอมพิวเตอร์ไม่สามารถทำงานได้ทันที จำเป็นต้องมีการแปลโปรแกรมในการแปลที่มีชื่อว่า “แอสเซมเบลอร์” (Assembler) ซึ่งแตกต่างไปตามเครื่องคอมพิวเตอร์แต่ละชนิด ดังนั้นแอสเซมเบลอร์ของเครื่องชนิดหนึ่งจะไม่สามารถใช้แปลโปรแกรมภาษาแอสเซมบลีของเครื่องชนิดอื่น ๆ ได้ ภาษาแอสเซมบลีนี้ยังคงใช้ยาก เพราะผู้เขียนโปรแกรมจะต้องเข้าใจในการทำงานของเครื่องคอมพิวเตอร์อย่างละเอียด ต้องรู้ว่าจำนวนที่จะนำมาคำนวณนั้นอยู่ ณ ตำแหน่งใดในหน่วยความจำในทำนองเดียวกับการเขียนโปรแกรมเป็นภาษาเครื่อง ภาษาแอสเซมบลีจึงมีผู้ใช้ช้อย และมักจะใช้ในกรณีที่ต้องการควบคุมการทำงานของหน่วยในตัวเครื่องคอมพิวเตอร์

3 ภาษาระดับสูง (High Level Language)

ภาษา ระดับสูงเป็นภาษาที่สร้างขึ้นเพื่อช่วยอำนวยความสะดวกในการเขียนโปรแกรม กล่าวคือลักษณะของคำสั่งจะประกอบด้วยคำต่าง ๆ ในภาษาอังกฤษ ซึ่งผู้อ่านสามารถเข้าใจความหมายได้ทันที ผู้เขียนโปรแกรมจึงเขียนโปรแกรมด้วยภาษาระดับสูงได้ง่ายกว่าเขียนด้วยภาษาแอสเซมบลีหรือภาษาเครื่อง ภาษาระดับสูงมีมากมายหลายภาษา อาทิเช่น ภาษาฟอร์แทรน (FORTRAN) ภาษาโคบอล (COBOL) ภาษาปาสคาล (Pascal) ภาษาเบสิก(BASIC) ภาษาวิซวลเบสิก (Visual Basic) ภาษาซี (C) และภาษาจาวา (Java) เป็นต้น โปรแกรมที่เขียนด้วยภาษาระดับสูงแต่ละภาษาจะต้องมีโปรแกรมที่ทำหน้าที่แปล ภาษาระดับสูงให้เป็นภาษาเครื่อง เช่น โปรแกรมแปลภาษาฟอร์แทรนเป็นภาษาเครื่อง โปรแกรมแปลภาษาปาสคาลเป็นภาษาเครื่อง คำสั่งหนึ่งคำสั่งในภาษาระดับสูงจะถูกแปลเป็นภาษาเครื่องหลายคำสั่ง ภาษาระดับสูงที่จะกล่าวถึงในที่นี้ ได้แก่

โปรแกรมจำลองเพื่อแก้ปัญหาทางคณิตศาสตร์และวิทยาศาสตร์ แบบออนไลน์และออฟไลน์ ได้แก่ Scratch, Kodu, Code.org

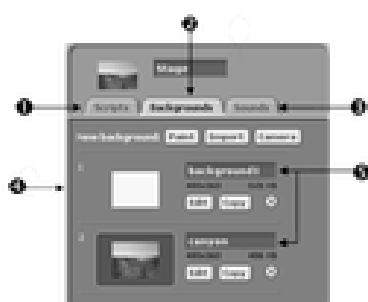
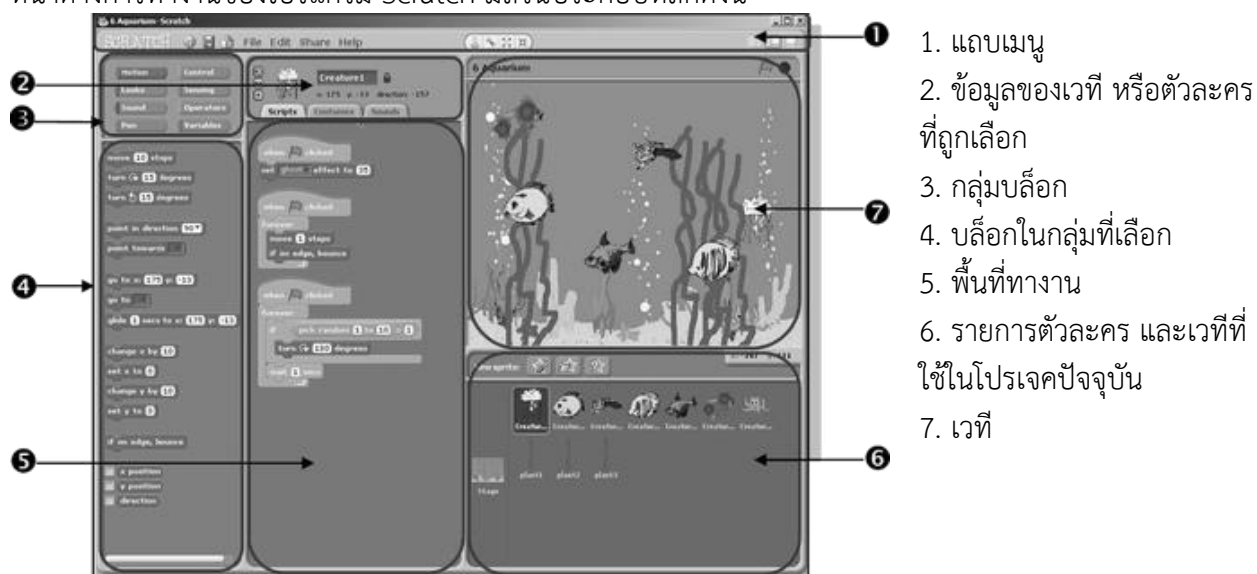
รู้จักโปรแกรม Scratch

โปรแกรม Scratch (อ่านว่า สะ-แครช) เป็นโปรแกรมภาษา ที่ผู้เรียนสามารถสร้างชิ้นงานได้อย่างง่าย เช่น นิทานที่สามารถโต้ตอบกับผู้อ่านได้ ภาพเคลื่อนไหว เกม ดนตรี และศิลปะ และเมื่อสร้างเป็นชิ้นงานเสร็จแล้วสามารถนำชิ้นงานที่สร้างสรรค์นี้ แสดง และแลกเปลี่ยนความคิดเห็นร่วมกับผู้อื่นบนเว็บไซต์ได้ ทำให้ผู้เรียนได้เรียนรู้หลักการทางคณิตศาสตร์ และแนวคิดการโปรแกรมไปพร้อมๆ กับการคิดอย่างสร้างสรรค์ มีเหตุผล เป็นระบบ และเกิดการทำงานร่วมกัน



1. ส่วนประกอบหลักของโปรแกรม

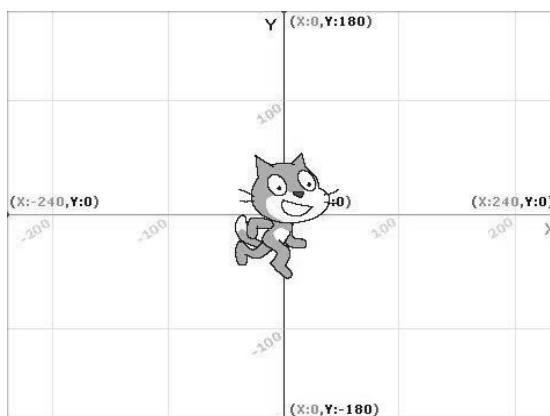
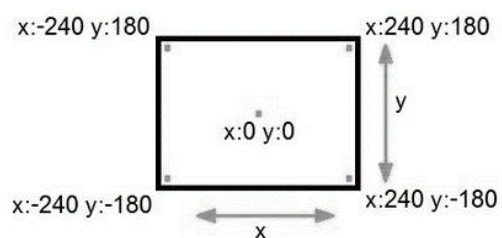
หน้าต่างการทำงานของโปรแกรม Scratch มีส่วนประกอบหลักดังนี้



รายละเอียดของเวที

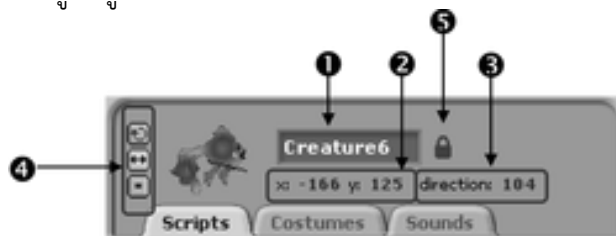
1. แท็บ Scripts
2. แท็บ Backgrounds
3. แท็บ Sounds
4. สร้างพื้นหลังใหม่ (New Background)
5. พื้นหลังลำดับที่ 1 และ 2

การบอกตำแหน่งใดๆ บนเวทีจะบอกโดยใช้ค่า (x, y) โดยค่า x และ y ที่ตำแหน่ง (0, 0) จะอยู่ตรงกลางเวที



2. ตัวละคร

ตัวละครแต่ละตัวจะมีข้อมูลแตกต่างกัน โดยสามารถคลิกที่ภาพตัวละครในพื้นที่แสดงรายการตัวละครเพื่อดูข้อมูลของตัวละครนั้น เช่น ตัวละครปลาทองในโปรเจก Aquarium มีข้อมูลดังตาราง



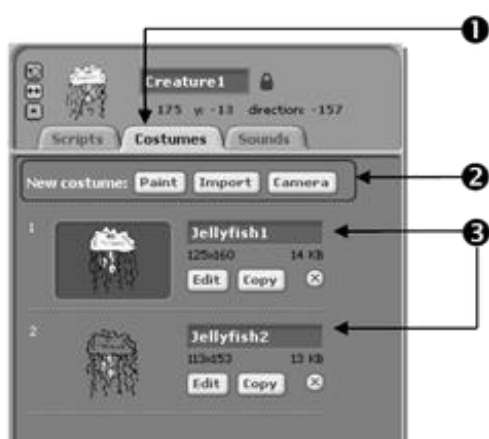
1. ชื่อตัวละคร Creature 6
2. ตำแหน่งบนเวทีที่ x: -166 y: 125
- 3 ทิศทางการเคลื่อนที่ (direction) 104 องศา
4. รูปแบบการหมุน มี 3 ลักษณะ
 - หมุนได้รอบทิศทาง
 - หันได้เฉพาะซ้ายหรือขวา
 - ห้ามหมุน
5. การลากตัวละครในโหมดนำเสนอหรือบนเว็บไซต์
 - แม่กุญแจปิด ลากตัวละครไม่ได้
 - แม่กุญแจเปิด

2.1 ชื่อตัวละคร

โปรแกรมจะตั้งให้เป็น Sprite 1, 2, 3... ตามลำดับที่สร้างขึ้นโดยอัตโนมัติ ถ้าต้องการเปลี่ยนชื่อตัวละครให้พิมพ์ชื่อใหม่บนแถบชื่อหมายเลข 1.

2.2 ชุดตัวละคร

ชุดตัวละคร (Costumes) เป็นลักษณะของตัวละคร ซึ่งสามารถเปลี่ยนแปลงภาพเดิม หรือเพิ่ม หรือเพิ่มภาพใหม่ และอาจเขียนสคริปต์ให้กับตัวละครเปลี่ยนแปลง หรือเคลื่อนไหวในรูปแบบต่างๆ



รายละเอียดชุดตัวละคร

1. แท็บ Costumes
2. สร้างชุดตัวละครใหม่
3. ชุดตัวละคร ในตัวอย่างเป็นชุดของตัวละครชื่อ Creature 1 ชื่อชุด Jellyfish1 มีลักษณะสีขาและ Jellyfish2 มีลักษณะโปร่งใส

2.3 สคริปต์

สคริปต์คือชุดคำสั่งสำหรับตัวละครหรือเวที เพื่อสั่งให้ตัวละครหรือเวทีทำงานตามวัตถุประสงค์ที่ต้องการ โดยการเลือกสคริปต์จากกลุ่มบล็อก ซึ่งแบ่งเป็น 8 กลุ่ม ดังนี้

กลุ่มบล็อก	การทำงาน
Motion	การเคลื่อนไหว เช่น เคลื่อนที่ไปข้างหน้า หมุนไปทางซ้ายหรือขวา
Control	การควบคุม เช่น การวนซ้ำ การตรวจสอบเงื่อนไข
Looks	การแสดงผล เช่น พูด คิด เปลี่ยนขนาด
Sensing	การรับรู้ เช่น สัมผัสกับขอบ คลิกเมาส์ จับเวลา
Sound	การแสดงผลเสียง เช่น เล่นเสียงสัตว์ กลอง โน้ต
Operators	ตัวดำเนินการ เช่น บวก และ หรือ
Pen	ปากกา เช่น ยกปากกาขึ้น ตั้งค่าสีปากกา
Variables	ตัวแปร เช่น สร้างค่าตัวแปร

3. การสั่งให้โปรเจกต์เริ่มทำงานและหยุดทำงาน

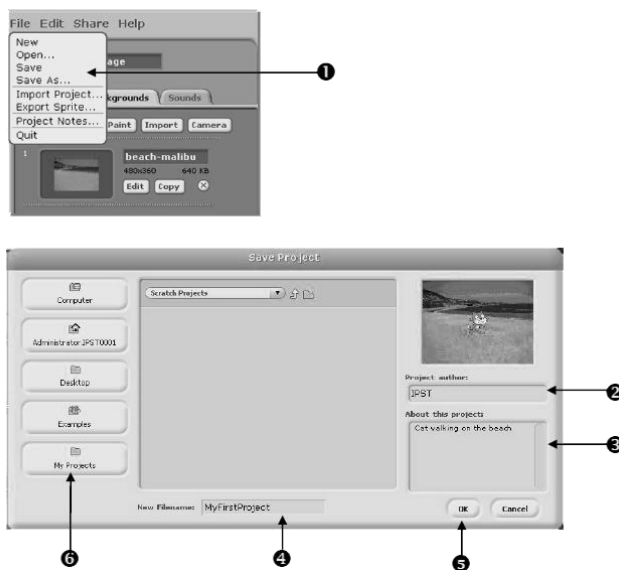
การสั่งให้โปรเจกต์เริ่มทำงาน ทำได้โดยคลิก  ซึ่งอยู่มุมขวาของเวที โดยทุกสคริปต์ของทุกตัวละคร

และเวที ที่เริ่มต้นสคริปต์ด้วยบล็อก



จะเริ่มทำงานพร้อมกัน และถ้าต้องการหยุดการทำงาน

ทั้งโปรเจกต์ ให้คลิก 



4. การบันทึกโปรเจค

4.1 คลิกเมนู File-> Save หรือ Save As จะปรากฏกรอบโต้ตอบ Save Project

4.2 พิมพ์ชื่อผู้ทำโปรเจค

4.3 พิมพ์รายละเอียดเกี่ยวกับโปรเจค

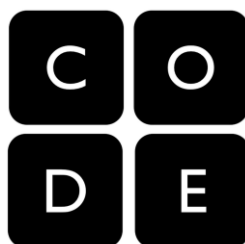
4.4 พิมพ์ชื่อโปรเจค จะได้ไฟล์ชื่อข้อมูลที่มีส่วนขยายเป็น .sb ในไฟล์เดออร์ที่ใช้บันทึกงาน เช่น

MyFirstProject.sb

4.5 คลิกปุ่ม OK

4.6 โปรเจคจะถูกบันทึกไว้ที่ C:\Documents and Settings\Administrator\My Documents\Scratch Projects (ไฟล์เดออร์ My Projects)

รู้จักโปรแกรม Code.org



code.org เป็นเว็บไซต์ไม่แสวงผลกำไรสัญชาติอเมริกัน เป็นองค์กรระดับโลกที่มีเป้าหมายชัดเจนเพื่อสนับสนุนการศึกษาด้าน Computer Science (วิทยาการคอมพิวเตอร์) เป็นการให้โอกาสนักเรียนทุกคนทั่วโลกได้เรียน โดยเฉพาะเด็กขาดโอกาสในหลายๆ โรงเรียน และโครงการขององค์กรที่หลายๆ คนนี้น่าจะคุ้นชื่อมากที่สุดก็คือ “Hour of Code” (หนึ่งชั่วโมงกับการเขียนโปรแกรม) ซึ่งเป็นโครงการที่จะช่วยฝึกให้ทุกคนได้ฝึกเรียนเขียนโปรแกรมแบบง่ายๆ ไม่ว่าจะเป็เด็กนักเรียน นักศึกษา ผู้ปกครอง และบุคคลทั่วไป

เนื่องจากผู้ก่อตั้ง code.org เชื่อว่าการเรียนรู้ด้านคอมพิวเตอร์นั้นมีความสำคัญอย่างยิ่งกับเด็กๆ และควรจะให้เด็กได้ศึกษาเช่นเดียวกับวิชาพื้นฐานอย่างคณิตศาสตร์ หรือวิทยาศาสตร์ จึงได้พยายามหาวิธีที่จะทำให้เด็กๆ หรือผู้ใหญ่) มาสนใจการเขียนโปรแกรมผ่านทาง code.org

ทางองค์กรนั้นได้รับการสนับสนุนจากหลายภาคส่วน เช่น Microsoft, Facebook, Google, Apple, Amazon, Disney เป็นต้น ซึ่งเหล่า CEO หลายๆ คน เช่น มาร์ก ซักเคอร์เบิร์ก ก็ลงมาสอนเขียนโปรแกรมเองด้วย



5 Missions หลักของ code.org

1. สร้างเครื่องมือในการเรียนเขียนโปรแกรม
2. มีหลักสูตรฟรีที่คุณครูสามารถนำไปปรับใช้สอนได้เลย
3. จัดโครงการอบรมคุณครูที่ไร้พื้นฐานในการเขียนโปรแกรม
4. ภารกิจขับเคลื่อนผลักดันนโยบายทางการศึกษา
5. ภารกิจเปลี่ยนค่านิยมผิดๆ เกี่ยวกับการเรียนเขียนโปรแกรม

สื่อเรียนรู้ใน code.org มีให้เลือกใช้หลากหลายตามระดับชั้น และอายุของผู้เรียน มีทั้งแบบใช้และไม่ใช้คอมพิวเตอร์ สำหรับสื่อที่ใช้คอมพิวเตอร์จะต้องเชื่อมต่ออินเทอร์เน็ต โดยการเรียนรู้จะมีลักษณะเป็นการฝึกทักษะ ซึ่งมีการฝึกต่างๆ ให้ผู้เรียนได้ฝึกเขียนโปรแกรมตามสถานการณ์ที่กำหนด โดยเริ่มจากภารกิจง่ายๆ ไปจนถึงภารกิจที่มีความซับซ้อน ในแต่ละภารกิจจะกำหนดคำสั่งให้เลือกใช้ตามระดับความยากของสถานการณ์ และมีการกำหนดจำนวนคำสั่งที่ใช้ได้สูงสุดในแต่ละภารกิจ นักเรียนสามารถเลือกทำภารกิจใดก่อนก็ได้ โดยถ้าผ่านภารกิจใดที่แถบด้านบนจะมีสัญลักษณ์เป็นวงกลมสีเขียว ถ้าไม่ผ่านจะมีสัญลักษณ์เป็นวงกลมสีเหลือง ถ้ายังไม่ได้ทำจะมีสัญลักษณ์เป็นวงกลมสีขาวตั้งรูป



การเขียนโปรแกรมในแต่ละภารกิจ นักเรียนจะต้องเลือกวางคำสั่งให้ได้ผลลัพธ์ตามสถานการณ์ที่กำหนด หลังจากนั้นสั่งให้โปรแกรมทำงานโดยคลิกปุ่ม **Run** หากเขียนโปรแกรมถูกต้องจะมีข้อความแสดงว่าผ่าน และแสดงจำนวนคำสั่งที่ใช้ หากเขียนไม่ถูกต้องจะมีข้อความแจ้งเหตุผล เช่น ใช้คำสั่งไม่ถูกต้อง จำนวนคำสั่งเกินกว่าที่กำหนด



การเรียนรู้ที่ดีควรเริ่มจากภารกิจที่ง่ายไปจนถึงภารกิจที่มีความซับซ้อน เพื่อให้นักเรียนเกิดความคิดรวบยอดในแต่ละเรื่อง แล้วนำมาประยุกต์ใช้ในการสร้างงานตามจินตนาการ หรืองานตามที่สถานการณ์กำหนด ซึ่งนักเรียนจะมีโอกาสได้ทำในตอนท้ายของแต่ละกิจกรรม เมื่อผ่านกิจกรรมแล้วนักเรียนจะได้รับเกียรติบัตรและสามารถแสดงความภาคภูมิใจนี้ผ่านสื่อ เช่น Facebook และ Twitter ได้ หรือจะเก็บไว้กับตัวเองโดยการดาวน์โหลดไว้ในเครื่องคอมพิวเตอร์

Congratulations on completing one Hour of Code



Personalize your certificate

Share your achievement