

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาไทม์แมชชีนด้วยภาษาเอชทีเอ็มแอลห้าและจาวาสคริปต์
DEVELOPMENT OF TIME MACHINE USING HTML5 AND
JAVASCRIPT

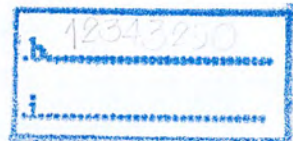


T117392



อภิสิทธิ์ ทรงวัฒนาสกุล
อมรินทร์ จันทจิรโกวิท
อรรณกร พรเอื้อไพโรจน์

เลขหมู่
เลขทะเบียน... 117392
มีเดือนปี... 1 ค.ศ. 2554



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ **ปีการศึกษา 2553** ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาไทม์แมชชีนด้วยภาษาเอชทีเอ็มแอลห้าและจาวาสคริปต์

DEVELOPMENT OF TIME MACHINE USING HTML5 AND JAVASCRIPT

ผู้จัดทำ

1. นายอภิสิทธิ์ ทรวงวัฒนาสกุล รหัสนักศึกษา 50011863
2. นายอมรินทร์ จันทจิร โกวิท รหัสนักศึกษา 50011880
3. นายอรรถกร พรเชื้อ ไพโรจน์ รหัสนักศึกษา 50011886



(Handwritten signature)

(ผศ.ดร.วิศิษฎ์ หิรัญภิตติ)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาใหม่เมซซิ่งด้วยภาษาเอชทีเอ็มแอลห้าและจาวาสคริปต์

นายอภิรักษ์	ทรงวัฒนาสกุล	50011863
นายอมรินทร์	จันทจิร โกวิท	50011880
นายอรรถกร	พรเอื้อไพโรจน์	50011886
ผศ.ดร.วิศิษฐ์	หิรัญกิตติ	อาจารย์ที่ปรึกษา
ปีการศึกษา 2553		

บทคัดย่อ

การพัฒนาใหม่เมซซิ่งด้วยภาษาเอชทีเอ็มแอลห้าและจาวาสคริปต์ นั้นเป็นการนำเอาภาษา CL (Communication Language) ที่สามารถใช้สั่งงานให้คอมพิวเตอร์นั้นทำงานเป็นคาบๆซ้ำๆ หรือ การสั่งงานตามกำหนดเวลาที่ได้กำหนดไว้ก็นำมาประยุกต์ใช้งานบนเว็บเบราว์เซอร์ เพื่อให้ผู้ใช้สามารถเข้าถึงได้อย่างสะดวก เนื่องจากโลกของการสื่อสารในปัจจุบันไม่ว่าจะเป็นโลกของโทรศัพท์เคลื่อนที่ หรือ โลกของอินเทอร์เน็ต สามารถเข้าถึงได้อย่างง่ายในทุกที่ทุกเวลา ดังนั้นปริญญาานิพนธ์ฉบับนี้จึงขอเสนอวิธีการนำเอาภาษา CL ที่สามารถทำงานได้บน เว็บเบราว์เซอร์ ซึ่งสามารถเข้าถึงได้ทุกที่ทุกเวลาในปัจจุบันไม่ว่าจะเป็นโทรศัพท์เคลื่อนที่ หรือแล็ปท็อป โดยสร้างอินเทอร์เน็ตฟรี้ดเตอร์ที่สามารถแปลภาษา CL ให้สามารถทำงานบนเว็บไซด์ซึ่งทำงานด้วยภาษาจาวาสคริปต์ อีกทั้งยังทำการปรับปรุงส่วนของ Object-Oriented Programming ที่มีปัญหาบนภาษาจาวาสคริปต์ โดยดำเนินการสร้างขึ้นมารวมไปถึงการสร้าง integrated development environment (IDE) ที่สามารถทำงานร่วมกับอินเทอร์เน็ตฟรี้ดเตอร์ได้อย่างมีประสิทธิภาพและเหมาะสมกับการใช้ร่วมกับภาษา CL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Development of Time Machine Using HTML5 and JavaScript

Mr. Apiluck Songwattanasakul 50011863

Mr. Amarin Chanthajirakowit 50011880

Mr. Attagorn Ponaupairoj 50011886

Asst.Prof.Dr. Visit Hirankitti Advisor

Academic Year 2010

ABSTRACT

Development of Time Machine Using HTML5 and JavaScript is the application of CL language. It can operate the computer is running a session any repeated or run on a schedule has been defined and bring it to apply on web browser. User can access easily because today the world of the communication is convenient, whether it is the world's cell phones or the internet easily accessible anywhere, anytime. Therefore, this thesis offers the solution to bring CL language work on web browser, which can be access anytime, whether it is a Mobile phone or Laptop. This thesis builds interpreter can be translate Python programming language to JavaScript programming language which works on web browser and then fix the problem in JavaScript about Object-oriented programming by using the MOOTOOLS Framework, in addition this thesis build the integrated development environment(IDE). It is compatible with the interpreter is efficient and suitable for using with CL language.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จไปได้ด้วยสมบูรณ์ ด้วยการให้คำแนะนำและคำปรึกษาจาก ผศ. ดร. วิศิษฐ์ หิรัญกิตติ ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญาานิพนธ์ และอาจารย์ทุกท่านในภาควิชาวิศวกรรมศาสตร์ สาขาคอมพิวเตอร์ที่ได้ให้คำแนะนำและการช่วยเหลือในการทำปริญญาานิพนธ์ฉบับนี้ ทำให้ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี กลุ่มผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านอาจารย์ทุกท่านและขอขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ทุกท่านตั้งแต่เริ่มเข้าศึกษาจนถึงปัจจุบันที่ได้ให้ความรู้เพื่อนำไปประกอบอาชีพทางสายงานต่างๆ ให้แก่ข้าพเจ้า ทำให้ข้าพเจ้าได้เป็นนักศึกษาคณะวิศวกรรมคอมพิวเตอร์และจบไปเป็นวิศวกรคอมพิวเตอร์ที่ดี

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทุกคนทั้งในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังและที่อื่น ที่ได้ให้คำแนะนำและให้ความร่วมมือช่วยเหลือในการปฏิบัติงาน ทำให้ปริญญาานิพนธ์ฉบับนี้บรรลุเป้าหมายได้อย่างสมบูรณ์แบบ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่คอยเป็นกำลังใจและให้การสนับสนุนในทุกๆ เรื่อง ให้ความรักและดูแลห่วงใยเป็นอย่างดี จนทำให้ข้าพเจ้าเติบโตเป็นตัวเป็นตนมาจนถึงทุกวันนี้ได้ ขอขอบพระคุณเป็นอย่างสูง

อภิสิทธิ์ ทรวงวัฒนาสกุล
อมรินทร์ จันทจิร โกวิท
อรรดกร พรเอื้อไพโรจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาของปัญหา.....	1
1.2 วัตถุประสงค์ของการศึกษา.....	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	1
1.4 ขอบเขตของปริญญาานิพนธ์.....	2
1.5 ขั้นตอนการดำเนินงาน.....	2
1.6 ส่วนประกอบของปริญญาานิพนธ์.....	2
บทที่ 2 ไวยากรณ์ภาษาไพธอน และ ภาษา CL.....	4
2.1 ภาษาไพธอน.....	4
2.2 ความสามารถของภาษาไพธอน.....	4
2.3 ไวยากรณ์ของภาษาไพธอน.....	5
2.4 ภาษา CL.....	8
2.5 ความหมายของเวลา.....	9
2.6 รูปแบบของเวลา.....	10
2.7 คำสั่งการทำงานในรูปแบบ “ช่วงเวลา: การทำงาน”.....	11
2.8 ชุดคำสั่งภาษา CL.....	11
2.9 การนิยามฟังก์ชันในภาษา CL.....	13
2.10 การสั่งงานด้วยเหตุการณ์.....	13
2.11 สรุปลงภาษาไพธอน.....	14
2.12 สรุปลงภาษา CL.....	14

สารบัญ (ต่อ)

	หน้า
บทที่ 3 ภาษาจาวาสคริปต์.....	15
3.1 ภาษาจาวาสคริปต์.....	15
3.2 เหตุผลที่เลือกใช้ความสามารถของภาษาจาวาสคริปต์.....	15
3.3 DOM (Document Object Model).....	16
3.4 Framework ของ ภาษาจาวาสคริปต์.....	17
บทที่ 4 HTML5.....	18
4.1 ภาษา HTML5.....	18
4.2 ไวยากรณ์ของ HTML5.....	19
4.3 เหตุผลที่เลือกใช้ความสามารถของ HTML5.....	21
4.4 Offline web applications.....	21
4.5 Local storage.....	22
บทที่ 5 CL อินเทอร์เน็ต.....	23
5.1 Lexer.....	24
5.2 Parser.....	25
5.3 Walker.....	26
5.4 Interpreter.....	26
5.5 CL อินเทอร์เน็ต.....	27
5.6 การออกแบบและพัฒนา CL อินเทอร์เน็ต.....	28
5.7 Time table manager	29
บทที่ 6 Integrated development environment.....	31
6.1 ส่วนประกอบของ Integrated development environment.....	31
6.2 รูปแบบในการออกแบบ IDE.....	32
บทที่ 7 Synchronization.....	33
7.1 การ Synchronization.....	33

สารบัญ (ต่อ)

	หน้า
7.2 หลักการทำงานของ การ Synchronization.....	34
7.3 Tool ในใช้ในการพัฒนา Synchronization.....	35
7.4 โปรแกรมของฝั่งคอมไพเตอร์.....	37
7.5 โปรแกรมของฝั่งเซิร์ฟเวอร์.....	38
บทที่ 8 ผลการทดลอง.....	39
8.1 ผลการทดลองส่วนของไพธอนอินเตอร์พรีเตอร์.....	39
8.2 ผลการทดลองส่วนของ Integrated development environment (IDE).....	44
8.3 ผลการทดลองส่วนของ CL Interpreter.....	49
บทที่ 9 บทสรุปและวิจารณ์ผลการทดลอง.....	54
9.1 บทสรุป.....	54
9.2 แนวทางในการพัฒนาต่อ.....	54
9.3 ข้อเสนอแนะ.....	55
บรรณานุกรม.....	56

สารบัญตาราง

ตาราง	หน้า
4.1 tag HTML5 ส่วนของการจัดการหน้าเว็บ	19
4.2 tag HTML5 ส่วนของการจัดการสื่อ multimedia และ ส่วนของการจัดการกราฟฟิก.....	20
4.3 tag HTML5 ส่วนของการจัดการ input และ output	20
4.4 HTML5 ส่วนของประเภทของ input	20
5.1 การ Assign ค่าไปตารางของ Token	25
8.1 ความสามารถในการอินเทอร์เน็ตฟรีของ skulpt.org.....	40
8.2 ความสามารถในการทำงานของอินเทอร์เน็ตฟรีเตอร์ของโครงการ	42



สารบัญรูป

รูป	หน้า
2.1 ประโยคเงื่อนไข IF	7
2.2 ประโยคทำซ้ำ WHILE	7
2.3 ประโยคทำซ้ำ FOR	7
2.4 ความหมายของเวลา.....	9
2.5 ความหมายของจุดเวลา ระยะเวลา.....	9
2.6 ชุดคำสั่งแบบซีควนท์	11
2.7 ชุดคำสั่งแบบอันออร์เดอร์เรด.....	12
2.8 ชุดคำสั่งแบบพาราเรล	12
2.9 ชุดคำสั่งแบบ Nested.....	13
2.10 การนิยามฟังก์ชันในภาษา CL	13
2.11 รูปแบบระบุช่วงเวลาที่เหตุการณ์นั้นจะเกิดขึ้น	14
2.12 รูปแบบการระบุการทำงานเพื่อตอบสนองเหตุการณ์เกิดขึ้นด้วยเงื่อนไข	14
3.1 รูปแบบของ DOM ในแต่ละระดับชั้น	16
4.1 ตัวอย่างเอลเมนต์ใหม่ใน HTML5.....	18
4.2 Offline web application architecture	21
4.3 การใช้งาน Local storage.....	22
5.1 การทำงานของอินเทอร์เน็ตพรีเตอร์	23
5.2 การทำงานของ Transition Diagram	24
5.3 การอินเทอร์เน็ตพรีเตอร์คำสั่งประเภทของ Comparison Operator.....	26
5.4 การทำงานของ CL อินเทอร์เน็ตพรีเตอร์	27
5.5 การประกาศ Token.....	28
5.6 BNF ของภาษา CL	29
5.7 BNF ของภาษา CL(2).....	29
5.8 หลักการสร้าง Symbol table.....	30
7.1 อีเวนท์ในปฏิทิน	33
7.2 โมดูล ในปฏิทิน.....	33
7.3 คอมพิวเตอร์ที่ใช้ในโปรไฟล์เดียวกัน	34
7.4 การ Synchronization เป็นคาบเวลา.....	34

สารบัญรูป (ต่อ)

รูป	หน้า
7.5 การ Synchronization เมื่อคอมพิวเตอร์มีการแก้ไข	35
7.6 Ajax Framework Code Skeleton.....	36
7.7 รูปแบบของรีเคิวสของคอมพิวเตอร์ เมื่อคอมพิวเตอร์ครบคาบเวลา.....	37
7.8 รูปแบบของรีเคิวส ของคอมพิวเตอร์ เมื่อคอมพิวเตอร์มีการแก้ไข	37
7.9 โปรแกรมที่ทำงานเมื่อฐานข้อมูลไม่เหมือนกัน	38
7.10 การทำงานฝั่งเซิร์ฟเวอร์	38
8.1 รูปเปรียบเทียบการทำงานของ skulpt.org กับการทำงานของปริณูณานิพนธ์	39
8.2 หน้าต่าง create module.....	44
8.3 การ suggest ฟังก์ชันที่อยู่ใน class	45
8.4 การ suggest บิวท์อินฟังก์ชันที่อยู่ใน list	45
8.5 การแสดงผลของ IDE โดยใช้สีที่แตกต่างกัน	45
8.6 รูปแบบของ IDE ในหน้าต่างของวัน และสัปดาห์	46
8.7 รูปแบบของ IDE ในหน้าต่างของเดือน และปี	46
8.8 หน้าต่างที่ใช้ในการเพิ่ม CL statement	47
8.9 การเพิ่มเติม CL statement ลงใน IDE (1).....	47
8.10 การเพิ่มเติม CL statement ลงใน IDE (2).....	47
8.11 รูปแบบของ IDE ในหน้าต่างของวัน และสัปดาห์ เมื่อเพิ่ม CL statement.....	48
8.12 รูปแบบของ IDE ในหน้าต่างของเดือน และปี เมื่อเพิ่ม CL statement	48
8.13 การแสดงผลในการค้นหา (1).....	48
8.14 การแสดงผลในการค้นหา (2).....	49
8.15 การ drag-drop ของ CL statement	49
8.16 การทำงานช่วงเวลา 10:38:13	50
8.17 การทำงานช่วงเวลา 10:40:00	50
8.18 การทำงานช่วงเวลา 10:52:52	51
8.19 การทำงานช่วงเวลา 10:55:02	51
8.20 การนำคำสั่งทำงานใน โปรแกรมช่วงเวลา 15:20:42	52
8.21 การทำงานในโปรแกรมช่วงเวลา 15:20:48	52
8.22 เปิดเว็บเบราว์เซอร์เมื่อถึงเวลา 15:21:00.....	53

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันนั้น โลกของอินเทอร์เน็ต นั้น ได้เป็นส่วนหนึ่งในชีวิตประจำวันของประชากรโลก เนื่องจากการที่สามารถเข้าถึงได้ง่ายและมีราคาที่เหมาะสม ผนวกกับการดำเนินกิจกรรมในชีวิตประจำวันนั้นมีความยุ่งยากและซับซ้อนมากยิ่งขึ้น ในโลกคอมพิวเตอร์นั้น มีโปรแกรมที่มีความสามารถจัดการเกี่ยวกับชีวิตประจำวันมากมาย แต่ยังไม่มียโปรแกรมใดที่สามารถเป็นตัวแทนที่จะทำกิจกรรมต่างๆในชีวิตประจำวันแทนได้ ซึ่งจะทำให้ผู้ใช้งานนั้นมีความสะดวกและช่วยในการจัดการกับชีวิตประจำวันในโลกของคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ ตัวอย่างเช่น การธุรกรรมต่างๆผ่านอินเทอร์เน็ต ไม่ว่าจะเป็นเกี่ยวกับการเงิน การส่งอีเมลล์ ฯลฯ

1.2 วัตถุประสงค์ของการศึกษา

- 1) เพื่อนำสิ่งที่เรียนรู้ในทางทฤษฎีนำมาประยุกต์ใช้ในทางปฏิบัติ
- 2) เพื่อนำความรู้มาพัฒนาให้เป็นประโยชน์แก่สังคม
- 3) เพื่อทำการศึกษาการทำงานของตัวแปลภาษาในคอมพิวเตอร์ ที่ทำงานบนเว็บเบราว์เซอร์
- 4) เพื่อทำการศึกษารูปแบบการทำงาน ไวยากรณ์ ของภาษาต่างๆ เช่น ไพธอน, จาวาสคริปต์, HTML5, CL
- 5) เพื่อทำการศึกษาการทำงานของ Integrated development environment (IDE) ที่ทำงานอยู่บนเว็บเบราว์เซอร์

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้รับความรู้ และความเข้าใจในการสร้างตัวแปลภาษาในทางคอมพิวเตอร์ ที่ทำงานบนเว็บเบราว์เซอร์
- 2) ได้รับความรู้ และความเข้าใจในรูปแบบการทำงาน ไวยากรณ์ ของภาษาต่างๆ เช่น ไพธอน, จาวาสคริปต์, HTML5, CL
- 3) ได้รับความรู้ และความเข้าใจในการทำงานของ Integrated development environment (IDE) ที่ทำงานอยู่บนเว็บเบราว์เซอร์
- 4) ได้รับความรู้ และความเข้าใจการทำงานร่วมกับผู้อื่น อีกทั้งยังเป็นการบริหารเวลาในการทำงานอย่างเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของโครงการ

ศึกษาการทำงานของตัวแปลภาษา แล้วนำความรู้ที่ได้นั้นมาสร้างตัวแปลภาษา CL ที่สามารถทำงานบนเว็บเบราว์เซอร์ซึ่งทำงานด้วยภาษาจาวาสคริปต์ รวมถึงการสร้างและออกแบบ Integrated development environment (IDE) เพื่อนำมาช่วยเขียนภาษาไพธอนได้อย่างสะดวกยิ่งขึ้น

1.5 ขั้นตอนการดำเนินงาน

- 1) ศึกษาการทำงานของตัวแปลภาษาในทางคอมพิวเตอร์ ที่ทำงานบนเว็บเบราว์เซอร์
- 2) ศึกษารูปแบบการทำงาน ไวยากรณ์ ของภาษาต่างๆ เช่น ไพธอน, จาวาสคริปต์, HTML5, CL
- 3) ศึกษาการทำงานของ Integrated development environment (IDE) ที่ทำงานบนเว็บเบราว์เซอร์
- 4) ทำการสร้างตัวแปลภาษาในทางคอมพิวเตอร์ ที่ทำงานบนเว็บเบราว์เซอร์โดยทำการสร้างในส่วนของฟังก์ชันที่ทำการ Execute คำสั่งจาก parser
- 5) สร้าง และออกแบบ Integrated development environment (IDE)
- 6) ทดสอบการทำงานของตัวแปลภาษาไพธอน โดยทำงานร่วมกับ IDE

1.6 ส่วนประกอบของปริญาณิพนธ์

เนื้อหาของปริญาณิพนธ์ฉบับนี้ประกอบไปด้วย 9 บท คือ บทนำ, ไวยากรณ์ภาษาไพธอน และ ภาษา CL, ภาษาจาวาสคริปต์, HTML 5, CL อินเตอร์พรีเตอร์, Integrated development environment, CL อินเตอร์พรีเตอร์ ใน Integrated development environment, Synchronization, ผลการทดลอง และบทสรุปและวิจารณ์ผลการทดลอง โดยสามารถจำแนกรายละเอียดได้ดังต่อไปนี้

- 1) บทที่ 1 กล่าวถึงความเป็นมา วัตถุประสงค์ ประโยชน์ที่ได้รับ ขอบเขตของการศึกษา ขั้นตอนการดำเนินงาน และส่วนประกอบของปริญาณิพนธ์เล่มนี้
- 2) บทที่ 2 กล่าวถึงไวยากรณ์ภาษาไพธอน และ ภาษา CL ในส่วนของหลักการการทำงานทั่วไปของภาษา เหตุผลที่เลือกใช้ภาษาไพธอนในการทำปริญาณิพนธ์ คำสงวนต่างๆในภาษาไพธอน คำทับศัพท์ต่างๆในภาษาไพธอน รวมไปถึง ไวยากรณ์ของภาษาในส่วนที่เกี่ยวข้องกับตัวแปลภาษา ในส่วนของภาษา CL เป็นการอธิบายหลักการการทำงานทั่วไปของภาษา เหตุผลที่เลือกใช้ภาษา CL ไวยากรณ์ของภาษา CL และกล่าวถึงตัวแปลภาษาไพธอนที่ทำงานบนเว็บเบราว์เซอร์หลักการทั่วไปของตัวแปลภาษา รวมถึงทฤษฎีพื้นฐาน ขั้นตอนการทำงานของตัวแปลภาษา lexical analysis, tokenizer, parser, interpreter and execute instruction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3) บทที่ 3 กล่าวถึงภาษาจาวาสคริปต์ ในส่วนหลักการทำงานของภาษา การนำไปใช้ให้เกิดประโยชน์ และกล่าวถึงเฟรมเวิร์ค ที่นำมาใช้ช่วยในการพัฒนา
- 4) บทที่ 4 กล่าวถึง HTML 5 ในส่วนของหลักการทำงานของฟีเจอร์ใหม่ที่แตกต่างกันจาก HTML 4 ส่วนที่นำมาใช้ในงานวิจัย Offline working, Local storage web socket
- 5) บทที่ 5 กล่าวถึงความหมายของ CL อินเทอร์เน็ตเวิร์ด วิธีการวิเคราะห์ศัพท์ขั้นตอนแรก ขั้นตอนในการสังเคราะห์รูปแบบกฎเกณฑ์ ตัวอย่างการอินเทอร์เน็ตเวิร์ดคำสั่ง และ ขั้นตอนการทำงานของ CL อินเทอร์เน็ตเวิร์ด
- 6) บทที่ 6 กล่าวถึง Integrated development environment (IDE) ที่สามารถทำงานบนเว็บเบราว์เซอร์หลักการทั่วไปของ IDE รวมถึงทฤษฎีพื้นฐาน รูปแบบในการออกแบบ IDE ฟังก์ชันการทำงาน
- 7) บทที่ 7 กล่าวถึงความเป็นมาของการ Synchronization หลักการทำงานของ Synchronization เครื่องมือที่ใช้ในการพัฒนา โปรแกรมของฝั่งผู้ใช้ และ ฝั่งเซิร์ฟเวอร์
- 8) บทที่ 8 กล่าวถึงผลการทดลอง ทดสอบการทำงานของตัวแปลภาษาที่ทำงานบนเว็บเบราว์เซอร์ การแสดงผลต่างๆ ในหน้าต่างของปฏิทิน และผลการทดลองเมื่อทำงานร่วมกับ IDE
- 9) บทที่ 9 กล่าวถึงข้อสรุปของการวิจัย

บทที่ 2

ไวยากรณ์ภาษาไพธอน และ ภาษา CL

ในบทนี้จะกล่าวถึงภาษาไพธอน และภาษา CL ซึ่งเหตุผลที่เลือกภาษาไพธอนนั้น เพราะว่า ภาษาไพธอนเป็นภาษาที่เข้าใจง่ายและมีการแปลภาษาโดยใช้หลักการของอินเตอร์พรีเตอร์เหมาะแก่การนำไปประยุกต์ใช้ในภาษา CL อีกทั้ง ภาษาไพธอนนั้นเป็นภาษาระดับสูง General-purpose ไวยากรณ์ของภาษานั้นค่อนข้างที่จะชัดเจน รวมถึงตัวไลบรารีของภาษาไพธอนนั้นมีมากมาย เนื่องจากภาษาไพธอนนั้นเป็นภาษาที่เป็นโอเพ่นซอร์สทำให้เปิดโอกาสให้โปรแกรมเมอร์นั้นสามารถนำไปพัฒนาเป็นไลบรารีต่างๆได้อย่างกว้างขวาง

ภาษาเพื่อการสื่อสาร (Communication Language) หรือ “ภาษา CL” เป็นภาษาคอมพิวเตอร์ในรูปแบบที่ใช้สื่อสารลำดับการทำงานและการติดต่อสื่อสาร แต่ยังสามารถอย่างที่ปรากฏในภาษาคอมพิวเตอร์ทั่วไป เพื่อให้เป็นภาษาที่ง่ายจึงได้นำเอาไวยากรณ์ของภาษาไพธอนมาใช้ในการออกแบบ โดยลักษณะสำคัญของภาษา CL คือสามารถประมวลผลการทำงานต่างๆ ตามกำหนดเวลา อีกทั้งยังสามารถโปรแกรมให้ทำงานตอบสนองต่อเหตุการณ์ (อีเวนต์) ทำให้ภาษา CL มีความยืดหยุ่นและมีการประมวลผลคล้ายคลึงกับการทำงานของมนุษย์

2.1 ภาษาไพธอน

ไพธอน (Python) เป็นภาษาโปรแกรมในลักษณะภาษาอินเตอร์พรีเตอร์โปรแกรมมิ่ง (Interpreted programming language) ผู้คิดค้นคือ Guido van Rossum ในปี 1990 ซึ่งไพธอนเป็นการจัดการชนิดของตัวแปรแบบแปรผันตามข้อมูลที่บรรจุอยู่ (Fully dynamically typed) และใช้การจัดการหน่วยความจำเป็นอัตโนมัติ (Automatic memory management) โดยได้เป็นการพัฒนาและผสมผสานของภาษาอื่น ๆ ได้แก่ ABC, Modula-3, Icon, ANSI C, Perl, Lisp, Smalltalk และ Tcl และภาษาไพธอนยังเป็นแนวคิดที่ทำให้เกิดภาษาใหม่ ๆ ซึ่งได้แก่ Ruby และ Boo เป็นต้น

ไพธอนนั้นพัฒนาเป็นโครงการ โอเพ่นซอร์สโดยมีการจัดการแบบไม่หวังผลกำไรโดยไพธอน Software Foundation และสามารถหาข้อมูลและตัวแปลภาษาได้จากเว็บไซต์ของไพธอนเอง ที่ <http://www.python.org/>

2.2 ความสามารถของภาษาไพธอน

1) ง่ายต่อการเรียนรู้ โดยภาษาไพธอนมีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย ซึ่ง

โครงสร้างภาษาไพธอนจะคล้ายกับภาษาซีมาก เพราะภาษาไพธอน สร้างขึ้นมาโดยใช้

ภาษาซี ทำให้ผู้ที่คุ้นเคยภาษาซี อยู่แล้วใช้งานภาษาไพธอนได้ไม่ยาก นอกจากนี้โดยตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติเห็นาไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาเองมีความยืดหยุ่นสูงทำให้การจัดการกับงานด้านข้อความ และแท็กไฟล์ได้เป็นอย่างดี

- 2) ไม่ต้องเสียค่าใช้จ่ายใดๆ ทั้งสิ้น เพราะตัวแปลภาษาไพธอนอยู่ภายใต้ลิขสิทธิ์ไพธอน Software Foundation License (PSFL) ซึ่งเป็นของไพธอน Software Foundation (PSF) ซึ่งมีลักษณะคล้ายกับ ลิขสิทธิ์แม่แบบอย่าง General Public License (GPL) ของ Free Software Foundation (FSF)
- 3) ใช้ได้หลายแพลตฟอร์ม ในช่วงแรกภาษาไพธอนถูกออกแบบใช้งานกับระบบยูนิกซ์อยู่ก็จริง แต่ในปัจจุบันได้มีการพัฒนาตัวแปลภาษาไพธอน ให้สามารถใช้กับระบบปฏิบัติการอื่นๆ อาทิเช่น Linux Platform, Windows Platform, OS/2, Amiga, Mac OS X และรวมไปถึงระบบปฏิบัติการที่มี .NET Framework, Java virtual machine ทำงานได้ ซึ่งใน Nokia Series 60 ก็สามารถทำงานได้เช่นกัน
- 4) ภาษาไพธอนถูกสร้างขึ้นโดยได้รวบรวมเอาส่วนดีของภาษาต่างๆ เข้ามาไว้ด้วยกัน อาทิเช่น ภาษา ABC, Modula-3, Icon, ANSI C, Perl, Lisp, Smalltalk และ Tcl
- 5) ไพธอนสามารถรวมการพัฒนาของระบบเข้ากับ COM, .NET และ CORBA objects

2.3 ไวยากรณ์ของภาษาไพธอน

ไวยากรณ์ของภาษาไพธอนที่ต้องคำนึงถึงในการเขียนอินเตอร์พรีเตอร์เพื่อให้อินเตอร์พรีเตอร์มีความถูกต้องและได้ผลที่ถูกต้อง โดยจะแบ่งไวยากรณ์เป็น 4 หัวข้อหลักๆ

2.3.1 ชนิดของข้อมูล

ในภาษาไพธอนนั้นชนิดข้อมูลไม่จำเป็นต้องมีการประกาศตัวแปร เนื่องจากตัวอินเตอร์พรีเตอร์จะต้องสามารถทราบได้ว่าชนิดข้อมูลนั้นควรเป็นชนิดอะไร และ จะมีการเปลี่ยนแปลงตามให้โปรแกรมเมอร์ตามความเหมาะสมทำให้การเขียน โปรแกรมนั้นสามารถทำได้ง่ายมากขึ้น โดยชนิดข้อมูลของภาษาไพธอนสามารถแบ่งได้ดังนี้

- 1) ตัวเลข (Numeric) ข้อมูลประเภทตัวเลขได้แก่
 - 1.1) เลขฐานสิบ (Decimal Number) เช่น 1, 2, 8755, 48 เป็นต้น
 - 1.2) เลขฐานแปด (Octal Number)
 - 1.3) เลขฐานสิบหก (Hexadecimal)
 - 1.4) เลขทศนิยม (Floating Point) เช่น 1.2, 55.56, 41e12 เป็นต้น
- 2) บูลีน (Boolean) ข้อมูลแสดงความจริงเท็จ มักแสดงอยู่ในรูปของ true, false, 0, 1
ค่าบูลีนเป็นผลลัพธ์ของนิพจน์ทางตรรกศาสตร์ ซึ่งส่วนมากคือการนำค่าจำนวนสอง

ค่า หรือค่าสตริงสองค่า มาเปรียบเทียบกัน เครื่องหมายที่ใช้ในการเปรียบเทียบในค่า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ภาษาไพทอนนั้นมี 6 เครื่องหมาย ได้แก่ == (เท่ากับ), != (ไม่เท่ากับ), < (น้อยกว่า), > (มากกว่า), <= (น้อยกว่าหรือเท่ากับ), และ >= (มากกว่าหรือเท่ากับ)
- 3) สตริง (String) ข้อมูลประเภทของประโยคหรือชุดของตัวอักษร เช่น “project”, “computer” ตัวอย่างเช่น a = "Hello World", b = 'Python is groovy', c = ""Computer says 'No'"" , b = a[4] # b = 'o'
 - 4) ลิสต์ (list) เป็นข้อมูลซึ่งแทนลำดับของค่าต่างๆ เหมือนที่พีเพิลแต่เราสามารถเปลี่ยนสมาชิกที่ตำแหน่งต่างๆ ของ ลิสต์ ได้ ซึ่งทำให้ลิสต์คล้ายอาเรย์ในภาษา C มากกว่า (tuple) เราสามารถสร้างลิสต์ได้ด้วยการเขียนลำดับของสมาชิกใน ลิสต์ ภายในวงเล็บก้ามปู ตัวอย่างเช่น a = [true, “computer”, 44]
#การอ้างถึงข้อมูลนั้นอ้างถึงคล้ายกับอาเรย์เช่น a[0] = true เป็นต้น
 - 5) ทัฟเปิล (tuple) เป็นโครงสร้างที่คล้ายกับอาเรย์คงที่ เหมือนกับค่าค่าไทป์ประเภท ลิสต์ แต่ในทัฟเปิลนั้นเราไม่สามารถที่จะแปลงค่าในทัฟเปิลได้ทำให้เหมาะกับการใช้ทัฟเปิลในการกำหนดค่าคงที่ต่างๆ ของ โปรแกรม ตัวอย่างเช่น a = (true, “computer”, 44) ซึ่งการอ้างถึงข้อมูลนั้นอ้างถึงคล้ายกับอาเรย์เช่น a[0] = true เป็นต้น
 - 6) ดิกชันนารี (Dictionary) เป็น โครงสร้างข้อมูลที่คล้ายอาเรย์แต่กรณีที่ใช้กับ ดิกชันนารีจะเป็นค่าอะไรก็ได้ ไม่จำเป็นต้องเป็นเลขจำนวนเต็มเท่านั้นเหมือนกับ อาเรย์เราสามารถสร้างดิกชันนารีว่างๆ ที่ไม่เก็บสมาชิกใดเอาไว้เลยได้ด้วยนิพจน์ {} ตัวอย่างเช่น inventory = {430: ‘Apple’, 431: ‘orange’, 432: ‘grape’} ซึ่งการอ้างถึง ข้อมูลนั้นอ้างถึงคล้ายกับอาเรย์แต่การทำงานของดิกชันนารีนั้นจะทำหน้าที่เหมือนกับการเสิร์ชลงบนดิกชันนารีคือ inventory [430] = ‘Apple’

2.3.2 Operator ในภาษาไพธอน

ในการทำงานของภาษาโปรแกรมมิ่งนั้น จะต้องมีโอเปอเรเตอร์เพื่อบอกถึงการกระทำของ 2 ค่าค่าไทป์โดยในภาษาไพธอนนั้นแบ่งเป็น 5 แบบ ดังต่อไปนี้

- 1) Arithmetic Operators เป็น โอเปอเรเตอร์ใช้ในการทำงานทางคณิตศาสตร์ ตัวอย่างเช่น +, -, *, /, mod, ^ เป็นต้น
- 2) Comparison โอเปอเรเตอร์จะทำการเปรียบเทียบค่าระหว่างสองตัวแปร ซึ่งผลลัพธ์ที่ได้จากการเปรียบเทียบคือค่าบูลีนนั่นคือ true และ false ตัวอย่างเช่น ==, ==, <, <, <=, >=, > เป็นต้น
- 3) Logical Operator เป็น โอเปอเรเตอร์ทางตรรกะหรือประพจน์ต่างๆในการ คณิตศาสตร์และทางโลกิจ ตัวอย่างเช่น and, or, not เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) Assignment Operator เป็นโอเปอเรเตอร์พิเศษเพื่อช่วยให้การเขียนโปรแกรมประเภทที่ใช้ Arithmetic Operation ได้สะดวกยิ่งขึ้น ตัวอย่างเช่น =, %=, /=, //, -=, +=, *=, **= เป็นต้น
- 5) Conditional Operator เป็นโอเปอเรเตอร์เงื่อนไขต่างๆ เช่น if else เป็นต้น

2.3.3 คำสงวนในภาษาไพธอน

['and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'exec', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'not', 'or', 'pass', 'print', 'raise', 'return', 'try', 'while', 'with', 'yield']

2.3.4 ประโยคในภาษาไพธอน

- 1) ประโยคเงื่อนไข IF ใช้คำสั่งของ if else หรือ elif ตัวอย่างไวยากรณ์เป็นดังนี้

```
if expression:
    statements
elif expression:
    statements
...
else:
    statements
```

รูป 2.1 ประโยคเงื่อนไข IF

- 2) ประโยคทำซ้ำ WHILE ใช้คำสั่งของ while ตัวอย่างไวยากรณ์เป็นดังนี้

```
while expression:
    statements
```

รูป 2.2 ประโยคทำซ้ำ WHILE

- 3) ประโยคทำซ้ำ FOR ใช้คำสั่งของ for ตัวอย่างไวยากรณ์เป็นดังนี้

```
for i in s:
    statements
```

รูป 2.3 ประโยคทำซ้ำ FOR

2.4 ภาษา CL

ภาษา CL เป็นภาษาคอมพิวเตอร์ที่ใช้สั่งงานคอมพิวเตอร์ตามเวลา และเหตุการณ์ มีจุดเริ่มต้นมาจากการพัฒนาเว็บเบราว์เซอร์ที่สามารถโปรแกรมได้ และพัฒนาต่อเนื่องมาจนเป็นอินเทอร์พรีเตอร์ CL ในการออกแบบภาษา ผู้ออกแบบภาษาได้เล็งเห็นความสำคัญของภาษาที่มีความเป็นนามธรรมสูง และสามารถโปรแกรมได้ง่าย ดังนั้นผู้ออกแบบจึงเลือกภาษาไพธอน เป็นต้นแบบในการพัฒนา ทำให้ไวยากรณ์ของภาษา CL ในส่วนที่ไม่เกี่ยวข้องกับเวลาจะมีลักษณะคล้ายภาษาไพธอน

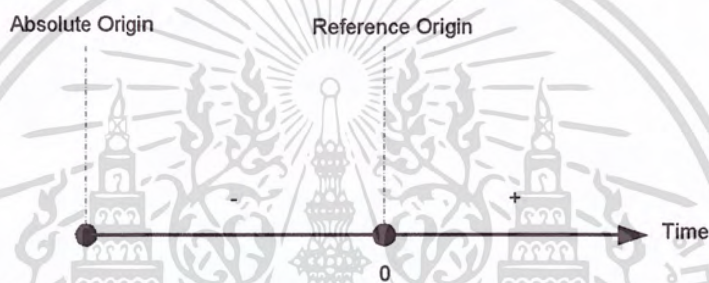
ภาษา CL เกิดจากการเพิ่มขยายภาษาไพธอนให้มี ความสามารถในการประมวลผลเกี่ยวกับเวลาและการสื่อสาร โดยไวยากรณ์หลักๆ ในภาษา CL มีดังนี้

- 1) ประเภทข้อมูล (Data Type) ส่วนใหญ่เป็นประเภทข้อมูลที่มีอยู่ในไพธอนได้แก่ Number, Boolean, string, list, tuple, dictionary และ none ส่วนประเภทข้อมูลใหม่ที่เพิ่มเติมเข้ามาคือ Time
- 2) โอเปอเรเตอร์ (Operators) มีโอเปอเรเตอร์ทางตรรกะ, การเปรียบเทียบ, การคำนวณเลขฐานสอง และโอเปอเรเตอร์ทางคณิตศาสตร์ ซึ่งทั้งหมดมาจากไพธอนส่วนโอเปอเรเตอร์ที่เพิ่มเติมเข้ามาคือ โอเปอเรเตอร์ที่ประมวลผลเกี่ยวกับเวลา
- 3) เอ็กซ์เพรสชัน (Expression) เป็นกลุ่มของโอเปอเรเตอร์ร่วมกับโอเปอเรนด์ ซึ่งเมื่อถูกประเมินผลจะให้ค่าออกมา
- 4) การนิยามฟังก์ชัน ใน CL ผู้ใช้สามารถนิยามฟังก์ชันขึ้นใหม่ได้
- 5) การสั่งงานด้วยเหตุการณ์ อยู่ในรูปแบบของ เหตุการณ์ → การทำงาน หมายถึง เมื่อ “เหตุการณ์” ให้ค่าบูลีนเป็น True อินเทอร์พรีเตอร์ CL จะกระทำ “การทำงาน” ทันที ส่วนนี้ไม่มีในไพธอน
- 6) ประโยค (Statement) มีความแตกต่างจาก ภาษาคอมพิวเตอร์ทั่วไป คือ มีการกำหนดช่วงเวลาทำงานให้กับ ทุกประโยคในรูปแบบของ “ช่วงเวลา: ประโยค” ซึ่งหมายถึง ประโยคจะทำงานในช่วงเวลานี้ โดยรายละเอียดจะขอกกล่าวใน หัวข้อที่ 2.7 ประโยคมีหลายรูปแบบดังนี้
- 7) ประโยคกำหนดค่าให้ตัวแปร จะอยู่ในรูป Variable = Value
- 8) ประโยคคำสั่ง ปัจจุบันมี 2 รูปแบบคือ ช่วงเวลา: function () และช่วงเวลา: module.function () ส่วนรูปแบบ ช่วงเวลา: class.function() จะขยายให้ครอบคลุมในภาษา CL ในอนาคต
- 9) ประโยคควบคุมลำดับการทำงานใน โปรแกรม มี 3 รูปแบบ คือ รูปแบบประโยค if, while และ for

- 10) ประโยคชุดคำสั่ง มีอยู่ 3 แบบ คือ แบบ ซีควนท์, อันออร์เดอร์เรด และ พาราเรล สรุปลแล้วโปรแกรมภาษา CL ประกอบ ด้วยลิสต์ของ คำนิยาม ฟังก์ชัน การทำงานด้วยเหตุการณ์ และ/หรือประโยค

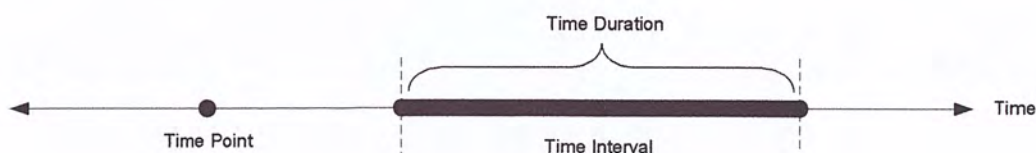
2.5 ความหมายของเวลา

เวลาเกิดจากการดำเนินไปของกระบวนการ (Process) อันหนึ่งอย่างต่อเนื่อง เมื่อไรที่เกิดมีขบวนการ ก็จะเกิดเวลา แต่ถ้าไม่มี เวลาก็ไม่ปรากฏ เนื่องจากการดำเนินไปของขบวนการมีความต่อเนื่อง จึงสมมติให้เวลามีความต่อเนื่องด้วย โดยให้มีค่าเพิ่มขึ้นแบบไม่ขาดตอน แทนด้วยเส้นถูกสรแสดงเป็นแกนของเวลา ดังรูป 2.4



รูป 2.4 ความหมายของเวลา

โดยเวลาจะมีค่าเพิ่มขึ้นเรื่อยๆ เมื่อเทียบกับจุดเริ่มต้นในอุดมคติ ของการเกิดเวลา (Absolute Origin) ที่จริงแล้วเราไม่สามารถรู้ถึงจุดกำเนิดที่แท้จริงของเวลา จึงได้กำหนดจุดเริ่มต้นสมมติของเวลาขึ้นเพื่อให้ง่ายต่อการอ้างอิง ขอเรียกจุดเริ่มต้นนี้ว่า “จุดเริ่มอ้างอิง” (Reference Origin) จุดต่างๆ บนเส้นแกนเวลาแสดงการอ้างอิงเวลา ณ ขณะใดขณะหนึ่ง จุดนี้เรียกว่า จุดเวลา (Time Point) โดยจุดเวลาที่อยู่ทางขวาของจุดเริ่มอ้างอิงมีค่าเป็นบวกและจุดที่อยู่ทางซ้ายมีค่าเป็นลบ ในกรณีที่มีการลากเส้นเชื่อมระหว่าง 2 จุดเวลา จากจุดที่น้อยกว่าเรียกว่า “จุดเริ่มต้น” ไปยังจุดที่มีค่ามากกว่าเรียกว่า “จุดสิ้นสุด” จะทำให้เกิด ช่วงเวลา (Time Interval) ซึ่งประกอบด้วยจุดเวลาหลายๆ จุดเรียงต่อเนื่องกันจาก จุดเริ่มไปจนถึงจุดสิ้นสุด โดยความยาวของช่วงเวลาหรือของ เส้นตรงนี้ถือว่าเป็นระยะเวลา (Time Duration) แสดงดังรูป 2.5



รูป 2.5 ความหมายของจุดเวลา ระยะเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 รูปแบบของเวลา

2.6.1 รูปแบบจุดเวลา

- 1) รูปแบบทางคณิตศาสตร์ ในทางคณิตศาสตร์จุดเวลาจะแสดงในรูปของหน่วยเวลาที่เล็กที่สุดเท่าที่นัยสำคัญจะมีผลกระทบต่อการทำงาน ในที่นี้กำหนดให้เป็นหน่วยวินาที โดยเริ่มนับจำนวนหน่วยเวลาดังแต่จุดเริ่มอ้างอิง รูปแบบนี้จะใช้สำหรับการคำนวณในภาษา CL
- 2) รูปแบบที่ใช้ในชีวิตประจำวัน มีรูปแบบ วัน เดือน ปี ชั่วโมง นาที และวินาที แทนที่จะกำหนดในหน่วยวินาที จึงเหมาะสำหรับการแสดงผลและสื่อสารกับคน
- 3) รูปแบบในภาษา CL มีรูปแบบ hh : minmin : ss | dd / mm / yyyy โดยที่ h, min, s, d, m, y แทน ชั่วโมง, นาที, วินาที, วันที่, เดือน และ ปี ตามลำดับ โดยให้จุดเริ่มอ้างอิงเป็นเที่ยงคืนวันที่ 1 ม.ค. ค.ศ. 1970 (epoch) เช่น 11:00:00|28/08/2002 หมายถึง ณ เวลา 11.00 น. ของวันที่ 28 ส.ค. ค.ศ. 2002 เป็นต้น

รูปแบบจุดเวลาใน CL ใช้รูปแบบชีวิตประจำวัน แต่ในการประมวลผลอินเตอร์พรีเตอร์จะแปลงให้เป็นรูปแบบคณิตศาสตร์ก่อนเพื่อการคำนวณ จากนั้นผลจากการคำนวณจะถูกแปลงกลับให้อยู่ในรูปแบบเดิมเพื่อการแสดงผล

2.6.2 รูปแบบช่วงเวลา (Time Interval)

- 1) รูปแบบทางคณิตศาสตร์ เรากำหนดให้ช่วงเวลาอยู่ในรูปแบบของเซตต่อเนื่องที่มีสมาชิกเป็นจุดเวลา คือ $[Ps, Pe]$ โดยที่ Ps คือ จุดเวลาเริ่มต้นมีค่าน้อยกว่า Pe จุดเวลาสิ้นสุด หรือ มีรูปแบบ $[Ps, Ps + d]$ โดย d คือระยะเวลา และ $d > 0$ หรือ มีรูปแบบของเซตต่อเนื่อง $\{t | Ps, t | Pe\}$
- 2) รูปแบบในภาษา CL มีรูปแบบ $<Ps, Pe>$ โดยที่ Ps และ Pe คือ ข้อมูลจุดเวลาเริ่มต้นและสิ้นสุดตามลำดับ สำหรับรูปแบบในภาษา CL เราจะใช้สัญลักษณ์ $<...>$ แทน [...] อย่างที่ใช้ในรูปแบบคณิตศาสตร์ ตัวอย่างเช่น $<11:00:00|28/08/2002, 12:00:00|28/08/2002>$

2.6.3 รูปแบบระยะเวลา (Time Duration)

- 1) รูปแบบทางคณิตศาสตร์ เหมือนกับรูปแบบของจุดเวลาในทางคณิตศาสตร์ คือจะใช้หน่วยวินาทีในการกำหนดระยะเวลา
- 2) รูปแบบในภาษา CL มีรูปแบบ Number1 TimeUnit1, Number2 TimeUnit2, ... โดยที่ Number คือ เลขจำนวนเต็มหรือเลขทศนิยม และ TimeUnit คือ หน่วยของเวลาซึ่งประกอบด้วย หน่วยวัน (d), ชั่วโมง (h), นาที (min) และวินาที (s) เช่น 3d,4h หมายถึงระยะเวลา 3 วัน 4 ชม.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 คำสั่งการทำงานในรูป “ช่วงเวลา: การทำงาน”

ใน CL ได้กำหนดรูปแบบคำสั่งการทำงานหรือรูปแบบประโยคการทำงานเป็น $[Ps, Pe]$: statement หมายถึงสแตทเม้นท์ที่ทำงานระหว่างจุดเวลา Ps และ Pe โดย Ps และ Pe อาจจะเป็นเอ็กเพรสชันใดๆ ที่ให้ค่าจุดเวลาออกมา

2.7.1 ประเภทของคำสั่งการทำงาน

คำสั่งการทำงานสามารถแบ่งเป็น 3 แบบ ดังนี้

- 1) แบบระบุเวลา $[Ps, Pe]$: statement เป็นคำสั่งที่กำหนดเวลาที่แน่นอนในการเริ่มต้นและสิ้นสุดการทำงาน
- 2) แบบกึ่งระบุเวลา $[, Pe]$: statement หรือ $[Ps,]$: statement เป็นคำสั่งที่กำหนดจุดเริ่มต้นหรือจุดสิ้นสุดที่แน่นอนในการทำคำสั่ง และไม่ระบุจุดเวลาที่เหลือ
- 3) แบบไม่ระบุเวลา $[,]$: statement เป็นคำสั่งที่ไม่มีการระบุทั้งจุดเริ่มต้นและจุดสิ้นสุดการทำงาน แต่ปล่อยให้จุดเวลาทั้งสองกำหนดโดยระบบขณะทำคำสั่ง ทำให้คำสั่งแบบนี้เหมือนกับคำสั่งในภาษาคอมไพเตอร์ทั่วไป

2.7.2 ผลการทำ Action ของอินเตอร์พรีเตอร์

เป็นได้ 2 ลักษณะ คือ ผลสำเร็จ (Success) หมายถึง การประมวลผลตามคำสั่งการทำงานได้อย่างครบถ้วน ผลที่ส่งกลับจะเป็นค่าข้อมูล, none, หรือเป็นผลกระทบ (Side-effect) และ ผลล้มเหลว (Failed) หมายถึง การประมวลผลตามคำสั่งนั้นได้ไม่สมบูรณ์ ผลลัพธ์ที่ได้จะอยู่ในรูปแบบของ Exception

2.8 ชุดคำสั่งภาษา CL

2.8.1 ชุดคำสั่งแบบ ซีควนท์

เป็นการทำงานตามคำสั่งทีละคำสั่งอย่างเป็นลำดับ โดยคำสั่งถัดไปจะทำงานได้ เมื่อคำสั่งก่อนหน้าได้ทำงานเสร็จแล้วเท่านั้น มีรูปแบบดังนี้

```
{ \ [ s1,e1 ]: statement1
...
\ [ sN,eN ]: statementN }
```

รูป 2.6 ชุดคำสั่งแบบซีควนท์

โดยที่ S_i และ e_i เป็นจุดเวลาเริ่มและสิ้นสุดของแต่ละสแตทเม้นท์ i และ

$S_1 < e_1 < \dots < S_n < e_n$ ซีควนท์ แบ่งเป็น 2 แบบคือ แบบต่อเนื่อง (Continuous) ที่ซึ่งสแตทเม้นท์
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนักเรียนเห็นใบเซอร์เช็คนัดในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังทำงานต่อเนื่องจากสแตทเมนต์ก่อนหน้านี้และแบบ ขาดตอน (Discrete) ที่ซึ่งสแตทเมนต์หลังไม่
ต้องทำงานต่อเนื่องจากสแตทเมนต์ก่อนหน้านี้

2.8.2 ชุดคำสั่งแบบ อันออร์เคอเรด

เป็นชุดคำสั่งที่ไม่ได้ให้ความสำคัญต่ลำดับการทำงานของแต่ละสแตทเมนต์ โดยสแตท
เมนต์ใดจะทำก่อนหรือหลังก็ได้ แต่ให้ทำได้ทีละ 1 ในแต่ละครั้ง มีรูปแบบดังนี้

{ # [_ _]: statement1	{ # statement1
...	OR
# [_ _]: statementN }	# statementN }

รูป 2.7 ชุดคำสั่งแบบอันออร์เคอเรด

โดยใช้ “#” นำหน้าทุกสแตทเมนต์ และช่วงเวลาที่กำกับหน้าทุกสแตทเมนต์จะต้องเป็น
แบบไม่ระบุเวลาเริ่มต้นและสิ้นสุดเท่านั้น

2.8.3 ชุดคำสั่งแบบ พาราแรล

เป็นชุดคำสั่งที่แต่ละคำสั่งจะทำงานไปพร้อมๆ กันอย่างอิสระมีรูปแบบดังนี้

{ \ [s1,e1]: statement1
...
\ [sN,eN]: statementN }

รูป 2.8 ชุดคำสั่งแบบพาราแรล

โดยที่ช่วงเวลาของแต่ละสแตทเมนต์มีค่าที่เป็นอิสระต่อกัน แต่จุดเวลาเริ่มต้นของทุก
คำสั่งจะต้องมากกว่าหรือเท่ากับจุดเวลา ณ ขณะที่ชุดคำสั่งนี้ทำงาน และเราใช้ “\” นำหน้าทุก สแตท
เมนต์

2.8.4 ชุดคำสั่งแบบ Nested

ให้ถือว่าชุดคำสั่งแบบ ซีควนท์, อันออร์เคอเรด และ พาราแรล เป็นเสมือน 1 สแตท
เมนต์ดังนั้นการสร้างชุดคำสั่งแบบ nested ทำได้โดยใส่ชุดคำสั่งทั้งสามเหมือน 1 สแตทเมนต์ไว้
ภายในชุดคำสั่งแบบ ซีควนท์, อันออร์เคอเรด, หรือ พาราแรล ที่สำคัญ ผู้ใช้ต้องยึดถือเงื่อนไขของ
แต่ละประเภทชุดคำสั่งอย่างเคร่งครัด ได้แก่ ในชุดคำสั่งแบบ อันออร์เคอเรด แต่ละสแตทเมนต์
ภายใน จะไม่มีการระบุเวลาเริ่มต้นและสิ้นสุด เป็นต้น ตัวอย่างเช่น

```

{ \ [ s1,e1 ]: statement1
  \ { # statement2
    # { [ _,_ ]: statement3
      [ _,_ ]: statement4 }
    }
  }
}

```

รูป 2.9 ชุดคำสั่งแบบ Nested

2.8.5 ช่วงเวลารวมชุดคำสั่ง

เราสามารถกำหนดช่วงเวลาเริ่มต้นและสิ้นสุดของการทำงานของทั้งชุดคำสั่งได้ใน

รูปแบบ $\langle Ps, Pe \rangle$: ชุดคำสั่ง

2.9 การนิยามฟังก์ชันในภาษา CL

มีรูปแบบการนิยามฟังก์ชันดังนี้

```

def function(arg1,arg2,...):
  definition

```

รูป 2.10 การนิยามฟังก์ชันในภาษา CL

โดย definition อาจเป็น $\langle Ps, Pe \rangle$: statement หรือ $\langle Ps, Pe \rangle$:ชุดคำสั่ง ซึ่งหมายถึงว่า ฟังก์ชันนี้เริ่มทำงาน ณ เวลา Ps และสิ้นสุดการทำงานที่เวลา Pe ดังนั้น การเรียกใช้งาน ฟังก์ชันนี้จะ เป็นรูปแบบ $\langle Ps, Pe \rangle$: function(arg1, arg2, ...)

2.10 การสั่งงานด้วยเหตุการณ์

เหตุการณ์ คือ ข่าวสารที่เกิดจากสภาพแวดล้อมภายนอก (Environment) ไม่ได้เกิดจากชุดคำสั่ง เราสามารถใช้การเกิดขึ้นของเหตุการณ์ในการขัดจังหวะ (interrupt) อินเตอร์พรีเตอร์ให้มาทำคำสั่งตามที่เราต้องการได้ เวลาที่เกี่ยวข้องกับเหตุการณ์คือ จุดเวลาที่เหตุการณ์นั้นเกิด แต่เพื่อให้ตัวจัดการเหตุการณ์ (อีเวนท์ Listener) ในภาษา CL สามารถทำงานได้อย่างมีประสิทธิภาพ ผู้เขียน โปรแกรมสามารถระบุช่วงเวลาที่เกิดกับไว้กับทุกๆ เหตุการณ์ได้ ทำให้รูปแบบเหตุการณ์เป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ได้รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Ps,Pe] : event

รูป 2.11 รูปแบบระบุช่วงเวลาที่เกิดเหตุการณ์นั้นจะเกิดขึ้น

ซึ่งส่วนที่ทำหน้าที่จัดการเหตุการณ์ จะตรวจสอบอีเวนที่ระหว่าง จุดเวลา Ps ถึงจุดเวลา Pe เท่านั้น ซึ่งช่วงเวลานี้สามารถระบุเป็น แบบระบุเวลา, แบบกึ่งระบุเวลา และแบบไม่ระบุเวลา เหตุการณ์ในภาษา CL มี 2 ชนิดคือ เมลอีเวนท์ และอีเวนท์ อื่นๆ

เราสามารถโปรแกรมการทำงานเพื่อตอบสนองต่อเหตุการณ์ที่เกิดขึ้นด้วยเงื่อนไข

[Ps,Pe] : event → action

รูป 2.12 รูปแบบการระบุการทำงานเพื่อตอบสนองเหตุการณ์เกิดขึ้นด้วยเงื่อนไข

ซึ่งหมายถึง เมื่อมีอีเวนท์เกิดขึ้นในช่วงเวลา [Ps,Pe] ระบบจะทำ Action ตามที่กำหนด โดย Action คือ คำสั่งการทำงานที่เป็น statement ใดๆ

2.11 สรุปภาษาไพธอน

ภาษาไพธอนนั้นเป็นภาษาที่มีไวยากรณ์ที่ง่าย และ ง่ายต่อการเขียน โปรแกรมและเหมาะสมกับการนำไปประยุกต์ร่วมกับ CL เพื่อให้โปรแกรมเมอร์ทั่วไปสามารถเขียน โปรแกรมได้อย่าง สะดวกมากขึ้น

2.12 สรุปภาษา CL

ภาษา CL เป็นภาษาที่ใช้สั่งงานคอมพิวเตอร์ตามกำหนดเวลา และเหตุการณ์ โดยมีรูปแบบของภาษาด้ายกับภาษาไพธอน แต่มีการเพิ่มช่วงเวลาเข้าไปในแต่ละคำสั่ง รวมทั้งประโยชน์สำหรับรองรับเหตุการณ์ เพื่ออำนวยความสะดวกในการเขียน โปรแกรมยิ่งขึ้น

บทที่ 3

ภาษาจาวาสคริปต์

ภาษาจาวาสคริปต์นั้นสามารถนำมาใช้เพื่อให้อินเทอร์เน็ตเบราว์เซอร์นั้นสามารถทำงานได้อย่างมีประสิทธิภาพ และช่วยให้อินเทอร์เน็ตเบราว์เซอร์ทำงานบนเว็บเบราว์เซอร์ได้ อีกทั้งยังสามารถทำการเขียนโปรแกรมลงไปได้อย่างมีประสิทธิภาพเหมือนกับแอปพลิเคชันโปรแกรมทั่วไปซึ่งจะทำให้เว็บแอปพลิเคชันนั้นสามารถทำงานได้เหมือนกับโปรแกรมที่ฝังของไคลเอนต์เซิร์ฟเวอร์ซึ่งรายละเอียดของภาษาจาวาสคริปต์นั้นจะกล่าวถึงต่อไป

โดยบทนี้จะกล่าวถึง ข้อมูลทั่วไปของภาษาจาวาสคริปต์, ความสามารถต่างๆของภาษาจาวาสคริปต์ และเฟรมเวิร์ค ที่ช่วยให้ ภาษาจาวาสคริปต์ทำงานอย่างมีประสิทธิภาพมากขึ้น โดยมีรายละเอียดดังต่อไปนี้

3.1 ภาษาจาวาสคริปต์

ภาษาจาวาสคริปต์ (JavaScript) เป็นภาษาสคริปต์ ที่มีลักษณะการเขียนแบบ โปรโตไทป์ (Prototyped-based Programming) ส่วนมากใช้ในหน้าเว็บเพื่อประมวลผลข้อมูลที่ฝั่งของผู้ใช้งาน แต่ก็ยังมีใช้เพื่อเพิ่มเติมความสามารถในการเขียนสคริปต์โดยฝังอยู่ในโปรแกรมอื่นๆ

จาวาสคริปต์มีโครงสร้างของภาษาและไวยากรณ์อยู่บนพื้นฐานของภาษาซีปัจจุบันมีการใช้จาวาสคริปต์ที่ฝังอยู่ในเว็บเบราว์เซอร์ในหลายรูปแบบ เช่น ใช้เพื่อสร้างเนื้อหาที่เปลี่ยนแปลงเสมอภายในเว็บเพจ, ใช้เพื่อตรวจสอบความถูกต้องของข้อมูลที่ผู้ใช้กรอกก่อนนำเข้าระบบ, ใช้เพื่อเข้าถึงข้อมูลที่อยู่ภายใต้โครงสร้างแบบ Document Object Model (DOM) เป็นต้น

นอกจากนี้จาวาสคริปต์ยังถูกฝังอยู่ในแอปพลิเคชันต่างๆ นอกเหนือจากเว็บเบราว์เซอร์ได้อีกด้วย โดยรวมแล้วจาวาสคริปต์ถูกใช้เพื่อให้นักพัฒนาโปรแกรม สามารถเขียนสคริปต์เพื่อสร้างคุณสมบัติพิเศษต่างๆ เพิ่มเติมจากที่มีอยู่บนแอปพลิเคชันดั้งเดิม

โดยภาษาจาวาสคริปต์นั้นมี บริษัท Sun Microsystems เป็นเจ้าของเครื่องหมายการค้า จาวาสคริปต์โดยมันถูกนำไปใช้ภายใต้สัญญาอนุญาตเพื่อการพัฒนาเทคโนโลยีโดย Netscape

3.2 เหตุผลที่เลือกใช้ความสามารถของภาษาจาวาสคริปต์

เนื่องจากเราต้องการที่จะทำให้อินเทอร์เน็ตเบราว์เซอร์มีความสามารถในการทำงานบนเว็บเบราว์เซอร์ ทำให้เราต้องการภาษาที่สามารถทำงานบนเว็บเบราว์เซอร์ และต้องมีความสามารถของ Object oriented model เนื่องจากเราต้องการสร้างอินเทอร์เน็ตเบราว์เซอร์ทำให้ต้องการ Object oriented มารองรับซึ่งภาษาจาวาสคริปต์มีสามารถทั้งนั้นได้อย่างครบถ้วน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

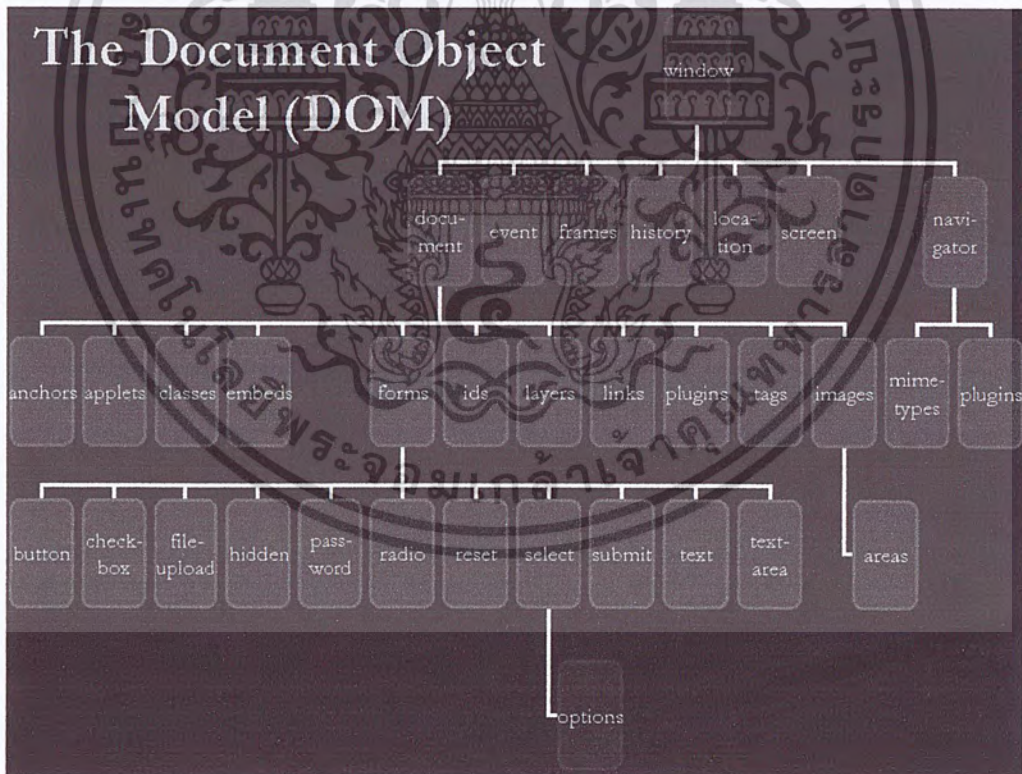
3.3 DOM (Document Object Model)

Document Object Model (DOM) เป็นวิธีการที่เว็บเบราว์เซอร์อธิบายว่าข้อมูลต่างๆ ในเอกสาร HTML หรือ XML จัดเรียงตัวแบบเชิงวัตถุ (object oriented) อย่างไร DOM เป็นวิธีการสร้าง API ให้สามารถควบคุมเนื้อหา โครงสร้าง และรูปแบบของเอกสารได้

เดิมทีนั้นเว็บเบราว์เซอร์แต่ละค่ายมีการพัฒนา DOM โดยใช้ภาษาสคริปต์ที่แตกต่างกันออกไป ทาง W3C จึงได้ร่างมาตรฐานกลางที่เรียกว่า W3C Document Object Model (W3C DOM) โดยไม่ขึ้นกับภาษาโปรแกรมมิ่งภาษาใดภาษาหนึ่ง

การจัดข้อมูลในรูปแบบ DOM นั้นใช้แผนภูมิต้นไม้ในการบอกความสัมพันธ์ระหว่างวัตถุแต่ละชั้น อย่างไรก็ตาม DOM มีจุดอ่อนในเรื่องประสิทธิภาพเมื่อต้องประมวลผลเอกสารขนาดใหญ่ ซึ่ง DOM จำเป็นต้องอ่านเอกสารให้ครบทั้งหมดก่อน จึงจะเริ่มประมวลผล ข้อเสียนี้จึงเกิดการออกแบบ SAX ที่ใช้วิธีอ่านเอกสารทีละส่วนแทน

ตามมาตรฐานของ W3C นั้น ได้แบ่ง DOM เป็นระดับชั้น (level) ที่มีข้อกำหนดแตกต่างกันออกไป ปัจจุบันมี Level 0-3



รูป 3.1 รูปแบบของ DOM ในแต่ละระดับชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 Framework ของ ภาษาจาวาสคริปต์

ภาษาจาวาสคริปต์นั้นเนื่องจากเป็นภาษาที่จัดการด้วย DOM บางครั้งอาจทำให้การจัดการทำได้ไม่ถนัด อีกทั้งการออกแบบในส่วนของผู้สเซอร์อินเตอร์เฟสนั้นหากต้องการให้ได้ผลที่มีประสิทธิภาพนั้นควรมีเฟรมเวิร์ค เข้ามาช่วย โดยเฟรมเวิร์ค ต่างๆนั้นประกอบด้วย

3.4.1 jQuery

jQuery คือไลบรารีของโค้ดจาวาสคริปต์ ซึ่งเป็นส่วนหนึ่งของการสร้างเว็บไซต์ jQuery ช่วยให้เราเรียกเขียนหรือ ใช้งานจาวาสคริปต์ และ Ajax ให้ง่ายขึ้น โดยหากมองถึงความสามารถของ jQuery นั้นสามารถแบ่งออกได้ดังนี้

- 1) ใช้งาน DOM element โดยการเขียนไม่จำเป็นต้องยึดติดกับบราวเซอร์
- 2) จัดการกับอีเวนท์เพิ่มอีเวนท์ลงในหน้าเว็บ ได้อย่างง่ายมากยิ่งขึ้น
- 3) การจัดการ CSS สามารถเพิ่ม ลบ class, id ของ CSS
- 4) เอฟเฟค และอนิเมชันลูกเล่นบนหน้าเว็บ
- 5) Ajax ช่วยให้เขียน Ajax ได้ง่ายขึ้นประหยัดเวลาในการเขียน Ajax และทำให้โค้ดสั้นลง
- 6) สามารถสืบทอดคุณสมบัติของ jQuery ไปใช้งานต่อได้
- 7) สามารถทำผู้สเซอร์อินเตอร์เฟสใหม่ บนบราวเซอร์ ได้

การนำ jQuery ไปใช้นั้นจะนำไปใช้ในส่วนของผู้สเซอร์อินเตอร์เฟส และการจัดการ DOM ในหน้าเว็บ

3.4.2 MOOTOOLS

MOOTOOLS คือ ไลบรารีภาษาจาวาสคริปต์ ซึ่งช่วยให้การเขียนภาษาจาวาสคริปต์นั้นเป็นในเชิงของภาษาออบเจ็คโอเรียนเต็ดได้อย่างสมบูรณ์แบบมากขึ้นอีกทั้ง MOOTOOLS เฟรมเวิร์ค นั้นยังช่วยโปรแกรมเมอร์ในการประยุกต์ใช้ DOM เข้ากับภาษาออบเจ็ค โอเรียนเต็ดได้อย่างลงตัว อีกทั้งยังสามารถใช้ร่วมกับคำท้าไทป์อื่นๆของภาษาจาวาสคริปต์ได้อย่างดี โดยหากจะมองความสามารถของ MOOTOOLS นั้นสามารถแบ่งออกได้ดังนี้

- 1) MOOTOOLS นั้นจัดการแบ่งแยกโมดูลของเฟรมเวิร์ค อย่างชัดเจนทำให้โปรแกรมเมอร์สามารถจัดการส่วนต่างๆได้อย่างอิสระ
- 2) MOOTOOLS ช่วยให้ ภาษาจาวาสคริปต์นั้นจัดการง่ายขึ้นเนื่องจากการเปลี่ยนแปลงภาษาจาวาสคริปต์ให้อยู่ในรูปของออบเจ็ค โอเรียนเต็ด
- 3) MOOTOOLS สามารถเพิ่มประสิทธิภาพการจัดการ DOM ช่วยให้โปรแกรมเมอร์สามารถเพิ่ม, แก้ไข, เลือก, ลบองค์ประกอบของ DOM ง่ายยิ่งขึ้น

บทที่ 4

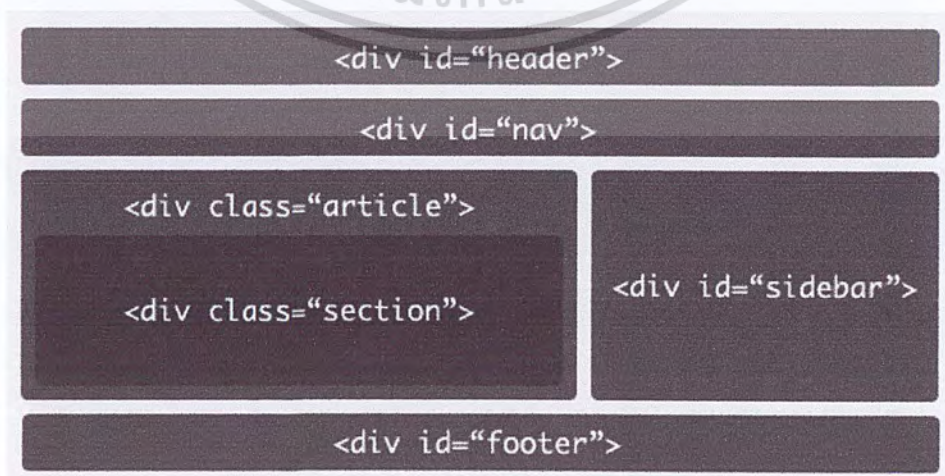
HTML 5

HTML5 เป็นมาตรฐานตัวต่อไปของ HTML ที่อยู่ในระหว่างการพัฒนาแต่ในปัจจุบันได้ออกมาแล้วในบางส่วน โดยมีลักษณะเหมือนมาตรฐานตัวก่อนหน้าทั้ง HTML 4.01 และ XHTML 1.1 ที่ใช้ในการจัดโครงสร้างและการแสดงผลของเนื้อหาสำหรับเว็บไซต์มาตรฐานใหม่จะมีคุณลักษณะเด่นที่สำคัญได้แก่ การใช้งาน วิดีโอ การแสดงตำแหน่งทางภูมิศาสตร์(Geolocation) การเก็บไฟล์ในลักษณะออฟไลน์ การมีดาต้าเบสในทางฝั่งของผู้ใช้ (Local storage) และ การแสดงกราฟิกส์ โดยคุณสมบัติเด่นหลายอย่างไม่จำเป็นต้องติดตั้งซอฟต์แวร์เพิ่ม เช่น Adobe flash player, ActiveXControl ฯลฯ

โดยในบทนี้จะกล่าวเกี่ยวกับฟีเจอร์ใหม่ของ HTML5 ที่นำมาใช้ในการทำวิจัยซึ่ง ได้แก่ การเก็บไฟล์ในลักษณะของ OFFLINE MODE, การมีดาต้าเบสในทางฝั่งของผู้ใช้ (local storage) และ web socket

4.1 ภาษา HTML5

HTML5 ได้มีการแนะนำ เอลเมนต์ใหม่หลายตัวเพื่อตอบสนองการใช้งานของเว็บไซต์รุ่นใหม่ โดยเอลเมนต์ใหม่ส่วนหนึ่งเป็นซีเมนติกทดแทนการใช้งานของบล็อกทั่วไป (<div>) และเอลเมนต์อินไลน์ () ยกตัวอย่างเช่น <nav> (บล็อกสำหรับเมนูบอกทาง) และ <footer> (ส่วนด้านล่างของเว็บเพจ) เอลเมนต์ส่วนอื่นแสดงถึงการใช้งาน เช่น เอลเมนต์ทางด้านสื่อ <audio> และ <video>เอลเมนต์บางตัวที่ตกຸ່ນสำหรับ HTML 4.01 ได้ถูกยกเลิก เช่น และ <center> ซึ่งถูกทดแทนด้วยการทำงานผ่าน CSS



รูป 4.1 ตัวอย่างเอลเมนต์ใหม่ใน HTML5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่อผู้ผู้ใดนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ไวยากรณ์ของ HTML5

ใน HTML5 มีการเปลี่ยนแปลงไวยากรณ์ใหม่ และเพิ่มไวยากรณ์บางส่วนเข้าไป อีกทั้งยังมีการเพิ่ม elements ใหม่ เช่น Markup elements, Media elements, Canvas element และ Attribute values

ตาราง 4.1 tag HTML5 ส่วนของการจัดการหน้าเว็บ

Tag	Description
<article>	ใช้สำหรับเนื้อหาภายนอกต่างๆ เช่น ข้อความจากข่าว หรือ blog แหล่งข้อมูลต่างๆ จากข้อมูลภายนอก
<aside>	สำหรับเนื้อหาโดยรอบของเนื้อหาหลัก โดยส่วนใหญ่แล้วจะอยู่ข้างๆ
<command>	Tag ปุ่มใหม่ใน HTML5 เช่น ปุ่มธรรมดา ปุ่มประเภท radio ปุ่มประเภท checkbox
<details>	ใช้อธิบายความหมายต่างๆ ของแต่ละเอกสารที่จัดทำขึ้น
<summary>	เป็นส่วนสรุปเมื่อจบ ส่วนใหญ่ใช้กับ Tag details
<figure>	จัดกลุ่มเนื้อหาประเภทของ standalone เช่น Video
<figcaption>	ใช้กับคำบรรยายได้ภาพหรือคำอธิบายแต่ละภาพ
<footer>	ใช้เป็นส่วนของท้ายหน้ากระดาษ โดยส่วนใหญ่แล้วจะเป็นส่วนของ Credit คนสร้างหรือโฆษณาต่างๆ
<header>	สำหรับส่วนของหน้ากระดาษหรือส่วนแนะนำต่างๆ ของหน้านั้นๆ
<hgroup>	สร้างกลุ่มให้กับส่วนของ Header
<mark>	ใช้เมื่อต้องการ Highlight Text
<meter>	ใช้ในเชิงการวัดตัวอักษรใน Element ต่างๆ เช่น ค่าสูงสุดของตัวอักษรหรือค่าต่ำสุดของตัวอักษร
<nav>	ใช้ในส่วนของ navigator
<progress>	สถานะการทำงานเพื่อแสดงว่าการทำงานของหน้าเพจนั้นทำงานไปแล้วถึงเท่าไร
<ruby>	ใช้ในส่วนของ การสร้างบันทึกย่อ
<rt>	ใช้ในการอธิบายเพิ่มเติมของบันทึกย่อที่สร้างขึ้นจาก Tag ruby
<rp>	แสดงส่วนที่เว็บเบราว์เซอร์ไม่ Support ในส่วนของ Tag ruby
<section>	กำหนดส่วนต่างๆ ของหน้าเพจ ไม่ว่าจะเป็น Header Footer หรือ Navigator
<time>	ใช้สำหรับแสดง Date time ต่างๆ
<wbr>	ใช้เมื่อต้องการ Break ตัวอักษรใน line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 4.2 tag HTML5 ส่วนของการจัดการสื่อ multimedia และ ส่วนของการจัดการกราฟฟิก

Tag	Description
<audio>	Tag ที่ใช้ในการอ่านไฟล์ประเภทของเสียง
<video>	Tag ที่ใช้ในการอ่านไฟล์ประเภทของเสียงและภาพ
<source>	Tag ระบุตำแหน่งที่จะให้ Tag audio หรือ Tag Video นำไฟล์มาอ่าน
<embed>	Tag สำหรับเรียก plug-in
<canvas>	ใช้ในการเรียกพื้นที่กราฟฟิกหรือพื้นที่รูปวาดต่างๆ

ตาราง 4.3 tag HTML5 ส่วนของการจัดการ input และ output

Tag	Description
<datalist>	ตัวเลือกหรือองค์ประกอบต่างๆภายใน Tag input
<keygen>	สร้าง key ในการยืนยันตัวตนแก่ผู้ใช้
<output>	สำหรับ Type ที่เป็น Dynamic Type เพื่อทำการออกค่า output

ตาราง 4.4 HTML5 ส่วนของประเภทของ input

Type	Description
tel	ข้อมูลที่ input ประเภทเบอร์โทรศัพท์
search	ชนิดของ input เมื่อใส่เข้าไปในส่วนของ input ประเภท search
url	ข้อมูลที่ input ประเภท URL
email	ข้อมูลที่ input ประเภท EMAIL
datetime	ข้อมูลที่ input ประเภท DATETIME
date	ข้อมูลที่ input ประเภท DATE
month	ข้อมูลที่ input ประเภทเดือน
week	ข้อมูลที่ input ประเภทสัปดาห์
time	ข้อมูลที่ input ประเภทเวลา
datetime-local	ข้อมูลที่ input ประเภทเวลาที่ท้องถิ่น
number	ข้อมูลที่ input ประเภทตัวเลข
range	ข้อมูลที่ input ประเภทช่วงของตัวเลข
color	ข้อมูลที่ input ประเภทสีอาจจะเป็นเลขฐานสิบหกหรือเป็นอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

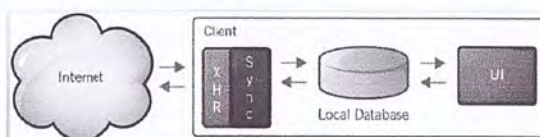
4.3 เหตุผลที่เลือกใช้ความสามารถของ HTML5

เนื่องจากเราต้องการใช้ความสามารถของ offline web applications ในการทำงานได้เมื่อไม่สามารถใช้งานอินเทอร์เน็ต หรือเมื่อขณะทำการใช้งานอินเทอร์เน็ตฟรีเตอร์ก็จะได้ทำงานอย่างต่อเนื่อง ส่วนการใช้ความสามารถของ local storage เราจะใช้ในการเก็บข้อมูลของทั้ง CL อินเทอร์เน็ตไว้ภายในเครื่องของไคลเอนท์เพื่อที่จะทำให้เราสามารถใช้งานข้อมูลโดยไม่ต้องพึ่งพาการเก็บข้อมูลของเซิร์ฟเวอร์ อีกทั้งยังมีหลายหัวข้อใหม่ๆที่น่าสนใจเพื่อเพิ่มความสามารถในการทำงานของเว็บแอปพลิเคชันได้ในบางส่วน เช่น การทำงานกับไฟล์ประเภทของวีดีโอและเสียงสามารถทำงานได้โดยที่ไม่ต้องทำการติดตั้งตัวเสริม

4.4 Offline web applications

ก็คือการที่เราสามารถใช้งานเว็บแอปพลิเคชันได้โดยที่ไม่จำเป็นต้องมีการเชื่อมต่ออินเทอร์เน็ต ซึ่งก็จะทำให้คุณสามารถอ่านอีเมล หรือ ทำการดูข้อมูลบางอย่างได้โดยที่ไม่ต้องต่ออินเทอร์เน็ต โดยการทำงานจะทำงานผ่าน Cache API ที่ได้ถูกเพิ่มเข้ามาใน HTML5 ลักษณะเด่นของ Offline web application

- 1) ทำให้ยูสเซอร์สามารถนำข้อมูลมาจากฝั่งเซิร์ฟเวอร์ได้โดยไม่ต้องทำการร้องขอเรื่อยๆ ช่วยในทางเรียกบีลิตี้เนื่องจากเมื่อเวลาเราไม่สามารถเชื่อมต่ออินเทอร์เน็ตได้แต่หน้าเว็บเรานั้นยังสามารถทำงานต่อได้เมื่อเราเชื่อมต่ออินเทอร์เน็ตได้จึงนำข้อมูล อัพเดทกับทางเซิร์ฟเวอร์อีกครั้ง
- 2) เนื่องจากในปัจจุบันนั้นเครื่องไคลเอนต์นั้นมีฮาร์ดแวร์ที่ถือว่าทำงานได้อย่างมีประสิทธิภาพการทำ offline web application นั้นช่วยให้การอินเตอร์แอคระหว่างยูสเซอร์กับ หน้าเว็บนั้นทำที่ฝั่ง ไคลเอนต์เลย ช่วยให้การงานนั้นรวดเร็ว อีกทั้งช่วยลดจำนวนครั้งในการแอคเซสเข้าสู่เซิร์ฟเวอร์ทำให้ช่วยลดโหลดของเซิร์ฟเวอร์ได้ส่วนหนึ่ง
- 3) เนื่องจากยูสเซอร์ส่วนใหญ่นั้นชอบที่ใช้แอปพลิเคชัน โปรแกรมมากกว่าเว็บแอปพลิเคชัน เนื่องจาก การทำงานที่เร็วกว่าและเสถียรกว่า offline web application ช่วยให้เว็บแอปพลิเคชันสามารถทำงานได้เหมือนกับแอปพลิเคชัน โปรแกรมมากยิ่งขึ้น



รูป 4.2 Offline web application architecture

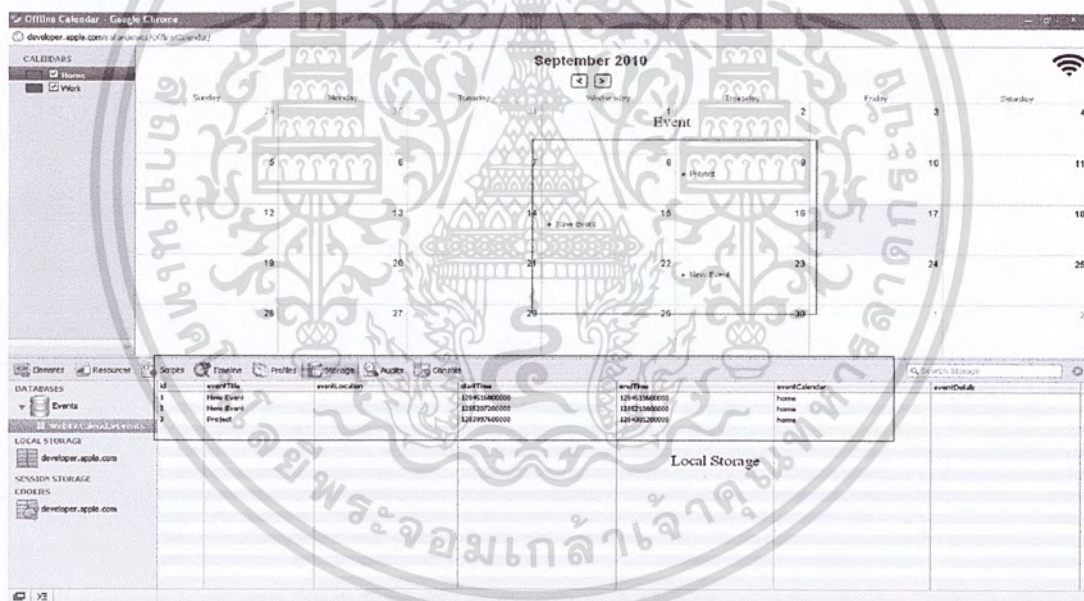
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 Local storage

คือ การที่เว็บเบราว์เซอร์สามารถมี local database ได้ เพื่อใช้เก็บข้อมูลการทำงานของโปรแกรมซึ่งดีกว่าการที่เราต้องมีการไปเก็บที่เซิร์ฟเวอร์ทุกครั้ง อีกทั้งช่วยในการพัฒนาการจดจำเครื่อง ไคลเอนต์ให้กับเซิร์ฟเวอร์แทนที่จะใช้คุกกี้ซึ่งมีขนาดเล็กกว่าเพียง 4 KB เท่านั้น

โดยโปรแกรมเมอร์ที่ใช้เขียน โปรแกรมนั้นสามารถเก็บข้อมูลลงใน local storage ได้เลยและเป็นการช่วยในการจัดการข้อมูลต่างๆ ในการนำมาใช้ ช่วยให้ข้อมูลที่เก็บในสามารถเก็บได้อย่างเป็นระบบมากขึ้นลักษณะเด่นของ local storage

- 1) เนื่องจากสามารถนำมาใช้แทนคุกกี้จึงทำให้สามารถเก็บอินฟอร์มเมชันได้เพิ่มขึ้นมาก
- 2) ข้อมูลสามารถจัดเก็บได้อย่างเป็นระบบด้วยคำสั่งภาษา SQL เช่น insert, delete ฯลฯ
- 3) การได้มาซึ่งข้อมูลนำมาใช้นั้น อาศัยหลักการของฐานข้อมูลมาช่วยในการได้มาซึ่งข้อมูลซึ่งสามารถส่งภาษา SQL เพื่อไป select ข้อมูลใน local storage ได้ทันที โดยประสิทธิภาพของฐานข้อมูลนั้นเทียบเท่ากับ SQLite



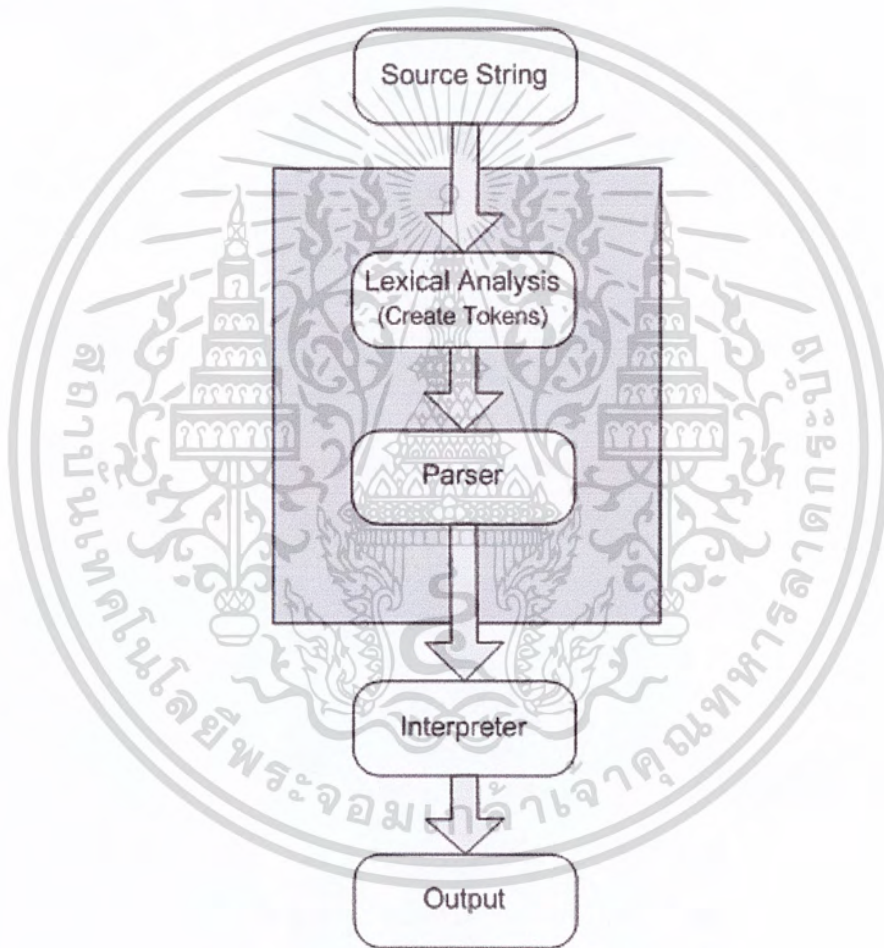
รูป 4.3 การใช้งาน Local storage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

CL อินเทอร์เน็ต

อินเทอร์เน็ต, โปรแกรมแปลคำสั่ง หรือ ตัวแปลคำสั่ง หมายถึง โปรแกรมคอมพิวเตอร์ที่ทำงานตามชุดคำสั่งที่เขียนไว้ทันที ซึ่งไม่เหมือนกับคอมไพเลอร์ (compiler) ที่แปลชุดคำสั่งจากภาษาคอมพิวเตอร์ภาษาหนึ่งไปเป็นอีกภาษาหนึ่งก่อนทำงาน โดยส่วนมากจะแปลเป็นภาษาเครื่อง โดยอินเทอร์เน็ตนั้นมีการทำงานดังรูปด้านล่าง



รูป 5.1 การทำงานของอินเทอร์เน็ต

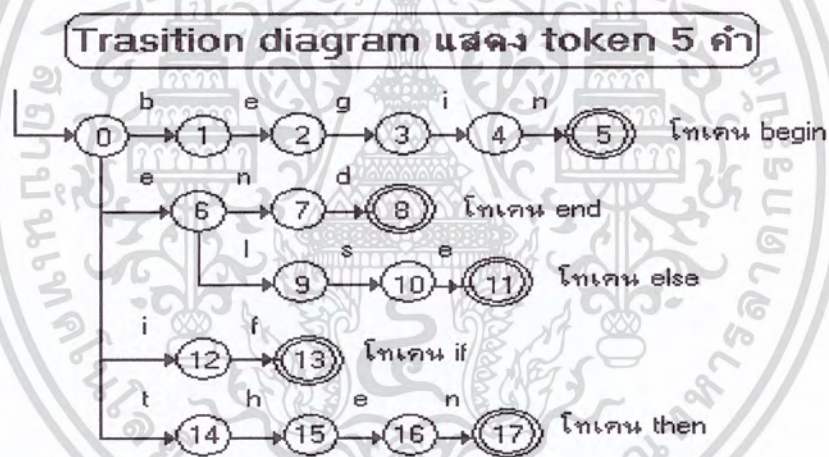
โดยในบทนี้จะกล่าวถึงการทำงานในแต่ละส่วนของอินเทอร์เน็ต ได้แก่ lexer และ parser ของอินเทอร์เน็ตเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1 Lexer

lexer หรือ lexical analysis เรียกว่า การวิเคราะห์ศัพท์ ขั้นตอนแรก ทำหน้าที่ซอร์สโค้ดแล้ว แยกอักขระออกตามลักษณะ และหน้าที่ เช่น คำสั่ง ตัวแปร ค่าคงที่ เป็นต้น รวมถึงการขจัด ส่วนหมายเหตุ(comment) ช่องว่างขาว (white space) เช่น blank, tab และ new line เป็นต้น ผลของขั้นตอนนี้จะแยกทุกสิ่งที่ควรแยกออกจากกัน เพื่อนำไปตรวจสอบในขั้นตอนต่อไป เป็นขั้นตอนแรกของอินเตอร์พรีเตอร์ หน้าที่หลักคืออ่านอักษรทีละตัวจากซอร์สโค้ดแล้วส่งผลลัพธ์ที่เป็น โทเคนต่าง ๆ และนำโทเคนต่าง ๆ แยกเก็บในตาราง โดยโทเคนที่เป็นตัวระบุจะนำไปเก็บที่ตารางสัญลักษณ์ ส่วนโทเคนอื่น ๆ จะเก็บในแฟ้มข้อมูลชั่วคราว ซึ่งโทเคนต่าง ๆ นี้ ให้กับขั้นตอนซินเทคอนาไลเซอร์ และเก็บค่าซึ่งจำเป็นต้องใช้ในขั้นตอนอื่น ๆ ไว้ในตารางสัญลักษณ์ ซึ่งขั้นตอนซินเทคอนาไลเซอร์ และเลกซิคัล ออนาไลเซอร์ ต้องมีความสัมพันธ์กันอย่างใกล้ชิด

ในส่วนของ transition diagram แสดงลำดับของแต่ละประโยค เช่นการใช้แยก begin end if then และ else



รูป 5.2 การทำงานของ Transition Diagram

lexemes เป็นหน่วยที่เล็กที่สุดของรูปแบบกฎเกณฑ์เช่น Identifier ค่าคงที่, ตัวดำเนินการและคำพิเศษต่างๆ เช่น begin if while +- */ 10

โทเคน คือ การจัดกลุ่มของ lexeme และบางครั้งโทเคนอาจเป็น lexeme ได้โทเคนที่เป็นตัวระบุ (ตัวแปร) จะถูกนำไปเก็บไว้ที่ตารางสัญลักษณ์เพื่อว่าส่วนอื่น ๆ จะนำไปใช้ ส่วนโทเคนอื่นๆ จะเก็บไว้ในแฟ้มชั่วคราว โดยโทเคนนั้นจะนำมาเก็บในตารางดังตัวอย่างด้านล่างตัวอย่าง

$$\text{Result} = \text{Oldsum} - \text{Value} / 100$$

ตาราง 5.1 แสดงการ Assign ค่าไปในตารางของ Token

ที่	Token	Lexemes
1	IDENT	RESULT
2	ASSIGN-OP	=
3	IDENT	OLDSUM
4	SUBTRACT-OP	-
5	IDENT	VALUE
6	DIVISION-OP	/
7	INT-literal	100
8	Semicolon	;

เมื่อจบกระบวนการของ lexical analysis แล้วนั้นจะทำให้ซอร์สโค้ดที่แยกอักขระแล้วแปลงเป็นโทเคนนำไปเก็บในตารางเพื่อนำไปใช้ในขั้นของการตรวจสอบซินเทกซ์ต่างๆต่อไป

5.2 Parser

ขั้นตอนในการสังเคราะห์รูปแบบกฎเกณฑ์ที่มีการตรวจสอบว่าคำสั่งถูกต้องตามหลักไวยากรณ์หรือไม่

ประเภทของ parser นั้นสามารถแบ่งได้เป็น 2 ประเภท คือ

5.2.1 Top down

เป็นการสร้าง parser tree โดยเริ่มจาก root โดยเรียงลำดับแบบ leftmost derivation และ Traces หรือ บิวท์พาซซิงทรีแบบพรีออเดอร์ (เป็นวิธีการท่องไปในทรีแบบ root left right) algorithm ที่มีชื่อเสียงของ top down parser คือ LL parser ใช้วิธีการอ่านแบบ Leftmost Derivation

5.2.2 Bottom up

เป็นการสร้างพาซซิงทรีที่เริ่มต้นจาก leaves Node Algorithm ที่มีชื่อเสียงของ bottom up parser คือ LR parser ใช้วิธีการอ่านแบบ rightmost derivation

5.3 Walker

เมื่อจบขั้นตอนของ parser แล้วจะได้ AST tree ออกมา walker จะทำการท่องเข้าไปใน AST tree เมื่อเจอ โหนดต่างๆ จะทำการพิจารณาว่าควรกระทำอย่างไรตามไวยากรณ์ของภาษาซึ่งเป็นขั้นตอนสุดท้ายของอินเทอร์พรีเตอร์

5.4 Interpreter

เมื่อจบขั้นตอนของ parser แล้วจะได้ parser tree ออกมา เพื่อนำไปทำการอินเทอร์พรีตคำสั่งของ parser tree เพื่อให้รู้ว่าคำสั่งเหล่านี้สมควรที่จะเอาต์พุตประเภทใดออกมา เป็นขั้นตอนสุดท้ายของอินเทอร์พรีเตอร์

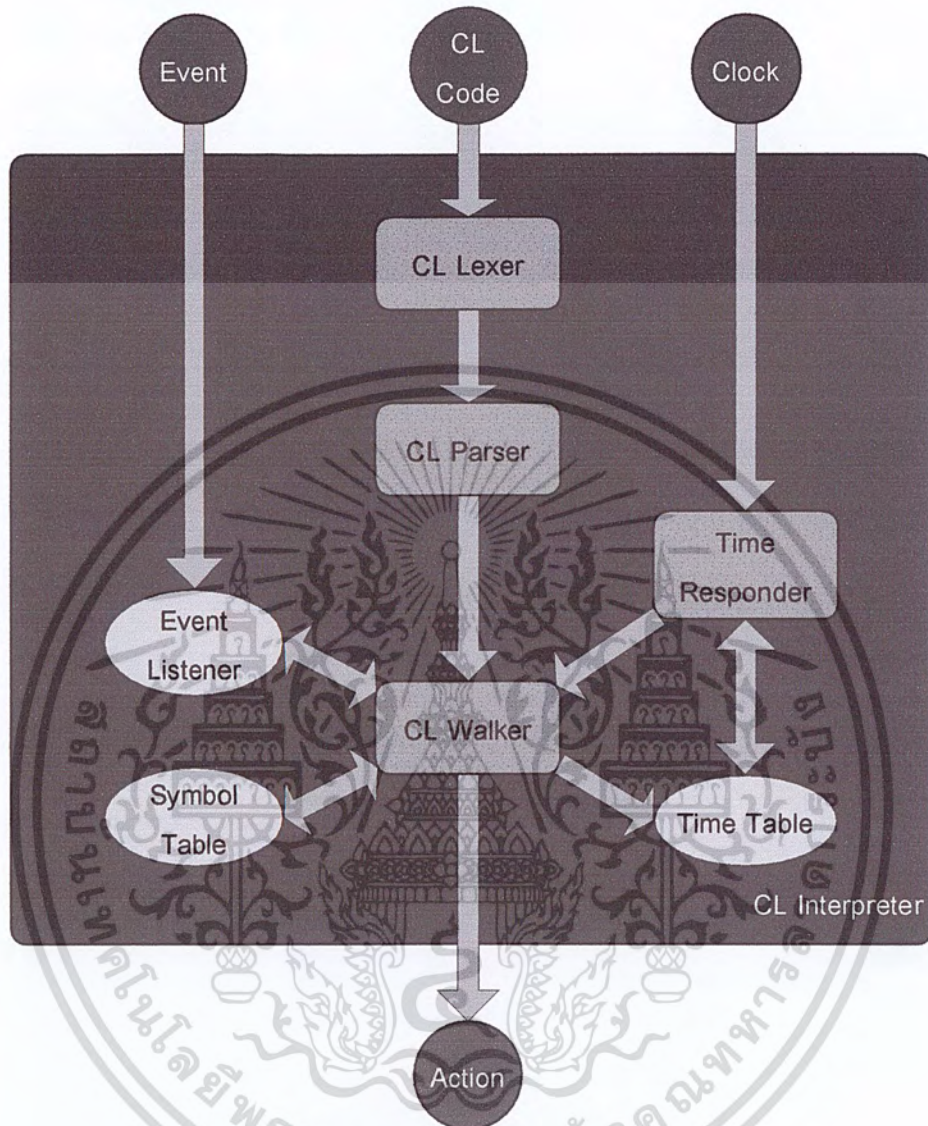
```
function sk$cmp(lhs, rhs, op)
{
  if (typeof lhs === "number" && typeof rhs === "number")
  {
    switch (op)
    {
      case '<': return lhs < rhs;
      case '<=': return lhs <= rhs;
      case '>': return lhs > rhs;
      case '>=': return lhs >= rhs;
      case '!=': return lhs !== rhs;
      case '===': return lhs === rhs;
      default: throw "assert";
    }
  }
  else
  {
    var ret;
    if (lhs.richcmp$ !== undefined)
      return lhs.richcmp$(rhs, op);
    else if (lhs.__cmp__ !== undefined)
      ret = lhs.__cmp__(rhs);
    else if (rhs.__cmp__ !== undefined)
      ret = -rhs.__cmp__(lhs);
    else
    {
      // todo; dispatch to the specific __eq__, etc.
      throw new AttributeError("no attribute __cmp__");
    }

    switch (op)
    {
      case '<': return ret < 0;
      case '<=': return ret <= 0;
      case '>': return ret > 0;
      case '>=': return ret >= 0;
      case '===': return ret === 0;
      case '!=': return ret !== 0;
      default: throw "assert";
    }
  }
}
}
```

รูป 5.3 การอินเทอร์พรีตคำสั่งประเภทของ Comparison Operator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 CL อินเทอร์เน็ตพรีเตอร์



รูป 5.4 การทำงานของ CL อินเทอร์เน็ตพรีเตอร์

CL Interpreter ประกอบด้วยองค์ประกอบต่างๆ ดังนี้

- 1) CL lexer : ทำการตรวจจับโทเคนภาษา CL จาก CL Code ผลลัพธ์ที่ได้คือกลุ่มของโทเคน
- 2) CL parser : ทำการตรวจจับว่าลำดับของโทเคนดังกล่าวเป็นไปตามแกรมม่าข้อใดแล้วทำการสร้าง AST
- 3) CL walker : ทำการตรวจจับว่า AST ที่มีลักษณะดังกล่าวจะให้ทำงานในลักษณะใด โดยจะทำการปรับปรุง time table, อีเวนท์ listener, symbol table และ action อย่างเหมาะสมตาม AST ที่ได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) Time responder : ทำการอ่านเวลาปัจจุบันจาก Clock เพื่อตรวจสอบกับแต่ละจุดเวลาของคำสั่งที่เก็บอยู่ในtime table
- 5) Time table : เป็นตารางเวลาที่ใช้เก็บจุดเริ่มต้นและสิ้นสุดของแต่ละคำสั่งตามที่ โปรแกรม CL กำหนด
- 6) อีเวนท์ listener : ทำการตรวจรับอีเวนท์ภายนอกตามที่ โปรแกรม CL กำหนด
- 7) Symbol table : เป็นตารางที่ใช้เก็บชื่อ symbol, ชนิดของ symbol และค่าของ symbol ของแต่ละสโคปโดยสโคปอาจแตกต่างกันในแต่ละช่วงของโปรแกรม

5.6 การออกแบบ และพัฒนา CL อินเทอร์พรีเตอร์

เนื่องจากภาษา CL เป็นภาษาที่พัฒนาจากภาษา Python โดยเพิ่มส่วนที่เกี่ยวกับเวลาและเหตุการณ์เข้าไปผู้พัฒนาจึงพัฒนาภาษา CL โดยนำไวยากรณ์ภาษาไพธอนมาเพิ่มเติมในส่วนที่เกี่ยวข้องกับเวลาและเหตุการณ์เข้าไปจนเป็นไวยากรณ์ของภาษา CL และใช้ ANTLR ซึ่งเป็น parser generator ในการพัฒนา CL lexer ทำการตรวจจับโทเคนภาษา CL จาก CL code ผลลัพธ์ที่ได้คือกลุ่มของโทเคน

```

LPAREN : '(' {this.implicitLineJoiningLevel++;} ;
RPAREN : ')' {this.implicitLineJoiningLevel--;} ;
LBRACK : '[' {this.implicitLineJoiningLevel++;} ;
RBRACK : ']' {this.implicitLineJoiningLevel--;} ;
UNDERSCORE : '_' ;
ARROW : '->' ;
COLON : ':' ;
COMMA : ',' ;
SEMI : ';' ;
PLUS : '+' ;
MINUS : '-' ;

```

รูป 5.5 การประกาศ Token

CL parser ทำการตรวจจับว่าลำดับของโทเคนดังกล่าวเป็นไปตามแกรมมาข้อใดแล้วทำการสร้าง AST โดย CL parser เป็น parser ประเภท top down parser

```

cl_keyword : 'parallel'|'sequential'|'unordered'
;
cl_stmt : cl_time cl_keyword COLON event=suite (ARROW action=suite)? -> ^(cl_keyword cl_time event action?)
| cl_keyword COLON event=suite (ARROW action=suite)? -> ^(cl_keyword event action?)
| cl_time COLON event=suite (ARROW action=suite)? -> ^(cl_time event action?)
;
cl_time : LBRACK tp1=timepoint COMMA tp2=timepoint RBRACK
;
timepoint returns [var value]
: c1=component COLON c2=component COLON c3=component VBAR c4=component SLASH c5=component SLASH c6=component
| UNDERSCORE {$value=null;}
;
component : NAME
| NUMBER
;

```

รูป 5.6 BNF ของภาษา CL

จากภาพเป็นการแสดง grammar rule ในส่วนที่เกี่ยวข้องกับเหตุการณ์และเวลาที่เพิ่มเข้าไปจากไวยากรณ์ของภาษาไพธอน จากภาพจะเห็นเครื่องหมาย '->' ซึ่งหลังจากเครื่องหมายดังกล่าวคือโครงสร้างของ AST

CL walker ทำการตรวจจับว่า AST ที่มีลักษณะดังกล่าวจะให้ทำงานในลักษณะใด โดยจะทำการส่งผลลัพธ์ที่ไปยัง time table manager, symbol table หรือกระทำ Action แก่ผู้ใช้ที่เหมาะสมตาม AST ที่ได้รับ

```

cl_keyword : 'parallel'|'sequential'|'unordered'
;
cl_stmt : ^(cl_keyword cl_time event=suite action=suite?) (managetimetable(cl_keyword,cl_time,event,action);)
| ^(cl_keyword event=suite action=suite?) (managetimetable(cl_keyword,null,event,action);)
| ^(cl_time event=suite action=suite?) (managetimetable("ordered",cl_time,event,action);)
;

```

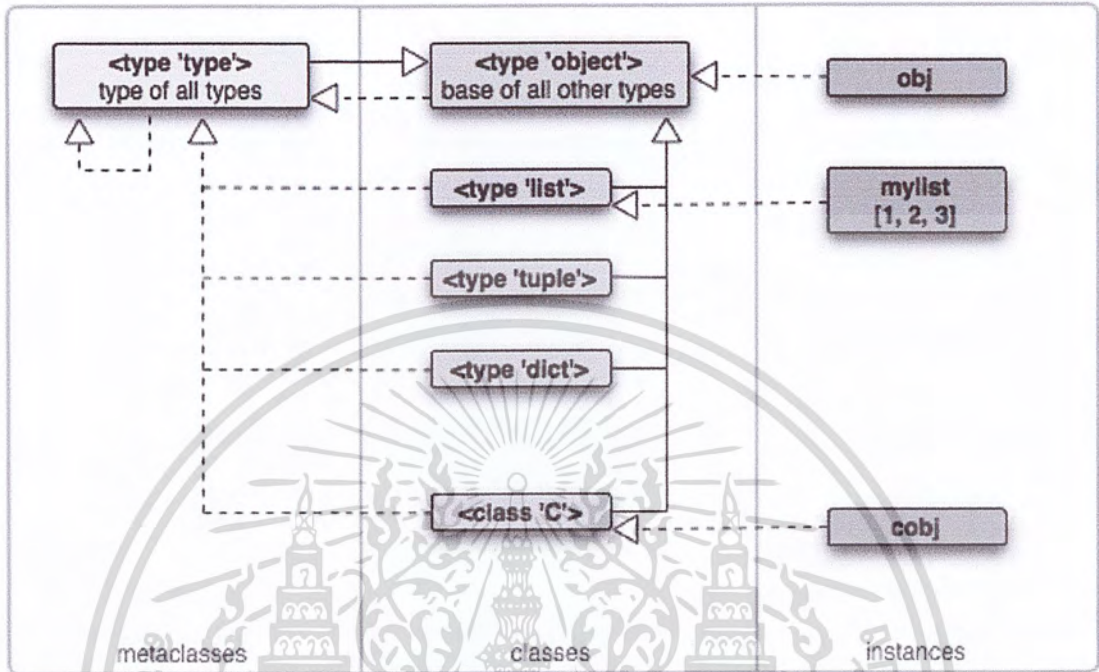
รูป 5.7 BNF ของภาษา CL(2)

จากภาพเป็นส่วนของ CL Walker ที่เมื่อเจอ โครงสร้างของ AST ตามที่กำหนดในที่นี้คือ AST ของ CL statement จะทำการเรียกฟังก์ชัน managetimetable() ซึ่งก็คือส่วน time table manager ที่ใช้ในการเพิ่มสมาชิกใน time table

5.7 Time table manager

Symbol table คือ ตารางที่ใช้เก็บชื่อ symbol, ชนิดของ symbol และค่าของ symbol ของแต่ละสโคปโดยสโคปจะแตกต่างกันในแต่ละช่วงของโปรแกรม และส่วนดังกล่าวนี้เองที่ทำให้ภาษา CL

สามารถทำออบเจ็กต์โอเรียนเต็ลได้อย่างสมบูรณ์โดยหลักการดังกล่าวเป็นหลักการที่ใช้ในภาษาไพธอน โดยมีหลักการทำงานในการจัดการกับตัวแปรต่างๆดังนี้



รูป 5.8 หลักการสร้าง symbol table

หลักการคือทุกตัวแปรที่เกิดขึ้นในโปรแกรมจะเป็นออบเจ็กต์ทั้งหมด โดยเริ่มต้นจะมีออบเจ็กต์อยู่สองตัวคือออบเจ็กต์ที่ชื่อไทป์ และออบเจ็กต์ที่ชื่อออบเจ็กต์ โดยไทป์เป็นชั้นคราสของออบเจ็กต์ (ถูกสรเส้นทึบที่จะชี้ไปที่ซูปเปอร์คราส) และออบเจ็กต์เป็นอินสแตนซ์ของไทป์ (ถูกสรเส้นประ) โดยบิวท์อินไทป์ของภาษา เช่น list, tuple เป็นต้น จะเป็นอินสแตนซ์ของไทป์และเป็นชั้นคราสของออบเจ็กต์

ถ้าสำหรับคราสที่ผู้ใช้สร้างขึ้นจะถูกมองเป็นบิวท์อินไทป์ตัวหนึ่งเพื่อให้สามารถเรียกใช้แอตทริบิวต์และเมทอดได้เหมือนกับทำในบิวท์อินไทป์ส่วนตัวแปรต่างๆที่เกิดขึ้นทั้งหมดในโปรแกรมจะเป็นอินสแตนซ์ของบิวท์อินไทป์ หรือ class ที่สร้างขึ้นอย่างเหมาะสม

บทที่ 6

Integrated development environment

Integrated development environment (IDE) คือ องค์กรประกอบ หรือตัวช่วยต่างๆ ที่คอยช่วยเหลือดีเวลลอปเปอร์ในการพัฒนาแอปพลิเคชันต่างๆ เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ และตรวจสอบระบบที่จัดทำได้ทั้งเชิงลึกหรือตื้น ทำให้การพัฒนางานเป็นไปอย่างรวดเร็วมากขึ้น โดยมีส่วนประกอบดังต่อไปนี้

6.1 ส่วนประกอบของ Integrated development environment

6.1.1 source code editor

source code editor คือ ส่วนของแท็กซอติเตอร์ซึ่งจะมีลักษณะคล้ายกับโน้ตแพทช์ใช้ในการบันทึกข้อมูลของซอร์สโค้ดเพียงแต่จะมีความสามารถจะเหนือกว่า เช่น

- 1) highlight คือ สีของคำสั่ง ตัวแปร ฟังก์ชันหรือค่าต่างๆ ในโค้ดได้
- 2) auto suggest คือ เมื่อพิมพ์อักษรออกมาถ้าตรงกับฟังก์ชันที่มีใน IDE จะมีการลิสต์ชื่อของฟังก์ชันทั้งหมดที่ขึ้นต้นด้วยตัวอักษรที่ผู้ใช้พิมพ์ขึ้นมาให้เลือก
- 3) syntax warning คือ มีการตรวจสอบ syntax เมื่อผู้ใช้พิมพ์ชื่อ ฟังก์ชันผิด โดยที่ผู้ใช้ไม่ทันสังเกต หรือ เมื่อผู้ใช้พิมพ์คำสั่งด้วยเครื่องหมาย ;
- 4) design form คือ การจัดแบบฟอร์มในการเขียนโปรแกรม ซึ่งโดยปรกติแล้วเมื่อผู้ใช้ขึ้นบรรทัดใหม่เคอร์เซอร์จะมาชิดซ้ายสุด แต่ว่าแท็กซอติเตอร์ใน IDE จะมีการจัดย่อหน้าให้

6.1.2 คอมไพเลอร์ หรืออินเตอร์พรีเตอร์

คอมไพเลอร์ คือ เครื่องมือที่ช่วยในการแปลงจากภาษาของมนุษย์ไปเป็นภาษาแอสเซมบลีเพื่อให้คอมพิวเตอร์เข้าใจคำสั่งของมนุษย์ และทำงานต่อไป ซึ่งโดยปกติแล้วคอมพิวเตอร์จะทำงานด้วยสัญญาณไฟฟ้า คือจะติดต่อกันด้วยสัญญาณไฟ 0 หรือ 1 โดย 0 ก็คือไม่มีไฟ 1 คือมีไฟ โดยจะสลับกับ เช่น 0001 1000 1101 เป็นต้น เพราะฉะนั้นเมื่อเวลาที่มนุษย์เขียนโปรแกรม มนุษย์จะไม่สามารถเขียนให้เป็น 00010111 ได้จึงทำให้มีการแปลงตัวเลขเหล่านี้มาเป็นภาษาของมนุษย์มากขึ้น นั่นคือภาษาแอสเซมบลีซึ่งเป็นภาษาที่มนุษย์เริ่มทำความเข้าใจได้ และ พื้นฐานที่สุดที่จะใช้เพื่อการแปลงลงไปเป็นคำสั่งของคอมพิวเตอร์ให้ทำงานตรงตามความต้องการ

6.1.3 build automation tools

Build automation tools คือ เครื่องมือที่จำเป็น และถูกเรียกใช้อยู่เป็นประจำของซอฟต์แวร์ดีเวลลอปเปอร์เพื่อเพิ่มความสะดวกในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.4 debugger

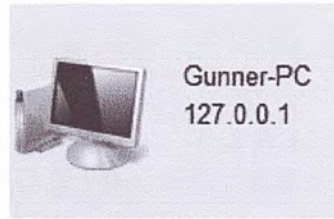
Debugger คือ เครื่องมือที่ใช้ในการทดลอง และตรวจสอบ โปรแกรมที่สร้างขึ้นมา โดยทำให้โปรแกรมทำงานได้ไปที่ละบรรทัด โดยที่มีการแสดงผลลัพธ์ต่างๆออกมาในขณะที่ทำงานอยู่ที่บรรทัดนั้น

6.2 รูปแบบในการออกแบบ IDE

ในการออกแบบ IDE นั้นต้องสร้างสะดวกในการทำงานของผู้ใช้ โดยที่เน้นไปทางด้านการใช้งาน และเข้าถึงตัวของผู้ใช้งานได้ง่าย แบ่งออกหัวข้อได้ดังต่อไปนี้

- 1) เมื่อนำ IDE มาใส่ไว้บนเว็บเบราว์เซอร์แล้วทำให้ผู้ใช้นั้นสามารถเข้าถึงได้ง่ายเพียงแค่เชื่อมต่อกับอินเทอร์เน็ตเนื่องจากต้องการให้ IDE สามารถควบคุมการทำงานต่างๆที่เราได้ทำการโค้ดตั้งเอาไว้ โดยที่มีการใช้ HTML5 มาใช้ในการพัฒนาซึ่งทำให้เว็บเบราว์เซอร์มีลูกเล่นในการทำงานมากขึ้น เช่นการใช้ local storage มาใช้งานในการเก็บข้อมูลที่โค้ดตั้งไว้ภายในเครื่อง และเนื่องจากเป็นเว็บเบราว์เซอร์ทำให้สามารถใช้งานได้ทุกแพลตฟอร์มไม่ว่าจะเป็น PC หรือ สมาร์ทโฟน
- 2) เมื่อนำ IDE มาใส่ไว้ใน calendar ทำให้สามารถใช้เป็นเพื่อนออนไลน์ในเซออร์ซึ่งผู้ใช้งานสามารถทำความเข้าใจในการใช้งานได้ง่าย ผนวกกับที่ภาษา CL นั้นมีความเกี่ยวข้องกับเวลาทำให้ตรงกับคอนเซ็ปของการจัดการกิจกรรมต่างๆผ่านภาษา CL

- 3) คอมพิวเตอร์ที่ใช้ในโปรไฟล์เดียวกัน เมื่อมีการเข้าถึงหลายเครื่องจะมีการอัปเดตไปที่เครื่องที่มีการ Active อยู่ก่อนหน้า

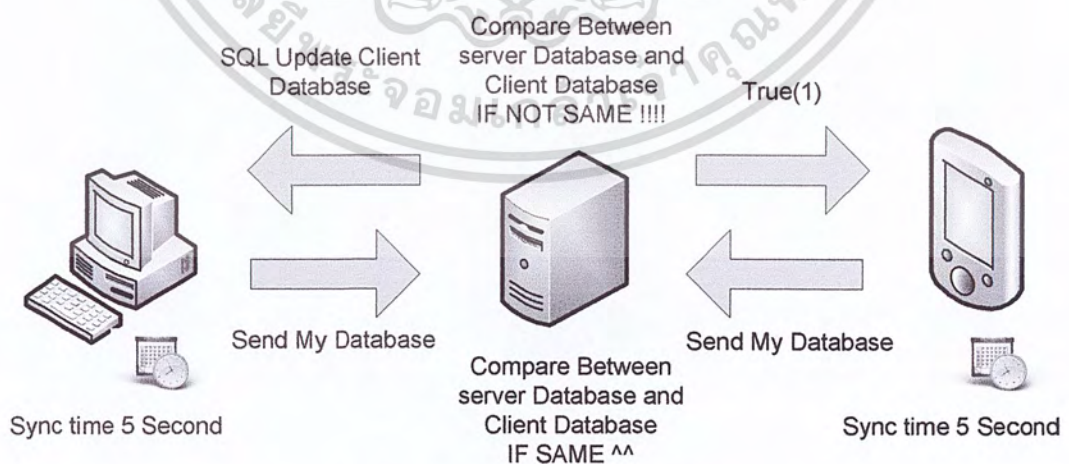


รูป 7.3 คอมพิวเตอร์ที่ใช้ในโปรไฟล์เดียวกัน

7.2 หลักการทำงานของ การ Synchronization

การทำงานของ การ synchronization ใน โครงงานนี้จะสามารถแบ่งออกได้เป็น 2 ลักษณะ

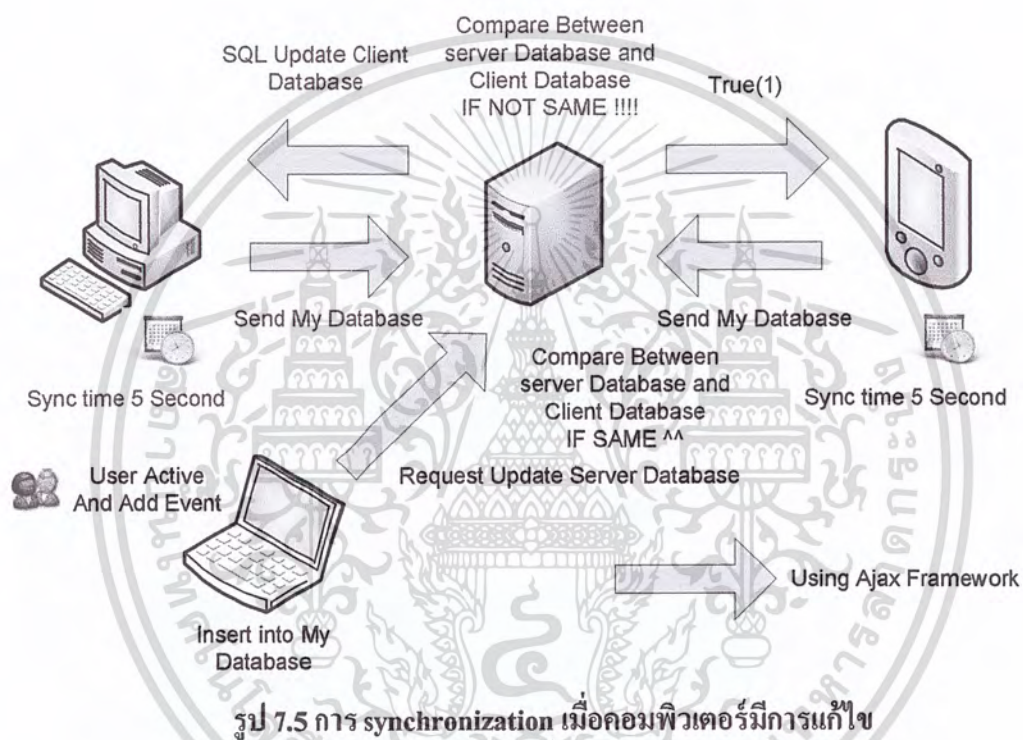
- 1) ใช้หลักการของการ synchronization เป็นคาบเวลา นั่นคือ เมื่อถึงเวลาที่กำหนด คอมพิวเตอร์ จะทำการส่งรีควีสไปที่เซิร์ฟเวอร์แล้วทำการเทียบฐานข้อมูลระหว่างคอมพิวเตอร์กับเซิร์ฟเวอร์ว่าตรงกันหรือไม่ หากฐานข้อมูลตรงกันแล้วก็ไม่ต้องการ เรสพอนส์กลับมาจากเซิร์ฟเวอร์แต่หากฐานข้อมูลของเซิร์ฟเวอร์กับ คอมพิวเตอร์นั้นไม่ตรงกันนั้นเซิร์ฟเวอร์ จะทำการสร้างคำสั่ง SQL ชุดหนึ่งขึ้นมาเพื่อให้คอมพิวเตอร์นั้นสามารถนำ SQL นั้นๆ ไป อัปเดตฐานข้อมูลของเครื่องคอมพิวเตอร์นั้นๆ เพื่อให้ข้อมูลนั้นตรงกับเซิร์ฟเวอร์การทำงานเป็นไปตามรูป 7.4



รูป 7.4 การ synchronization เป็นคาบเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) ใช้หลักการของการ synchronization เมื่อคอมพิวเตอร์มีการแก้ไขหรือปรับเปลี่ยนคอมพิวเตอร์เครื่องที่ถูกแก้ไขจะทำการส่งอัปเดตไปเซิร์ฟเวอร์เพื่อให้คอมพิวเตอร์ที่ Active อยู่ในเวลาเดียวกันนั้นสามารถรู้ได้ว่าตอนนี้มีการแก้ไขที่คอมพิวเตอร์เครื่องอื่นที่ใช้โปรไฟล์เดียวกันจากนั้นเครื่องคอมพิวเตอร์เครื่องอื่นๆก็จะทำการอัปเดตข้อมูลตัวเองให้เหมือนกับเซิร์ฟเวอร์เมื่อถึงคาบเวลาที่กำหนดโดยใช้หลักการตามข้อ 1 การทำงานเป็นไปตามรูป 7.5



7.3 Tool ในใช้ในการพัฒนา Synchronization

- 1) Local Storage ใช้เป็นฐานข้อมูลในฝั่งของเครื่องคอมพิวเตอร์ ซึ่งเป็นการพัฒนาขึ้นมาใหม่ใน HTML5 โดยในการทำงานภายในมีการพัฒนาได้เทียบเท่ากับ SQLite
- 2) Ajax Framework ปัจจุบันนี้ ลักษณะการทำงาน โดยการติดต่อระหว่างเครื่องคอมพิวเตอร์กับ เซิร์ฟเวอร์นั้นเริ่มถูกนำมาใช้งานอย่างแพร่หลายในลักษณะการติดต่อสื่อสารผ่านทางเว็บเบราว์เซอร์ ซึ่งการทำงานแบบนี้ จะมีการทำงาน โดยคอมพิวเตอร์ จะร้องขอและต้องการข้อมูลบางอย่างจากเซิร์ฟเวอร์ดังนั้นการโหลดและการรีเฟรชหน้าจอก็เป็นสิ่งที่หลีกเลี่ยงไม่ได้จึงเป็นผลให้การทำงานของฝั่งคอมพิวเตอร์นี้ทำให้ผู้ใช้ต้องหยุดการโหลดและการ

รีเฟรชหน้าจอก็ ซึ่งถือว่าการทำงานที่ไม่มีประสิทธิภาพ ดังนั้น จึงมีเฟรมเวิร์กนี้ขึ้นมา
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อทำการแก้ไขปัญหาดังกล่าวโดยหลักการทำงาน Ajax engine ทำหน้าที่เป็นตัวกลางระหว่างคอมพิวเตอร์ กับเซิร์ฟเวอร์ฉะนั้นเมื่อคอมพิวเตอร์ มีการรีควสโดยปกติจะส่ง HTTP request ไปยังเซิร์ฟเวอร์โดยตรงคอมพิวเตอร์ จะส่ง JavaScript call ไปยัง Ajax engine เพื่อโหลดข้อมูลที่ผู้ใช้ต้องการ และหาก Ajax engine ต้องการข้อมูลเพิ่มเติมในการตอบสนองต่อผู้ใช้ Ajax engine จะส่งรีควสไปยังเซิร์ฟเวอร์โดยใช้ XML รูปแบบ Ajax Framework เป็นไปตามรูป 7.6

```

var HttpRequest = false;
if (window.XMLHttpRequest) { // Mozilla, Safari,...
  HttpRequest = new XMLHttpRequest();
  if (HttpRequest.overrideMimeType) {
    HttpRequest.overrideMimeType("text/html");
  }
}
else if (window.ActiveXObject) { // IE
  try {
    HttpRequest = new ActiveXObject("Msxml2.XMLHTTP");
  } catch (e) {
    try {
      HttpRequest = new ActiveXObject("Microsoft.XMLHTTP");
    } catch (e) {}
  }
}
if (!HttpRequest) {
  alert('Cannot create XMLHttpRequest instance');
  return false;
}
var url = 'synchronizeserver.php';
var pmeters = 'firsttable='+j1+
              '&secondtable='+j2+
              '&thirdtable='+j3+
              '&user2='+reusername();
HttpRequest.open('POST',url,true);
HttpRequest.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
HttpRequest.send(pmeters);

```

รูป 7.6 Ajax Framework Code Skeleton

- 3) PHP เป็นภาษาในการเขียนเว็บแอปพลิเคชันในส่วนของเซิร์ฟเวอร์
- 4) MySQL ฐานข้อมูลของเซิร์ฟเวอร์


```

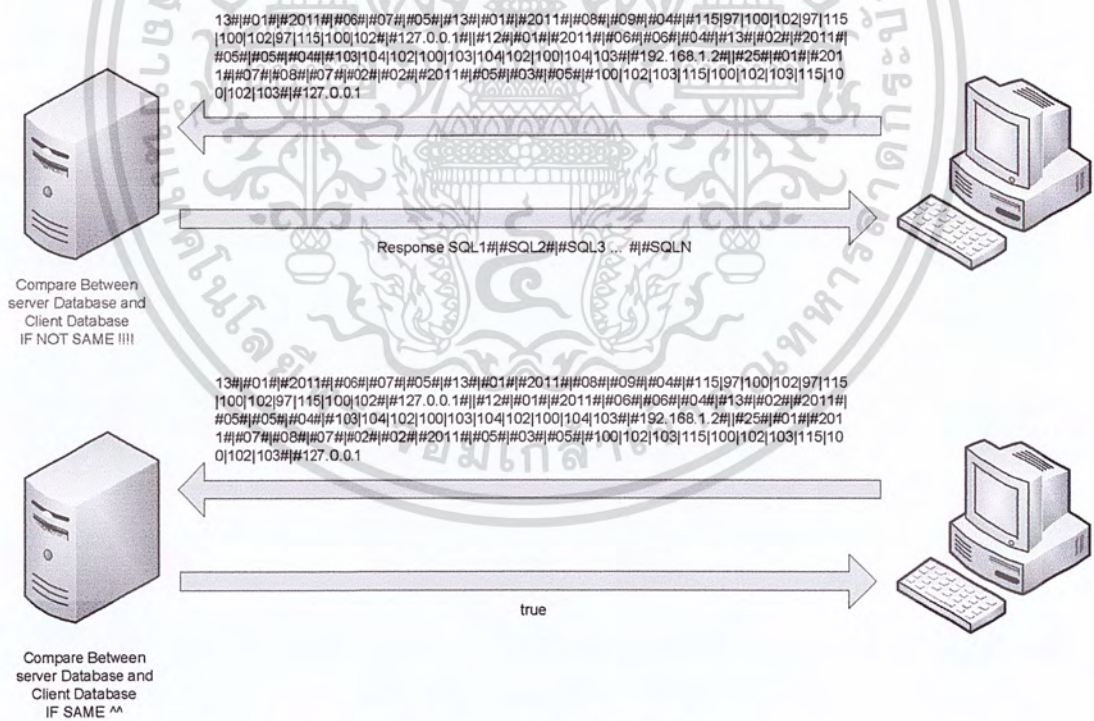
if(addquery.search("#|#")!=-1)
{
    lastquery=addquery.split("#|#");
    for(var i=0; i<=lastquery.length-1; i++){
        if(checkalreadydb(lastquery[i])){dbobj.webdb.ExecuteInsertSQL(lastquery[i]);}
    }
    dbobj.webdb.ExecuteSelectAddClient("SELECT * FROM COMPUTER");
    choosepageshow();
}
else {
    if(checkalreadydb(addquery)){
        dbobj.webdb.ExecuteInsertSQL(addquery);
        dbobj.webdb.ExecuteSelectAddClient("SELECT * FROM COMPUTER");
        choosepageshow();
    }
}
}

```

รูป 7.9 โปรแกรมที่ทำงานเมื่อฐานข้อมูลไม่เหมือนกัน

7.5 โปรแกรมของฝั่งเซิร์ฟเวอร์

เซิร์ฟเวอร์ทำการตรวจสอบว่าฐานข้อมูลของเซิร์ฟเวอร์กับคอมพิวเตอร์นั้นตรงกันหรือไม่ หากตรงกันแล้วเซิร์ฟเวอร์จะส่ง 1 กลับไปแต่หากไม่ตรงกันจะส่ง SQL เพื่อให้คอมพิวเตอร์ทำการเปลี่ยนแปลงฐานข้อมูลของตัวเอง การทำงานเป็นดังรูป 7.10



รูป 7.10 การทำงานฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

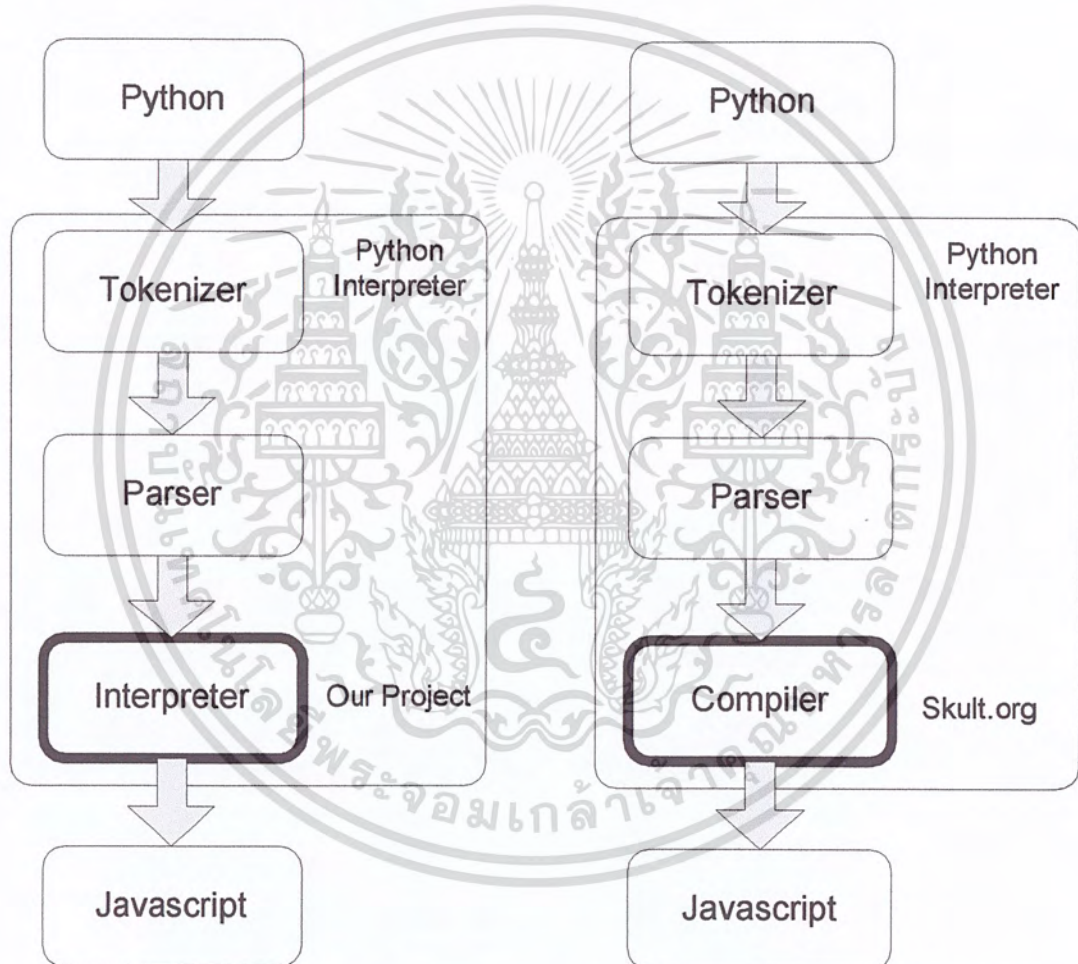
บทที่ 8

ผลการทดลอง

8.1 ผลการทดลองส่วนของไพธอนอินเตอร์พรีเตอร์

การทดลองของโครงการนี้แบ่งเป็น 2 ส่วน คือ ส่วนของ CL อินเตอร์พรีเตอร์ และ ส่วนของ Integrated Development Environment (IDE)

8.1.1 การทำงานของไพธอนอินเตอร์พรีเตอร์



รูป 8.1 รูปเปรียบเทียบการทำงานของ skulpt.org กับการทำงานของปริณญาณิพนธ์

หากเปรียบเทียบการทำงานของ skulpt.org แล้ว เนื่องจากการทำงานเป็นแบบทรานสเลเตอร์ โดยที่เมื่อเราผ่านขั้นตอนของ parser แล้วได้ผลลัพธ์เป็น AST (Abstract tree) แล้วในกระบวนการ คอมไพเลอร์ นั้นจะทำการนำ AST ที่ได้มาทำการประมวลผล โดยทำการประมวลผลที่รีทั่มหมดทุกโหนดแล้วทำการเจนเนอเรทออกมาเป็นโค้ดจาวาสคริปต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่วิธีที่ดีกว่าคือการที่เราได้ผลลัพธ์จาก parser เป็น AST แล้วก็นำมาทำการแทรกเวิร์สทีละ โหนด โดยในแต่ละ โหนดนั้นเมื่ออ่าน โหนดนั้นมาแล้วทำการประมวลผลเลยโดยไม่ต้องรอการประมวลผลจากทรีทั้งต้นซึ่งกระบวนการนี้เป็นกระบวนการของอินเตอร์พรีเตอร์ซึ่งแตกต่างจาก คอมไพเลอร์ ซึ่งทำการให้การทำงานนั้นมีประสิทธิภาพมากขึ้น จึงควรจะทำการแก้ไขที่ส่วนของ คอมไพเลอร์ นั้นให้เป็นการทำงานของ อินเตอร์พรีเตอร์

8.1.2 ความสามารถในการทำงานของ skulpt.org

โดยความสามารถในการทำงานนั้นเป็นไปตามตารางด้านล่างนี้

ตาราง 8.1 ความสามารถในการอินเตอร์พรีตของ skulpt.org

ฟังก์ชันการทำงานต่างๆ	ทำงานได้	ทำงานไม่ได้
1. Variables and Arithmetic Expressions	✓	
2. File Input and Output		✓
3. Strings (Data types)	✓	
4. Lists (Data types)	✓	
5. Tuples (Data types)	✓	
6. Dictionary (Data types)	✓	
7. Build In function Append in list data types	✓	
8. Complex Number	✓	
9. Get type from variable	✓	
10. Reference and Copy	✓	
11. หาค่าสูงสุดต่ำสุดใน Lists		✓
12. Build In function Extend in list data types	✓	
13. Build In function insert in list data types	✓	
14. Build In function count in list data types		✓
15. Build In function Reverse in list data types		✓
16. Build In function Remove in list data types		✓
17. Build In Function ของ String ส่วนใหญ่		✓
18. ในไพธอน3.0 มี Data types ชนิด sets		✓

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 8.1 ความสามารถในการอินเตอร์พรีตของ skulpt.org (ต่อ)

19. Condition Sentences	✓	
20. การวนซ้ำด้วยคำสั่ง While	✓	
21. การวนซ้ำด้วยคำสั่ง For	✓	
22. Exception		✓
23. คลาส และออบเจ็ก โอเรียนเต็ล		✓
24. การ import library จากภายนอก		✓
25. Operators and Expression	✓	
26. Arithmetic Operators	✓	
27. Comparison Operators	✓	
28. Logical Operators	✓	
29. Conditional Operators	✓	
30. Assignment Operators	✓	
31. Function	✓	
32. Recursion	✓	
33. Inheritance and Polymorphism		✓

จากตารางจะพบว่าเว็บ skulpt.org ยังไม่สามารถทำงานได้อย่างสมบูรณ์ เราได้ทำการแก้ไขปัญหาดังกล่าว โดยปัญหาการทำงานที่เป็นปัญหาสำคัญคือปัญหาออบเจ็ก โอเรียนเต็ลในภาษาจาวาสคริปต์ที่รูปแบบการใช้งานออบเจ็ก โอเรียนเต็ลแตกต่างไปจากภาษาอื่นที่คุ้นเคย ซึ่งเราแก้ไขปัญหาดังกล่าวโดยใช้ MOOTOOLS เฟรมเวิร์ค ในการแก้ปัญหา

8.1.3 การแก้ไขปัญหาลออบเจ็กโอเรียนเต็ลของภาษาจาวาสคริปต์

เนื่องจากภาษาจาวาสคริปต์นั้นไม่มีการทำงานซึ่งเป็นออบเจ็ก โอเรียนเต็ลที่แท้จริง เพราะว่า ภาษาจาวาสคริปต์นั้นหากต้องการเขียน โปรแกรมแบบออบเจ็ก โอเรียนเต็ลต้องทำการสร้าง เรฟเฟอร์เรนซ์ใหม่ขึ้นมาเพื่อให้จาวาสคริปต์มองว่า ฟังก์ชันการทำงานนี้อยู่ในเรฟเฟอร์เรนซ์ใหม่ใดซึ่งการดีไฟน์นั้นจะต้องดีไฟน์โดยการใช้ชันทัดคั้งนี้

โปรแกรม 8.1 การทำงานของ object oriented ในภาษาจาวาสคริปต์

```
function CustomerBooking (bookingId, customerName, film, showDate)
{
  this.customerName = customerName;
  this.bookingId = bookingId;
  this.showDate = showDate;
  this.film = film;
}

CustomerBooking.prototype.setCustomerName = function(customerName)
{
  this.customerName = customerName;
}
```

การทำงานดังนี้ไม่เป็นออบเจ็กต์โอเรียนเตดที่แท้จริงเนื่องจากเราไม่สามารถ inheritance หรือ polymorphism ได้เลยซึ่ง จึงทำการแก้ไขโดยใช้เฟรมเวิร์ค ในจาวาสคริปต์ซึ่งทำการอิมพลีเม้นท์มาเพื่อทำการจัดการกับจาวาสคริปต์ให้ทำการเป็นออบเจ็กต์โอเรียนเตดอย่างแท้จริงที่สามารถทำการ inheritance และ polymorphism โดยนำหลักการของ MOOTOOLS เฟรมเวิร์ค ส่วนที่เกี่ยวข้องกับออบเจ็กต์โอเรียนเตดนำมาใช้ในการแก้ปัญหา

8.1.4 ความสามารถในการทำงานของอินเตอร์พรีเตอร์ของโครงการ

โดยความสามารถในการทำงานนั้นเป็นไปตามตารางด้านล่างนี้

ตาราง 8.2 ความสามารถในการทำงานของอินเตอร์พรีเตอร์ของโครงการ

ฟังก์ชันการทำงานต่างๆ	ทำงานได้	ทำงานไม่ได้
1. Variables and Arithmetic Expressions	✓	
2. File Input and Output		✓
3. Strings (Data types)	✓	
4. Lists (Data types)	✓	
5. Tuples (Data types)	✓	
6. Dictionary (Data types)	✓	
7. Build In function Append in list data types	✓	
8. Complex Number	✓	
9. Get type from variable	✓	
10. Reference and Copy	✓	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 8.2 ความสามารถในการทำงานของอินเทอร์พรีเตอร์ของโครงการงาน (ต่อ)

11. หาค่าสูงสุดต่ำสุดใน Lists	✓	
12. Build In function Extend in list data types		✓
13. Build In function insert in list data types	✓	
14. Build In function count in list data types	✓	
15. Build In function Reverse in list data types	✓	
16. Build In function Remove in list data types	✓	
17. Build In Function ของ String ส่วนใหญ่	✓	
18. ในไพทอน3.0 มี Data types ชนิด sets		✓
19. Condition Sentences	✓	
20. การวนซ้ำด้วยคำสั่ง While	✓	
21. การวนซ้ำด้วยคำสั่ง For	✓	
22. Exception		✓
23. คลาส และออบเจ็กต์ โอเรียนเต็ล	✓	
24. การ import library จากภายนอก		✓
25. Operators and Expression	✓	
26. Arithmetic Operators	✓	
27. Comparison Operators	✓	
28. Logical Operators	✓	
29. Conditional Operators	✓	
30. Assignment Operators	✓	
31. Function	✓	
32. Recursion	✓	
33. Inheritance and Polymorphism	✓	

จากตารางจะพบว่าเราสามารถแก้ปัญหาต่างๆที่เกิดขึ้นได้โดยใช้หลักการการทำงาน
ดังที่ได้กล่าวไปแล้วในบทที่ 5 ในเรื่องของการสร้างอินเทอร์พรีเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

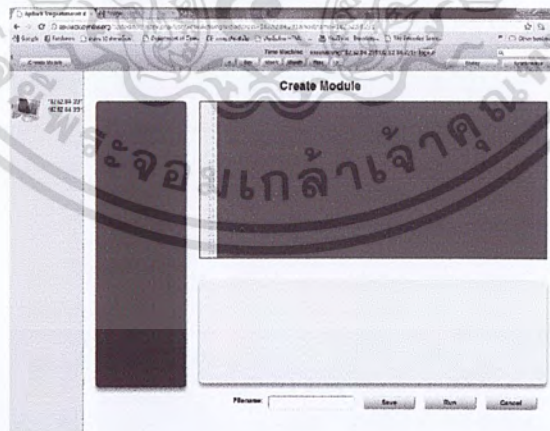
8.2 ผลการทดลองส่วนของ Integrated Development Environment (IDE)

8.2.1 หลักการออกแบบ

IDE ใช้สำหรับทดสอบและใช้งานภาษา CL โดยทำในรูปแบบของเว็บแอปพลิเคชัน ข้อแตกต่างของ IDE ที่สร้างขึ้นกับ IDE ทั่วไปคือ IDE ที่สร้างขึ้นใช้สำหรับการทดสอบและใช้งานจริง แต่ IDE ทั่วไปใช้เพียงแค่การทดสอบโปรแกรม เนื่องจากเราจะต้องใช้ IDE ดังกล่าวในการใช้งานภาษา CL จริงๆ ประกอบกับภาษา CL เป็นภาษาที่สั่งงานด้วยเหตุการณ์และเวลาเราจึงออกแบบ Interface ในรูปแบบของปฏิทินเพื่อความสะดวกในการใช้งาน ซึ่งการทำงานของ IDE จะคล้ายคลึงกับเพอซอแนลอออร์กานาไอเซอร์แต่ฟังก์ชันการใช้งานนั้นหลากหลายกว่ามาก โดยเพอซอแนลอออร์กานาไอเซอร์ทำได้เพียงการเตือนด้วยวิธีการต่างๆ แต่ IDE ที่สร้างขึ้นเป็นอออร์กานาไอเซอร์ที่สามารถโปรแกรมงานต่างๆที่คอมพิวเตอร์สามารถทำได้ให้ทำการตามเหตุการณ์ และเวลา

8.2.2 ส่วนประกอบของ IDE

- 1) หน้าต่างคลิเอทโมดูลเป็นหน้าต่างที่ใช้สำหรับเขียนโมดูล ภาษา CL หรือภาษาไพธอนเพื่อนำโมดูลดังกล่าวไปใช้ภายหลังบนปฏิทิน โดยอิดิเตอร์ที่ใช้เป็นของ codemirror ที่สามารถทำ syntax-highlighting โดยเราได้เพิ่มคุณสมบัติให้สามารถทำ auto-suggest เพื่อความสะดวกในการใช้งาน นอกจากนี้หน้าต่างดังกล่าวยังแสดงรายการ โมดูลที่ถูกสร้างขึ้นอีกด้วย



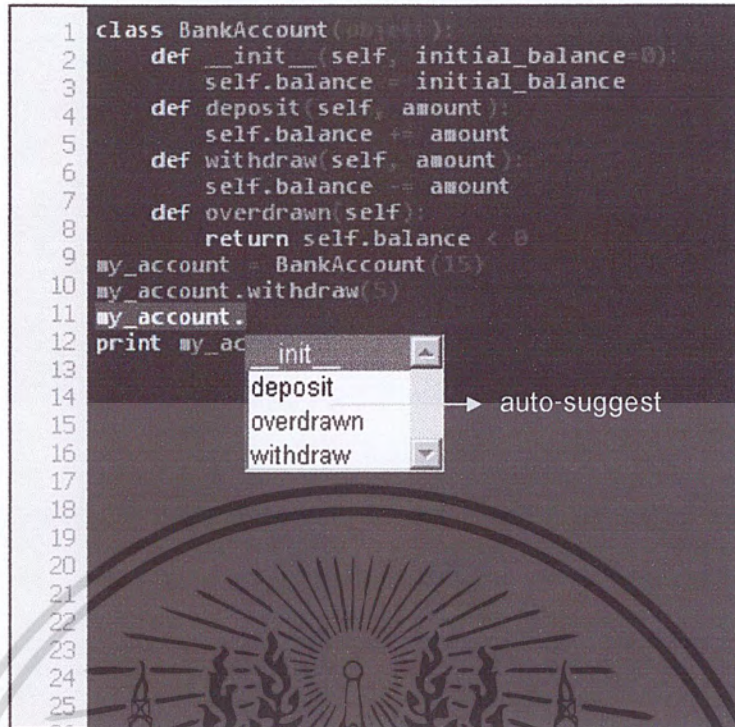
รูป 8.2 หน้าต่าง create module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

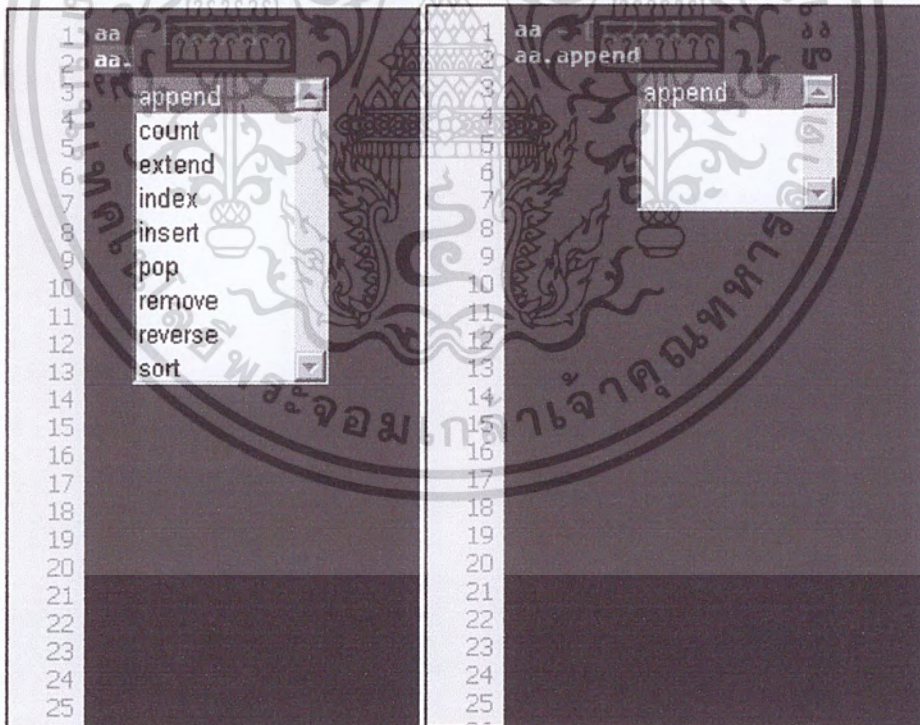
```

1 class BankAccount(object):
2     def __init__(self, initial_balance=0):
3         self.balance = initial_balance
4     def deposit(self, amount):
5         self.balance += amount
6     def withdraw(self, amount):
7         self.balance -= amount
8     def overdrawn(self):
9         return self.balance < 0
10 my_account = BankAccount(15)
11 my_account.withdraw(5)
12 my_account.
13 print my_ac
14
15
16
17
18
19
20
21
22
23
24
25

```



รูป 8.3 การ suggest ฟังก์ชันที่อยู่ใน class



```

1 aa
2 aa.
3
4 append
5 count
6 extend
7 index
8 insert
9 pop
10 remove
11 reverse
12 sort
13
14
15
16
17
18
19
20
21
22
23
24
25

```

```

1 aa
2 aa.append
3
4 append
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

```

รูป 8.4 การ suggest บิวท์อินฟังก์ชันที่อยู่ใน list

```

1 print 'Hello world'
2

```

รูป 8.5 การแสดงผลของ IDE โดยใช้สีที่ต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเนื้อหาไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) หน้าต่าง Calendar เป็นหน้าต่างที่ใช้สำหรับ add, edit และ remove โปรแกรมภาษา CL บนปฏิทิน โดยส่วนพื้นที่ในแต่ละวันเป็นอติเตอร์ที่ใช้สำหรับเขียน โปรแกรมภาษา CL ซึ่งสามารถเรียกใช้โมดูลที่ได้สร้างไว้ในหน้าต่างคลิเอทโมดูล
- 2.1) หน้าต่างส่วนที่ติดต่อกับผู้ใช้ในรูปแบบของ 'วัน' จะประกอบด้วยการแสดงผลคำสั่ง CL ทั้งหมดที่ได้ทำการบันทึกไว้ภายใน 24 ชั่วโมงของวันที่เลือก
 - 2.2) หน้าต่างส่วนที่ติดต่อกับผู้ใช้ในรูปแบบของ 'สัปดาห์' จะประกอบไปด้วยการแสดงผลคำสั่ง CL ทั้งหมดที่ได้ทำการบันทึกไว้ภายใน 24 ชั่วโมงของแต่ละวันที่อยู่ในสัปดาห์ที่เลือก
 - 2.3) หน้าต่างส่วนที่ติดต่อกับผู้ใช้ในรูปแบบของ 'เดือน' จะประกอบไปด้วยการแสดงผลคำสั่ง CL ทั้งหมดที่ได้ทำการบันทึกไว้ภายในเดือนที่เลือก
 - 2.4) หน้าต่างส่วนที่ติดต่อกับผู้ใช้ในรูปแบบของ 'ปี' จะประกอบไปด้วยการแสดงผลคำสั่ง CL ทั้งหมดที่ได้ทำการบันทึกไว้ภายในปีที่เลือก



รูป 8.6 รูปแบบของ IDE ในหน้าต่างของวัน และสัปดาห์



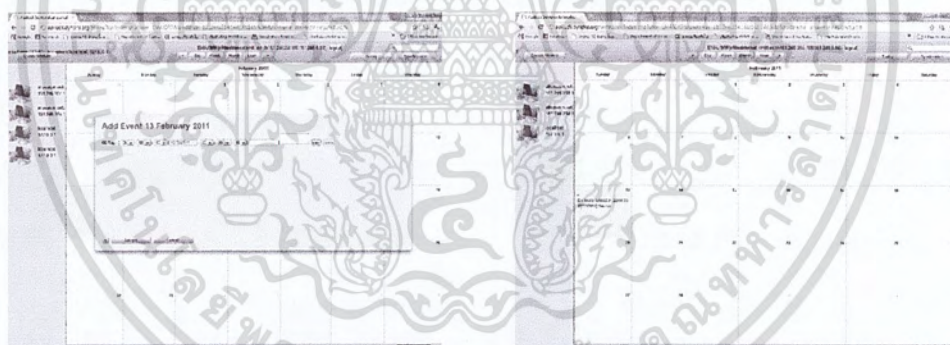
รูป 8.7 รูปแบบของ IDE ในหน้าต่างของเดือน และปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

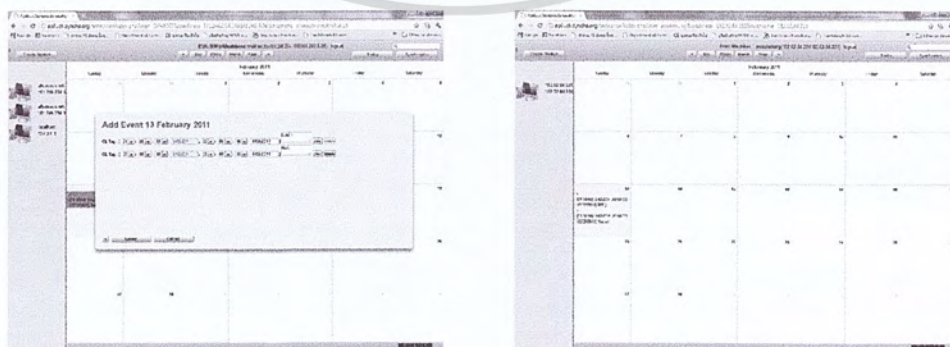
- 2.5) หน้าต่างส่วนที่ใช้ในการเพิ่ม CL statement จะถูกเรียกขึ้นมาเมื่อทำการดับเบิ้ลคลิกลงในพื้นที่ของวัน หรือ ปี ที่ต้องการสร้าง CL statement และสามารถทำการเพิ่ม CL statement ได้เรื่อยๆผ่านทางหน้าต่าง add อีเวนท์ ได้เลยโดยที่ไม่ต้องกลับไปทำการดับเบิ้ลคลิก อีกทั้งเมื่อทำการเพิ่ม CL statement แล้วจะมีการอัพเดทไปยังหน้าต่างส่วนแสดงผลทุกรูปแบบ



รูป 8.8 หน้าต่างที่ใช้ในการเพิ่ม CL statement

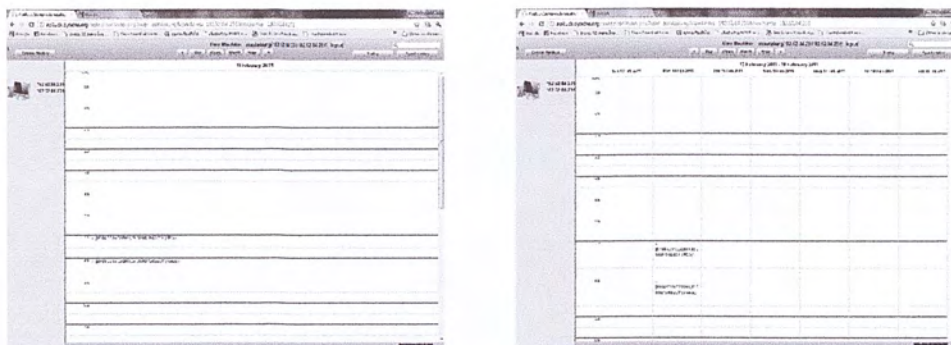


รูป 8.9 การเพิ่มเติม CL statement ลงใน IDE (1)

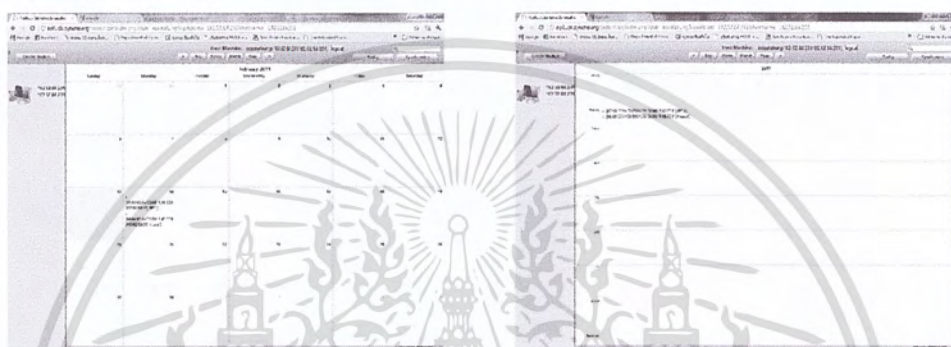


รูป 8.10 การเพิ่มเติม CL statement ลงใน IDE (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 8.11 รูปแบบของ IDE ในหน้าต่างของวัน และสัปดาห์ เมื่อเพิ่ม CL statement



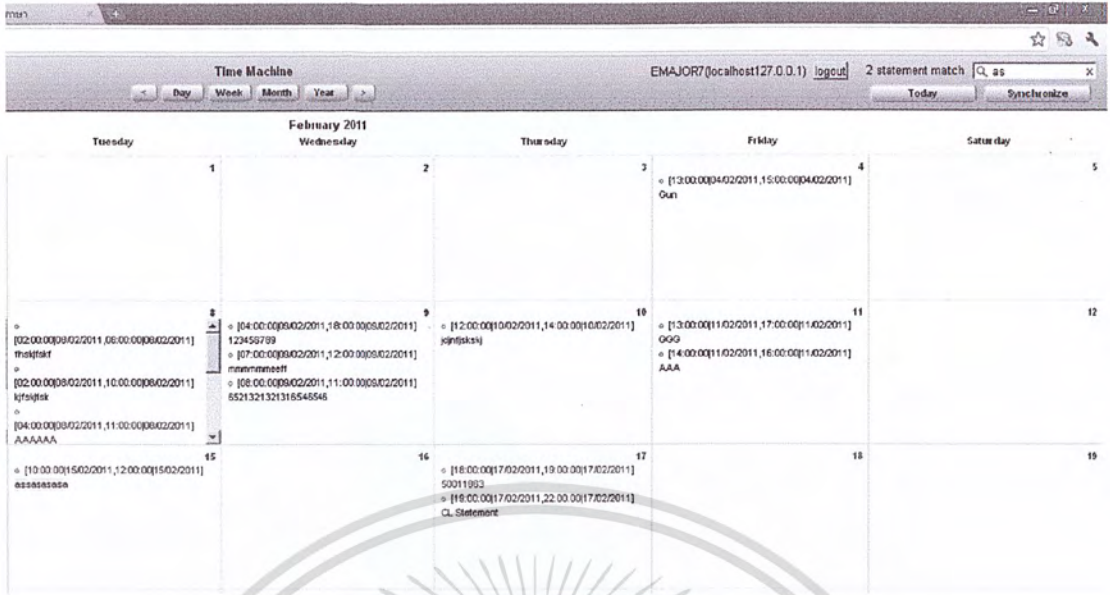
รูป 8.12 รูปแบบของ IDE ในหน้าต่างของเดือน และปี เมื่อเพิ่ม CL statement

3) พื้นที่ที่ใช้ในการค้นหา statement ที่อยู่ด้านบนของหน้าต่างปฏิทินเพื่อความสะดวกในการค้นหา CL statement และการ drag-drop ของ CL statement เพื่อความสะดวกในการเคลื่อนย้าย CL statement ไปในวันที่เราต้องการได้ทั้งในหน้าต่าง วัน, สัปดาห์, เดือน, ปี



รูป 8.13 การแสดงผลในการค้นหา (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 8.14 การแสดงผลในการค้นหา (2)

- 4) ส่วนของพื้นที่การแสดงผลในปฏิทินของทุกรูปแบบสามารถทำการ drag-drop ของ CL statement เพื่อความสะดวกในการเคลื่อนย้าย หรือแก้ไข CL statement โดยที่ไม่ต้องไปทำการแก้ไขภายในหน้าต่าง add อีเวนต์



รูป 8.15 การ drag-drop ของ CL statement

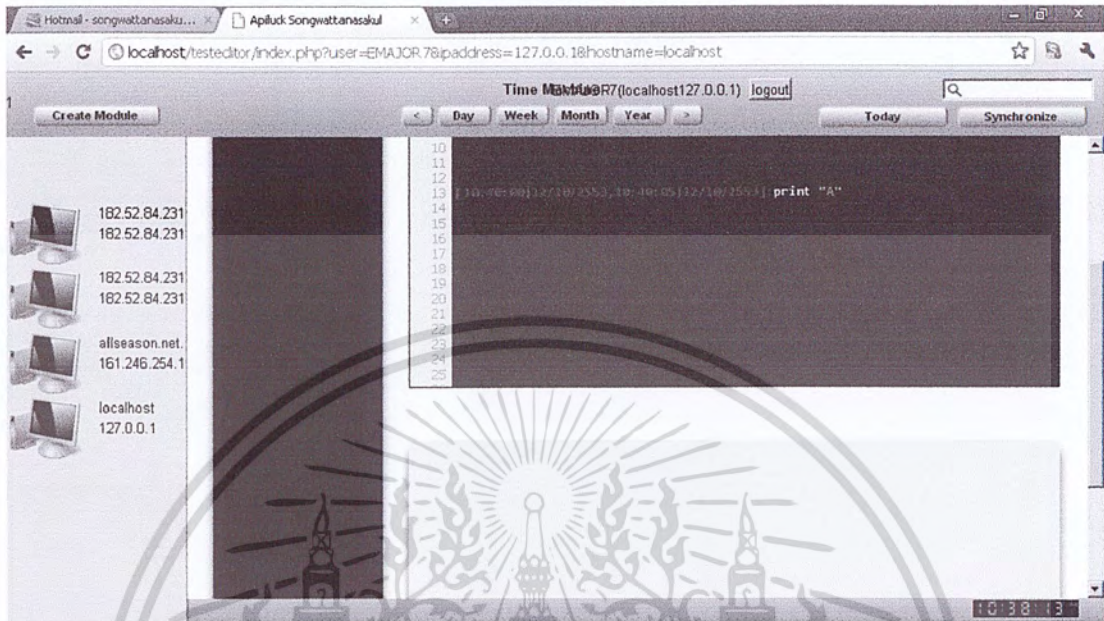
8.3 ผลการทดลองในส่วนของ CL Interpreter

การทดลองนั้นมีการเพื่อทำการทดสอบการทำงานในเชิงเวลา โดยการสร้างเหตุการณ์ โปรแกรมซึ่งทดลองรัน โดยใช้ซอร์สโค้ดด้านล่าง

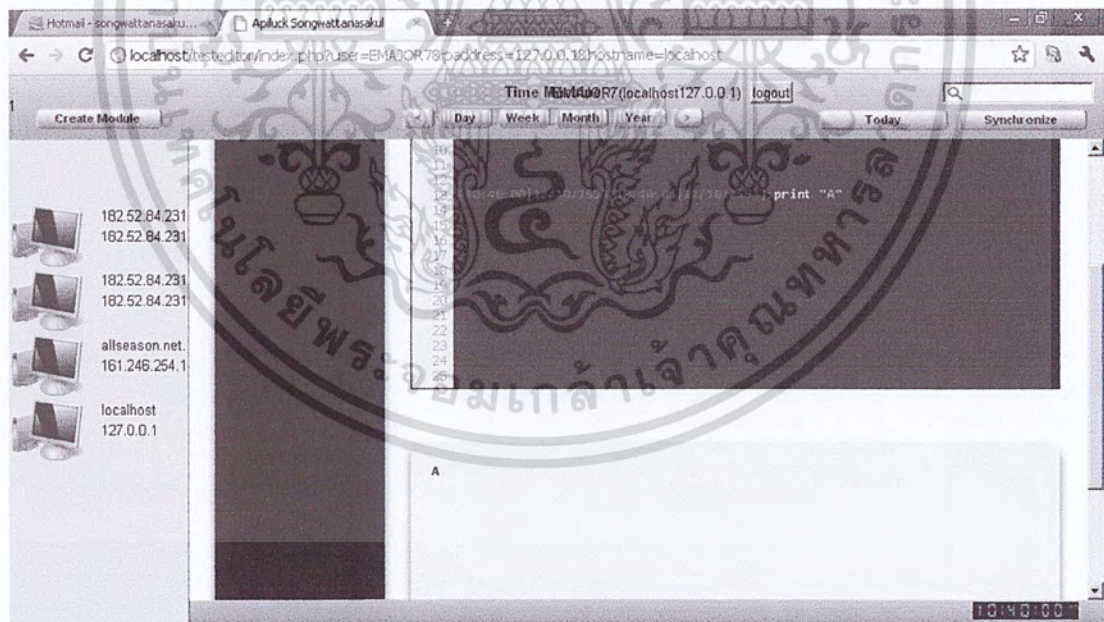
```
[10:40:00|12/10/2553,10:40:05|12/10/2553];print "A"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมด้านบนนี้จะทำการพิมพ์ตัวอักษร A ออกทางเอาต์พุตโดยจะเริ่มพิมพ์ระหว่าง 10:40:00 – 10:40:05 ดังรูป 8.15 และ 8.16



รูป 8.16 การทำงานช่วงเวลา 10:38:13



รูป 8.17 การทำงานช่วงเวลา 10:40:00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการทดลองกับอีก 1 คำสั่งด้านล่างซึ่งมีคำสั่งเป็นดังต่อไปนี้

```
[10:55:00|12/10/2553,10:55:02|12/10/2553]:
```

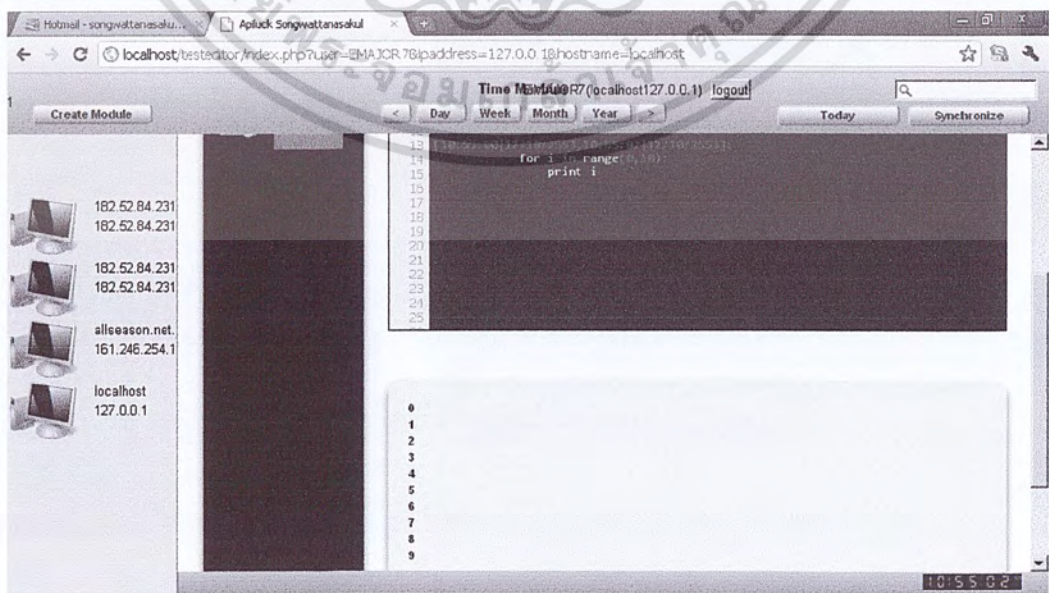
```
for i in range(0,10):
```

```
    print i
```

โปรแกรมด้านบนนี้จะทำการพิมพ์ค่าของ i ออกทางเอาต์พุตเมื่อ i มีค่าอยู่ระหว่าง 0 ถึง 9 โดยจะเริ่มพิมพ์ระหว่าง 10:55:00 – 10:55:02 ดังรูป 8.17 และ 8.18



รูป 8.18 การทำงานขณะเวลา 10:52:52



รูป 8.19 การทำงานขณะเวลา 10:55:02

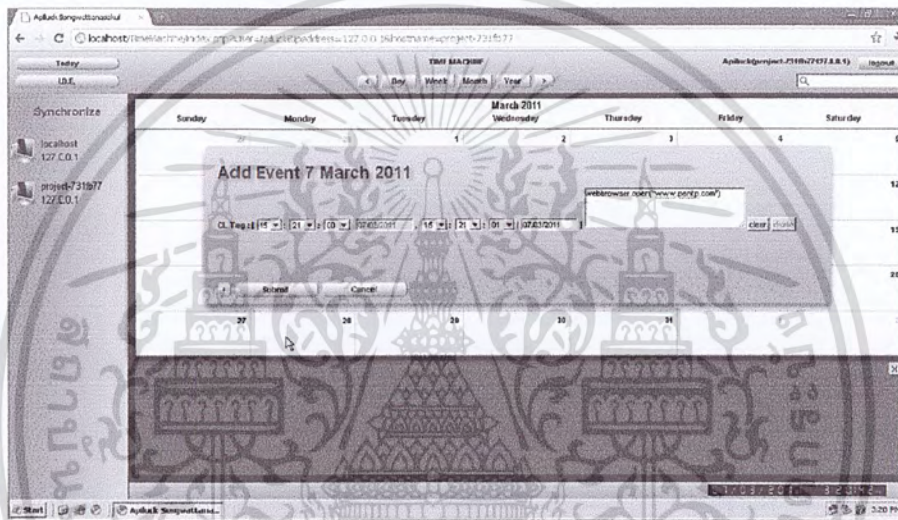
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปเล่มงานวิจัยที่จัดทำขึ้นเพื่อวัตถุประสงค์ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการทดลองกับอีก 1 คำสั่งด้านล่างซึ่งมีคำสั่งเป็นดังต่อไปนี้

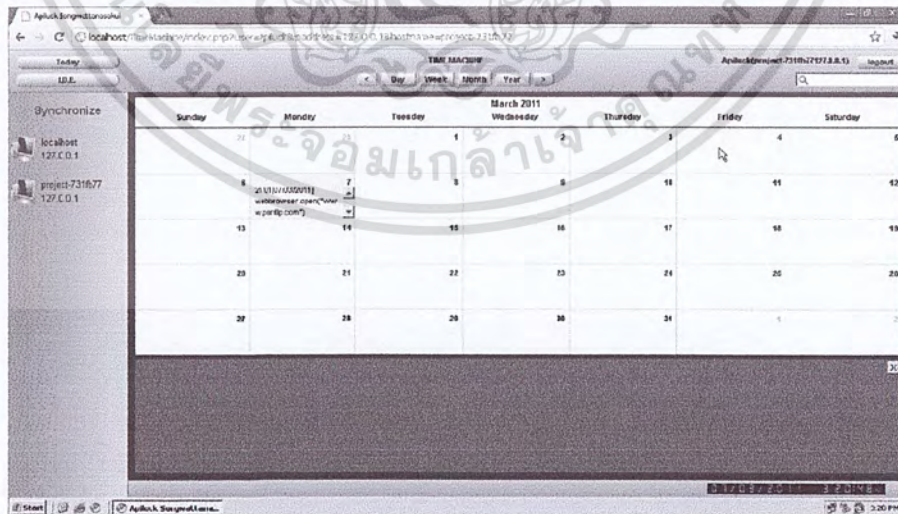
```
[15:21:00|07/03/2554,15:21:01|07/03/2554]:
```

```
webbrowser.open("www.pantip.com")
```

โปรแกรมด้านบนนี้จะทำงานเมื่อถึงเวลา 15: 21:00 แล้วหยุดทำคำสั่งเวลา 15:21:01 โดยคำสั่งนั้นจะทำการเปิดหน้าต่างของเว็บเบราว์เซอร์ซึ่งมี URL คือ www.pantip.com การทำงานจะเป็นไปดังรูป 8.19, รูป 8.20 และ รูป 8.21

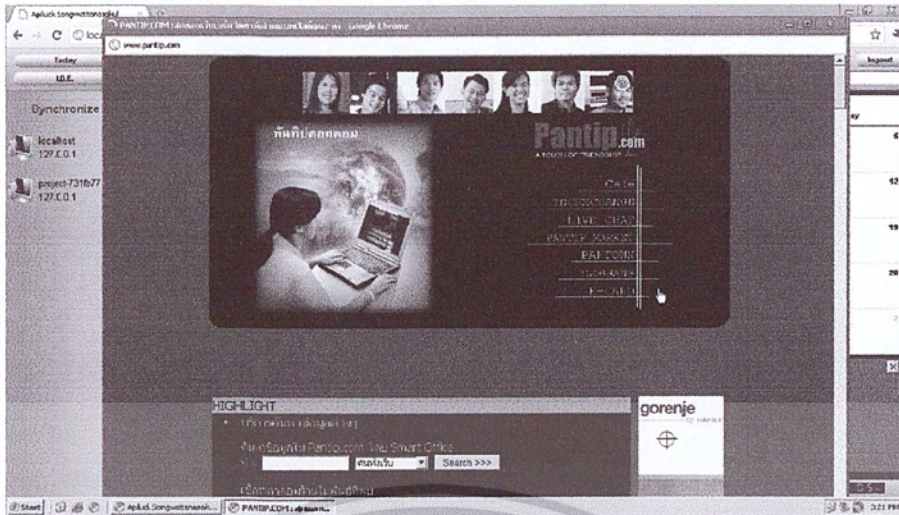


รูป 8.20 การนำคำสั่งทำงานในโปรแกรมขณะเวลา 15:20:42



รูป 8.21 การทำงานในโปรแกรมขณะเวลา 15:20:48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 8.22 เปิดเว็บเบราว์เซอร์เมื่อถึงเวลา 15:21:00



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

บทสรุปและวิจารณ์ผลการทดลอง

9.1 บทสรุป

ในโครงการนี้ได้ทำการสร้าง CL อินเทอร์เน็ตฟรีดเตอร์ และพัฒนา IDE โดยมีการเพิ่มเติมในส่วน
ของ CL อินเทอร์เน็ตฟรีดเตอร์เพื่อสามารถสั่งการในเชิงเวลาได้เป็นอย่างดีซึ่งอยู่ในรูปแบบของเว็บ
แอปพลิเคชัน โดย CL อินเทอร์เน็ตฟรีดเตอร์นั้นได้พัฒนาด้วยความรู้พื้นฐานในส่วนของอินเทอร์เน็ตฟรีด
เตอร์ด้วยหลักการของ Lexical analyser parser interpreter อีกทั้งอินเทอร์เน็ตฟรีดเตอร์สามารถใช้งาน
คำสั่งพื้นฐานในภาษาไพธอนตั้งแต่โอเปอเรเตอร์พื้นฐานต่างๆ ในส่วนของ Arithmetic operators,
Comparison operators, Logical operators, Assignment operator and Conditional operators
function และสามารถแปลภาษาในส่วนของออบเจกต์โอเรียนเตดให้สมบูรณ์แบบ จัดการเรื่องของ
inheritance, polymorphism ที่ค่อนข้างเป็นข้อจำกัดของภาษาจาวาสคริปต์โดยได้ใช้หลักการของ
การแปลภาษาอย่างถูกต้องและสามารถแปลภาษาในส่วนอื่นซึ่งทำงานได้ใกล้เคียงกับภาษาจริงอีก
ทั้งได้พัฒนาในส่วนของการจัดการเชิงเวลาในส่วนของ timetable, timeresponder รวมไปถึงการทำ
ให้ IDE ที่ใช้หัวข้อใหม่ของภาษา HTML5 มาช่วยให้การใช้งานแอปพลิเคชันนั้นมีประสิทธิภาพ
มากขึ้น สามารถเขียนโค้ดภาษา CL โดยได้สร้างรูปแบบของ user interface ในรูปแบบของปฏิทิน 4
มุมมอง ได้แก่ วัน สัปดาห์ เดือนและ ปี โดยฟังก์ชันเสริมของ IDE ที่สามารถ add statement CL ลง
ในปฏิทินได้อย่างถูกต้อง auto-suggest, search, synchronization และทำงานประสานกันระหว่าง
อินเทอร์เน็ตฟรีดเตอร์และ IDE ได้อย่างมีประสิทธิภาพ

9.2 แนวทางในการพัฒนาต่อ

ทำการพัฒนา IDE ที่ทำงานบนเว็บเบราว์เซอร์ให้สามารถทำงานบนโทรศัพท์เคลื่อนที่ หรือ
สมาร์ตโฟนได้อย่างมีประสิทธิภาพ

9.3 ข้อเสนอแนะ

ควรนำโปรแกรมที่ได้นั้นทำงานบนเบราว์เซอร์ที่สนับสนุนในส่วนของ HTML5 เพื่อให้ได้ผล
ออกมาอย่างมีประสิทธิภาพ

บรรณานุกรม

สุรัชย์ ล้อเจริญ. 2550. “การออกแบบและพัฒนาอินเทอร์เน็ตพริตเตอร์ภาษา CL.” วิทยานิพนธ์
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย, สถาบัน
เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

Kanda Runapongsa Saikaew. 2009. “Developing Offline Web Application.” ขอนแก่น:
มหาวิทยาลัย ขอนแก่น. เอกสาร

Apple developer guide. 2010. **Offline Calendar**. [online].
Available : <http://developer.apple.com/safaridemos/offline-calendar.php>

วิศิษฐ์ หิรัญกิตติ และ สุพัฒน์ดา โชติพันธ์. 2003. “CL: ภาษาสำหรับการใช้งานคอมพิวเตอร์ด้วย
เวลาและเหตุการณ์.” ห้องวิจัยการสื่อสารและคมนาคมชาวมหาวิทยาลัยวิศวกรรม
คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง. เอกสาร

Parr Terence. 2010. **Language Implementation Patterns**. 1st ed. North Carolina: The Pragmatic
Bookshelf

Peter Lubbers, Brian Albers and Frank Salim. 2010. **Pro HTML5 Programming**. 1st ed. New
York: Apress

Mark Pilgrim. 2010. **HTML5 Up and Running**. 1st ed. Cambridge: O'REILLY

Paul Wilton and Jeremy Mcpack. 2010. **Beginning Javascript**. 4th ed. Canada: Wrox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้