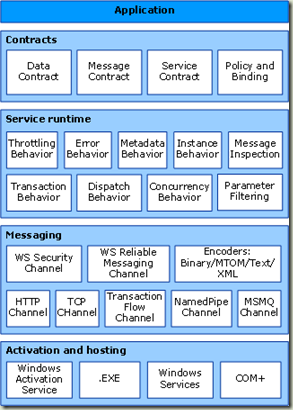
**บทที่ 2**

**ทฤษฎีและงานวิจัยที่เกี่ยวข้อง**

**2.1 ความรู้พื้นฐานเกี่ยวกับ Windows Communication Foundation (WCF)**

Windows Communication Foundation (WCF) มีวัตถุประสงค์คือ เพื่อช่วยเพิ่มความสามารถของ ASP.NET Web Service ซึ่งไมโครซอฟต์ตั้งใจพัฒนา WCF ขึ้นเพื่อรองรับลักษณะงานประเภท SOA ในกรณีที่มีระบบหรือผู้ใช้ (User) ที่มีความหลากหลายในด้านมาตรฐานการสื่อสาร(Protocol) WCF จะช่วยให้มีประสิทธิภาพการทำงานที่รวดเร็วขึ้นกว่าเดิม สามารถนำ Service ที่พัฒนาไว้นั้นไปรันบนโฮสต์ชนิดต่างๆ ได้



**รูปที่ 2.1 โครงสร้างพื้นฐานของ WCF โดยได้แบ่งเอาไว้เป็นชั้นๆ (Layer)**

* Application Layer คือส่วนที่ทำการ implement code ตัวเซอร์วิสไว้ เช่น method, code logic, entity ต่างๆ ที่ใช้ในการทำงานของตัวเซอร์วิส
* Contract Layer เป็นชั้นที่เชื่อมต่อกับ Application Layer ซึ่งนักพัฒนาจะทำงานโดยตรงกับส่วนนี้เพื่อใช้ในการสร้างตัวเซอร์วิสต่างๆ สามารถแบ่งได้ ดังนี้

1. **Data Contract**  เป็นส่วนที่กำหนดโครงสร้างข้อมูลต่าง ๆ ที่มีรับส่งข้อมูลทั้งเข้าและออกโดย Operation Contract ของ Service Contracts ซึ่งอาจจะประกอบไปด้วย ชนิดข้อมูลพื้นฐาน (Primitive Data Type) เช่น string, int, decimal, float, double, boolean  และกลุ่มข้อมูลโครงสร้างที่ซับซ้อน (Complex Data Type) อย่างเช่น Class, Enumeration Data ต่างๆ
2. **Message Contract** โดยปกติข้อมูลที่ส่งข้ามกันระหว่าง Service และ Client นั้น ถูกกำหนดรูปแบบอยู่ใน SOAP Message แต่บางครั้งเนื่องจากความต้องการในการรองรับการทำงานต่างๆ เช่น ต้องการจัดรูปแบบโครงสร้างข้อมูลใหม่, อยากใช้ Header, การเพิ่มเติม Status เหล่านี้  เราจะต้องอาศัย Message Contract เป็นตัวช่วย
3. Service Contracts ส่วนที่บ่งบอกว่าเซอร์วิสที่เราสร้างขึ้นนั้นมี operation (Method) อะไรไว้ให้บริการบ้าง
4. **Policies and Binding** ส่วนนี้เป็นตัวกำหนดตั้งค่าเงื่อนไขและวิธีการให้เซอร์วิสมีการติดต่อสื่อสารในรูปแบบใด(Protocol) ใช้ระบบการรักษาปลอดภัย(Security) แบบไหน

- Service Runtime ส่วนที่กำหนดพฤติกรรม (Behavior) การทำงานของเซอร์วิส ขณะที่เซอร์วิสทำงานอยู่ มีสามารถแบ่งได้ ดังนี้

1. Throttling Behavior ใช้กำหนดจำนวนของเมสเสจที่จะเข้ามาทำงานในเซอร์วิส
2. Error Behavior หากมีปัญหาเกิดขึ้นภายในเซอร์วิส ใช้มาช่วยทำการจัดการปัญหา ดังกล่าว
3. Metadata Behavior ใช้กำหนดการเปิดเผยข้อมูลโครงสร้าง (MetaData) ของเซอร์วิส แก่ผู้ใช้งาน
4. Instance Behavior ใช้กำหนดจำนวนของผู้ใช้ที่จะเข้ามาใช้งานในเซอร์วิสโดยพร้อม ๆ กันขณะที่เซอร์วิสทำงานอยู่
5. Transaction Behavior ช่วยในการย้อนกลับ (Rollback) การทำงานหากโอเปอเรชั่นเกิดการทำงานที่ผิดพลาดขึ้น
6. Dispatch Behavior เป็นตัวควบคุมว่าเมสเสจต่างๆ นั้นจะเข้าการโปรเซสด้วยสถาปัตยกรรมใดของ WCF

#### ****Messaging**** Layer เป็นการจัดการเมสเสจที่ส่งเข้าออกภายในเซอร์วิส ทั้งการกำหนด รูปแบบของเมสเสจ โดยการวิธีการแลกเปลี่ยนเมสเสจซึ่งพอจะแบ่งได้เป็นสองกลุ่มดังนี้

1. Transport channels ทำการอ่านและเขียนเมสเสจจากเน็ตเวิร์กภายในหรือภายนอกองค์กร มีบาง transports จำเป็นต้องมีการแปลง (encoder) เมสเสจจากหน่วยสตรีมไบท์ที่ใช้รับส่งกันในเน็ตเวิร์กมาเป็น XML เช่น HTTP, named pipes, TCP, และ MSMQ
2. Protocol channels เป็นการทำงานโปรโตคอลระดับเมสเสจ โดยส่วนใหญ่เป็นการเขียนอ่านในส่วนของหัวเมสเสจเช่น WS-Security Protocol เป็นการกำหนดใช้ความปลอดภัยในระดับชั้นเมสเสจ และ WS-Reliability Protocol เป็นการการันตีการส่ง เมสเสจระหว่างกัน

#### **Activation and Hosting**เซอร์วิสที่ได้สร้างไว้เป็นเพียงไลบราลี่เท่านั้น เมื่อต้องการนำไปรันบนโฮส เพื่อให้เซอร์วิสสามารถใช้งานได้ ซึ่งมีหลายวิธีดังนี้

#### Windows Activation Service (WAS) มาพร้อมกับ IIS 7.0 จะทำงานเมื่อมีการร้องขอบริการเข้ามา รองรับเกือบทุกโปรโตคอล HTTP, TCP, Named Pipes, MSMQ

1. Self-Hosting executeable ได้เช่น console app, win app, WPF app เป็นต้น ไม่สามารถทำงานอัตโนมัติได้ ต้องรันโปรแกรมทุกครั้งเพื่อให้โฮสทำงาน
2. Windows Service คล้ายกับ Executable โฮส แต่จะทำงานอัตโนมัติทุกครั้งที่ เริ่มระบบ Windows
3. COM+ โฮสตัวนี้ส่วนใหญ่ไม่มีคนได้ใช้งาน เพราะค่อนข้างจะลึกลึกเรื่อง Windows Component Programming

**2.2 เว็บ****แอปพลิเคชัน (Web Application)**

2.2.1 ความหมายของเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันเป็นแอปพลิเคชันที่สามารถเรียกใช้งานผ่านบราวเซอร์ต่างๆ โดยที่ผู้ใช้งานไม่ต้องติดตั้งแอปพลิเคชันเหล่านี้ ลงไปในเครื่องคอมพิวเตอร์ ทำให้สามารถเรียกใช้งานได้ทุกที่ทุกเวลา

2.2.2 การสร้างเว็บแอปพลิเคชัน

การสร้างเว็บแอปพลิเคชันเหมือนกับการสร้างซอฟต์แวร์อื่นๆ ที่ผู้สร้างจะต้องใช้ความคิดทางตรรกะและความคิดสร้างสรรค์ร่วมกันเพื่อให้แอปพลิเคชันที่สร้างขึ้นมานั้นสามารถใช้งานได้ตามวัตถุประสงค์ที่ตั้งไว้ และทำให้เกิดความสนใจจากกลุ่มเป้าหมาย ในส่วนของขั้นตอนการสร้างจะต้องเตรียมการดังนี้

1) จัดเก็บข้อมูลความต้องการของระบบว่าเว็บแอปพลิเคชันที่สร้างสามารถทำอะไรได้บ้าง และสร้างมาเพื่อตอบสนองต่อกลุ่มเป้าหมายกลุ่มใด ซึ่งข้อมูลเหล่านี้เป็นประโยชน์ต่อคนออกแบบหน้าเว็บ (User Interface) และคนเขียนโปรแกรม

2) เตรียมเครื่องมือหรือโปรแกรมสาหรับสร้างเว็บแอปพลิเคชัน ซึ่งเครื่องมือที่ใช้สาหรับการเขียนเว็บแอปพลิเคชันมีให้เลือกมากมาย อาทิเช่น Sublime text, DreamWeaver เป็นต้น นอกจากนี้ต้องจัดเตรียมเครื่องมือที่ใช้ในการจัดการฐานข้อมูล เช่น Apache และ Mysql สำหรับการประมวลผลและการจัดการฐานข้อมูล

3) เริ่มการออกแบบโดยการร่างหน้าเว็บแต่ละหน้า

4) เมื่อการออกแบบเสร็จเรียบร้อยแล้ว จะเข้าสู่ขั้นตอนของการสร้างโครงสร้างของเว็บ โดยทำเว็บต้นแบบขึ้นมา เพื่อให้เค้าโครงของเว็บแต่ละหน้ามีความใกล้เคียงกัน

5) จากนั้นจึงเริ่มเขียนโปรแกรมให้ได้ตรงตามฟังก์ชันที่ได้ออกแบบไว้

6) ตรวจสอบการทำงานของระบบ เพื่อไม่ให้เกิดข้อผิดพลาด

**2.3 ความรู้พื้นฐานเกี่ยวกับ Application Programming Interface (API)**

**Application Programming Interface (API)** คือช่องทางการเชื่อมต่อระหว่างเว็บไซต์หนึ่งไปยังอีกเว็บไซต์หนึ่ง หรือเป็นการเชื่อมต่อระหว่างผู้ใช้งานกับ Server หรือจาก Server เชื่อมต่อไปหา Server ซึ่ง API เปรียบได้เป็นภาษาคอมพิวเตอร์ที่ทำให้คอมพิวเตอร์สามารถสื่อสารและแลกเปลี่ยนข้อมูลกันได้อย่างอิสระ

API ทำหน้าที่ช่วยในการเข้าถึงข้อมูลต่างๆหรือจะเป็นการจัดการข้อมูลต่างๆบนเว็บไซต์ โดยเจ้าของเว็บไซต์ที่มี API สามารถกำหนดขอบเขตในการเข้าถึงบริการต่างๆของทางเว็บไซต์ ประโยชน์ของ API สามารถแบ่งออกมาได้หลายอย่างคือ

1) ช่วยในการพัฒนาเว็บไซต์หรือ Application ได้ง่ายและรวดเร็วซึ่ง API จะเป็นตัวช่วยที่นักพัฒนาไม่ต้องเข้าไปแก้ไข Code คำสั่งเลยทำให้สะดวกสบายในการใช้งาน  
2) ช่วยให้นักพัฒนาเว็บไซต์หรือเจ้าของเว็บไซต์สามารถขยายฐานผู้ชมเว็บไซต์ให้มากขึ้น  
3) ทำให้ผู้ใช้งานเว็บไซต์ต่างๆที่มีการติดตั้ง API ของอีกเว็บไซต์หนึ่ง ไม่ต้องเข้าหน้าเว็บไซต์ที่เป็นเจ้าของ API เพียงแต่เข้ามายังเว็บไซต์ที่มีการติดตั้ง API เท่านั้นทำให้การรับรู้ข่าวสารต่าง ๆทั่วถึงกันและสะดวกในการใช้งานของผู้ใช้งานเว็บไซต์  
4) API สามารถรับส่งข้อมูลข้าม Server ได้

ปัจจุบันเว็บไซต์ใหญ่ๆ หลายเว็บไซต์จะมีการเปิดให้ใช้งาน API ซึ่งเราอาจจะเห็นการใช้งาน API ได้มากขึ้นโดยเฉพาะเว็บไซต์ที่ด้านการติดต่อสื่อสาร Social Network และ E-commerce เช่น บริการของ Amazon มี API ที่เปิดให้ผู้ที่สนใจที่จะเป็นตัวแทนขายสินค้าหรือเจ้าของเว็บทั่วไป ได้นำสินค้าที่มีขายอยู่ใน Amazon ไปติดไว้ในเว็บไซต์หรือบล็อคของตัวเองได้ โดยเจ้าของเว็บไซต์หรือผู้สนใจจะได้รับคอมมิสชั่นเมื่อมีการคลิกซื้อสินค้า จากเว็บไซต์หรือบล็อคที่นำ API ไปติดตั้ง

**2.4 ความรู้พื้นฐานเกี่ยวกับภาษาซีชาร์ป (C# Language)**

C# คือ ภาษาคอมพิวเตอร์ประเภทการเขียนโปรแกรมเชิงวัตถุ (Object - Oriented Programming หรือ OOP) พัฒนาโดยไมโครซอฟท์ (Microsoft) โดยมีจุดมุ่งหมายในการวมความสามารถการคำนวณของ C++ ด้วยการโปรแกรมที่ง่ายกว่าของ Visual Basic โดย C# มีพื้นฐานจาก [C++](http://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2183-c++-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html)และเก็บส่วนการทำงานคล้ายกับ [Java](http://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2185-java-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html)  
     C# ได้รับการออกแบบให้ทำงานกับ .NET platform ของไมโครซอฟท์ จุดมุ่งหมายคือ อำนวยความสะดวกในการแลกเปลี่ยนสารสนเทศและบริการผ่านเว็บ และทำให้ผู้พัฒนาสร้างโปรแกรมประยุกต์ในขนาดกะทัดรัด C# ทำให้โปรแกรมง่ายขึ้นผ่านการใช้ Extensible Markup Language (XML) และ Simple Object Access Protocol (SOAP) ซึ่งยอมให้เข้าถึงวัตถุ (Object) ของโปรแกรมหรือเมธอด โดยปราศจากความต้องการให้ผู้เขียนโปรแกรมเขียนคำสั่งเพิ่มในแต่ละขั้นตอน เนื่องจากผู้เขียนโปรแกรมสามารถสร้างบนคำสั่งที่มีอยู่ แทนที่การคัดลอกซ้ำ C#  ภาษา C# ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการการนำข้อดีของภาษาต่างๆ (เช่นภาษา Delphi, ภาษา C++) มาปรับปรุงเพื่อให้มีความเป็นโปรแกรมเชิงวัตถุมากขึ้น ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง ซึ่งเรียบง่ายกว่าภาษา C++ และมีสิ่งที่เกินความจำเป็นน้อยลงเมื่อเทียบกับJava  
     C# ถูกรับรองจากหน่วยงาน ECMA ซึ่งเป็นหน่วยงานกำหนดมาตรฐานสากลด้านสารสนเทศ และ ISO และปัจจุบันไมโครซอฟท์ยังพัฒนาภาษานี้อย่างต่อเนื่อง

**2.5 ความรู้พื้นฐานเกี่ยวกับกิต (Git)**

**Git** คือ Version Control ประเภทหนึ่ง ซึ่งเป็นระบบที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจ็ค มีการ backup code โดยสามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชั่นต่างๆของ โปรเจ็คที่ใดเวลาใดก็ได้ หรือแม้แต่ดูว่าไฟล์นั้นๆใครเป็นคนเพิ่มหรือแก้ไข หรือจะดูว่าไฟล์นั้นๆถูกเขียนโดยใครบ้างก็ได้ ทำให้ **Git** เหมาะอย่างยิ่งสำหรับนักพัฒนาไม่ว่าจะเป็นคนเดียวหรือจะมีประสิทธิภาพมากขึ้นหากเป็นการพัฒนาเป็นทีม **โดย Git มีคุณสมบัติคร่าวๆดังนี้**

- สามารถ Backup และ Restore ไฟล์ได้

- Synchronization ไฟล์ทำให้ไฟล์นั่นเป็นเวอร์ชันใหม่เสมอไม่ว่าจะมีผู้ใช้งานไฟล์นั่นกี่คนก็ตาม

- สามารถ Track back history ของแต่ละไฟล์ได้

- ใน History นั่นสามารถบอกรายละเอียดได้มากกว่าการจัดเก็บแบบธรรมคือสามารถบอกได้ว่า

ไฟล์นี้ใครเป็นเจ้าของ และใครบ้างที่แก้ไขไฟล์นี้

- ควบคุมจัดการได้ง่ายเมื่อต้องการควบคุมโค้ดที่พัฒนาโดยโปรแกรมเมอร์หลายคนในทีม

**2.6 ความรู้พื้นฐานเกี่ยวกับเอชทีเอ็มแอล (Hyper Text Markup Language: HTML)**

2.6.1 ความหมายของเอชทีเอ็มแอล

เอชทีเอ็มแอล เป็นภาษามาร์กอัป (Markup) ที่ใช้ในการสร้างเว็บเพจ ถูกพัฒนาและ กำหนดมาตรฐานโดยองค์กร World Wide Web Consortium (W3C) สำหรับการสร้างเว็บเพจโดยใช้ภาษาเอชทีเอ็มแอล สามารถทำได้โดยใช้โปรแกรมเท็กซ์อีดิตเตอร์ (Text Editor) หรือเครื่องมือช่วยสร้างเว็บเพจ เช่น ดรีมเวฟเวอร์ ส่วนการเรียกใช้งานหรือทดสอบการทำงานของเอกสารเอชทีเอ็มแอลจะใช้โปรแกรมอินเทอร์เน็ตบราวเซอร์ เช่น โครม อินเทอร์เน็ตเอกซ์พลอเรอร์, ไฟร์ฟอกซ์ เป็นต้น

2.6.2 โครงสร้างคำสั่งของเอชทีเอ็มแอล

โครงสร้างของเอชทีเอ็มแอลจะประกอบไปด้วยส่วนของคาสั่ง 2 ส่วน คือ ส่วนที่เป็น ส่วนหัว (Head) และส่วนที่เป็นเนื้อหา (Body) โดยมีรูปแบบคำสั่งดังนี้

1) คำสั่งหรือแท็ก (Tag)

แท็กคือคาสั่งที่ใช้ในภาษาเอชทีเอ็มแอลอยู่ภายในเครื่องหมายน้อยกว่า (<) และเครื่องหมายมากกว่า (>) แท็กในภาษาเอชทีเอ็มแอลแบ่งออกเป็นสองชนิดดังนี้

* แท็กเดี่ยว เป็นแท็กที่ไม่ต้องมีแท็กปิด เช่น <HR> <BR>
* แท็กคู่ เป็นแท็กที่ประกอบด้วยแท็กเปิดและแท็กปิด โดยแท็กปิดจะต้องใส่เครื่องหมายทับ (/) นำหน้าคำสั่งในแท็กนั้น ๆ เช่น <b>....</b>, <p>….</p> เป็นต้น
* <HTML>..........</HTML>
* คาสั่ง <HTML> เป็นคาสั่งเริ่มต้นในการเขียนโปรแกรมและคาสั่ง </HTML> เป็นการสิ้นสุดโปรแกรมเอชทีเอ็มแอล คาสั่งนี้จะไม่แสดงผลในโปรแกรมเว็บเบราเซอร์ แต่ต้องเขียนเพื่อให้เกิดความเป็นระบบของงาน และเพื่อแจ้งให้เว็บเบราเซอร์ทราบว่าเอกสารนี้เป็นเอกสารของภาษาเอชทีเอ็มแอล
* <HEAD>..........</HEAD>
* ส่วนหัวเป็นส่วนที่ใช้อธิบายข้อมูลเฉพาะของหน้าเว็บนั้น ๆ เช่น ชื่อเรื่องของหน้าเว็บ (Title) ชื่อผู้จัดทาเว็บ (Author) คีย์เวิร์ดสาหรับการค้นหา (Keyword) และยังใช้ในการระบุไฟล์รหัสต้นฉบับ (Source Code) ต่าง ๆ โดยมีแท็กสำคัญ คือ
* <TITLE>..........</TITLE>

ข้อความที่ใช้เป็นชื่อเรื่อง (TITLE) ไม่ควรพิมพ์เกิน 64 ตัวอักษร ไม่ต้องใส่ลักษณะพิเศษ เช่น ตัวหนา เอียง หรือสี และควรใช้ภาษาที่มีความหมายครอบคลุมถึงเนื้อหาของเว็บเพจนั้น หรือเป็นคาสาคัญในการค้นหา (Keyword)

- <BODY>..........</BODY>

ส่วนเนื้อหาเป็นส่วนเนื้อหาหลักของหน้าเว็บ ซึ่งการแสดงผลจะต้องใช้แท็กจานวนมาก ขึ้นอยู่กับลักษณะของข้อมูล เช่น ข้อความ รูปภาพ เสียง วีดิโอ หรือไฟล์ต่าง ๆ การป้อนคาสั่งในส่วนนี้ ไม่มีข้อจากัดสามารถป้อนติดกัน หรือ 1 บรรทัดต่อ 1 คาสั่ง แต่ส่วนใหญ่จะยึดรูปแบบที่อ่านง่ายคือ การทาย่อหน้าในชุดคาสั่งที่เกี่ยวข้องกัน โดยคาสั่งทั้งหมดดจะถูกป้อนภายใต้แท็ก <BODY>… </BODY>

2) แอทริบิวต์ (Attribute)

แอทริบิวต์เป็นส่วนขยายในแท็ก ใช้สาหรับจัดรูปแบบเพิ่มเติม เช่น ขนาด สีระยะห่าง เป็นต้น ค่าของแอทริบิวต์จะอยู่ในเครื่องหมาย “…” เช่น

<p align="right"> ข้อความในย่อหน้านี้จัดวางในตำแหน่งชิดขวาหน้าจอ</p>

<a href = “https://www.google.com” > ไปยังเว็บไซต์ google.com </a>

**2.7 ความรู้พื้นฐานเกี่ยวกับภาษาซีเอสเอส (Cascading Style Sheets: CSS)**

**CSS** เป็นภาษาที่มีรูปแบบการเขียน Syntax ที่เฉพาะและถูกกำหนดมาตรฐานโดย W3C (World Wide Web Consortium) เช่นเดียวกับ HTML และ XHTML ใช้สำหรับตกแต่งเอกสาร HTML/ XHTML ให้มีหน้าตา สีสัน ตัวอักษร เส้นขอบ พื้นหลัง ระยะห่าง ฯลฯ อย่างที่เราต้องการ ด้วยการกำหนดคุณสมบัติให้กับ Element ต่างๆ ของ HTML เช่น <body>, <p>, <h1> เป็นต้น

#### ประโยชน์ของ CSS

1. การใช้ CSS ในการจัดรูปแบบการแสดงผล จะช่วยลดการใช้ภาษา HTML ในการตกแต่งเอกสารเว็บเพจ ทำให้ code ภายในเอกสาร HTML เหลือเพียงส่วนเนื้อหา ทำให้เข้าใจง่ายขึ้น การแก้ไขเอกสารทำได้ง่ายและรวดเร็ว  
  
2. เมื่อ code ภายในเอกสาร HTML ลดลง ทำให้ขนาดไฟล์เล็กลง จึงดาวน์โหลดได้เร็ว  
  
3. สามารถกำหนดรูปแบบการแสดงผลจากคำสั่ง style sheet ชุดเดียวกัน ให้มีผลกับเอกสาร HTML ทั้งหน้า หรือทุกหน้าได้ ทำให้เวลาแก้ไขหรือปรับปรุงทำได้ง่าย ไม่ต้องไล่ตามแก้ที่ HTML tag ต่างๆ ทั่วทั้งเอกสาร   
  
4. สามารถควบคุมการแสดงผลให้เหมือนกัน หรือใกล้เคียงกัน ได้ในหลาย Web Browser  
  
5. สามารถกำหนดการแสดงผลในรูปแบบที่เหมาะกับสื่อชนิดต่างๆ ไม่ว่าจะเป็นการแสดงผลบนหน้าจอ, บนกระดาษเมื่อสั่งพิมพ์, บนมือถือ หรือบน PDA โดยที่เป็นเนื้อหาเดียวกัน  
  
6. ทำให้เป็นเว็บไซต์ที่มีมาตรฐาน ปัจจุบันการใช้ attribute ของ HTML ตกแต่งเอกสารเว็บเพจ นั้นล้าสมัยแล้ว W3C แนะนำให้เราใช้ CSS แทน ดังนั้นหากเราใช้ CSS กับเอกสาร HTML ของเรา ก็จะทำให้เข้ากับเว็บเบราเซอร์ในอนาคตได้ดี

**2.6 ความรู้พื้นฐานเกี่ยวกับจาวาสคริปต์ (JavaScript)**

2.6.1 ความหมายของจาวาสคริปต์

จาวาสคริปต์เป็นภาษาสคริปต์ที่มีลักษณะการเขียนเชิงโพรโทไทป์ (Prototype-based Programming) ได้รับการพัฒนาขึ้นโดยเน็ตสเคป (Netscape) โดยมีวัตถุประสงค์ เพื่อช่วยให้เว็บเพจสามารถแสดงเนื้อหาที่มีการเปลี่ยนแปลงได้ตามเงื่อนไขหรือสภาพแวดล้อมต่าง ๆ หรือสามารถตอบโต้ตอบกับผู้ใช้งานได้มากขึ้น ทั้งนี้เพราะภาษาเอชทีเอ็มแอลไม่สามารถโต้ตอบกับผู้ใช้งานได้ทันที แต่ใช้แสดงผลเอกสารที่มีเนื้อหาคงที่แน่นอน

2.6.2 การทำงานของจาวาสคริปต์

การทางานของจาวาสคริปต์เกิดขึ้นบนบราวเซอร์ (Client-Side Script) มีข้อจากัดคือ ไม่สามารถรับและส่งข้อมูลต่าง ๆ กับเซิร์ฟเวอร์ได้โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนามาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ใช้งาน เพื่อไปเก็บบนเซิร์ฟเวอร์ เป็นต้น ดังนั้นงานในลักษณะดังกล่าว ต้องอาศัยภาษาฝั่งเซิร์ฟเวอร์ เช่น พีเอชพี เพิร์ล (Perl) เอเอสพี (ASP)

ปัจจุบันมีการใช้จาวาสคริปต์ที่ฝังอยู่ในเว็บเบราว์เซอร์หลายรูปแบบ เช่น ใช้เพื่อสร้างเนื้อหาที่เปลี่ยนแปลงเสมอภายในเว็บเพจ ใช้เพื่อตรวจสอบความถูกต้องของข้อมูลที่ผู้ใช้กรอกก่อนนาเข้าระบบ ใช้เพื่อเข้าถึงข้อมูลที่อยู่ภายใต้โครงสร้างแบบดอม เป็นต้น

**2.7 ความรู้พื้นฐานเกี่ยวกับเจคิวรี่ (jQuery)**

2.7.1 ความหมายของเจคิวรี่

เจคิวรี่คือไลบารี่ของภาษาจาวาสคริปต์ ซึ่งถูกออกแบบมาเพื่อให้การเขียนจาวาสคริปต์นั้นง่ายขึ้น เพราะสามารถเขียนโค้ดยาว ๆ ของจาวาสคริปต์ให้สั้นลงเหลือเพียงบรรทัดเดียว เจคิวรี่ถูกพัฒนาโดย John Resig โดยเป็นการพัฒนาแบบโอเพนซอร์ส

2.7.2 คุณสมบัติหลักของเจคิวรี่

1) การเข้าถึงเอชทีเอ็มแอลและซีเอสเอส

ส่วนนี้ถือได้ว่าเป็นคุณสมบัติหลักของเจคิวรี่ เพราะทาให้ผู้ใช้งานสามารถเข้าถึงอีลีเมนต์เอชทีเอ็มแอลและตัวเลือกซีเอสเอสต่าง ๆ ได้ง่ายขึ้น

2) การจัดการเอชทีเอ็มแอลและซีเอสเอส

คุณสมบัติในหัวข้อที่แล้วเกี่ยวกับการเข้าถึงอีลีเมนต์ต่าง ๆ แต่สาหรับคุณสมบัตินี้จะใช้ในการดัดแปลง แก้ไข เพิ่ม หรือลดอีลีเมนต์ต่าง ๆ

3) เหตุการณ์ (Events) ในจาวาสคริปต์

สาหรับจาวาสคริปต์จะพบปัญหาในเรื่องของเหตุการณ์บ่อยครั้ง เนื่องจากความไม่เข้ากันของเว็บเบราว์เซอร์ แต่เมื่อใช้เจคิวรี่ปัญหาที่เกิดขึ้นจะลดน้อยลงหรือหมดไป เพราะเจคิวรี่จะช่วยตรวจสอบชนิดของเว็บเบราเซอร์ให้โดยอัตโนมัติ

4) วิชวลเอฟเฟกต์และแอนิเมชัน (Visual Effects and Animation)

วิชวลเอฟเฟกต์และแอนิเมชันเป็นอีกเทคนิคหนึ่งที่กาลังได้รับความนิยมสาหรับการทาเว็บไซต์ในปัจจุบัน เช่น การทาจางเข้า จางออก เป็นต้น เจคิวรี่ได้เตรียมฟังก์ชันต่าง ๆ เพื่อรองรับการแสดงผลในส่วนนี้

5) เอแจ็กซ์

เอแจ็กซ์ก็เป็นอีกหนึ่งรูปแบบของการเขียนเว็บแอปพลิเคชันในปัจจุบัน โดยเจคิวรี่มีฟังก์ชันสาหรับการเรียกใช้งานเอแจ็กซ์ให้ง่ายขึ้นกว่าเดิม

6) ส่วนประสานงานผู้ใช้ (User Interface)

ส่วนประสานงานผู้ใช้คือส่วนควบคุมต่าง ๆ ที่ควรมีหรืออยากให้มี เช่น เมนู (Menu) ปฏิทิน (Calendar) เป็นต้น ทำให้ผู้ใช้งานใช้งานได้สะดวกมากยิ่งขึ้น

**2.8 ความรู้พื้นฐานเกี่ยวกับเอแจ็กซ์ (Asynchronous JavaScript and XML: Ajax)**

2.8.1 ความหมายของเอแจ็กซ์

เอแจ็กซ์เป็นกลุ่มของเทคนิคในการพัฒนาเว็บแอปพลิเคชันเพื่อให้ความสามารถในด้านการโต้ตอบกับผู้ใช้งานได้ดีขึ้น โดยการรับส่งข้อมูลจะอยู่ในส่วนหลังบ้าน (Backend) ทำให้หน้าเว็บเพจไม่ต้องโหลดใหม่ทั้งหมดทุกครั้งที่มีการเปลี่ยนแปลง ส่งผลให้หน้าเว็บเพจสามารถตอบสนองต่อผู้ใช้งานได้รวดเร็วมากยิ่งขึ้น เอแจ็กซ์ไม่ใช่เทคโนโลยีที่คิดค้นขึ้นมาใหม่ แต่เป็นเทคนิคที่นำเทคโนโลยีหลายอย่างที่มีอยู่แล้วมารวมกันดังต่อไปนี้

1) การแสดงผลลัพธ์และรูปแบบข้อมูล ใช้ภาษาเอชทีเอ็มแอลและซีเอสเอส

2) การแสดงข้อมูลที่การเปลี่ยนแปลงให้โต้ตอบกับผู้ใช้งานทันที โดยใช้อีซีเอ็มเอสคริปต์ (ECMAScript) เช่น ภาษาจาวาสคริปต์ในการเข้าถึง เช่น ดอม (Document Object Model: DOM)

3) การเลือกเปลี่ยนข้อมูลโดยที่ช่วงเวลาไม่สอดคล้องกันกับเซิร์ฟเวอร์ โดยใช้เอ็กเอชอาร์ (XMLHttpRequest: XHR)

4) รูปแบบข้อมูลที่ใช้ในการแลกเปลี่ยน เช่น เอ็กเอ็มแอล เจสัน (JSON) อีบีเอ็มแอล (EBML) หรือ เพลนเท็กซ์

2.8.2 หลักการทางานของเอแจ็กซ์

วิธีการทางานของเว็บแอปพลิเคชันแบบดั้งเดิมนั้น โดยปกติแล้วเมื่อผู้ใช้ทาการร้องขอข้อมูลจากเซิร์ฟเวอร์ เว็บเบราว์เซอร์จะทาการส่งข้อมูลการร้องขอโดยใช้โพรโทคอลเอชทีทีพี (HTTP) เพื่อติดต่อกับเว็บเซิร์ฟเวอร์ และที่เว็บเซิร์ฟเวอร์จะทาการประมวลผลจากการร้องขอที่ได้รับ และส่งผลลัพธ์เป็นหน้าเอชทีเอ็มแอลกลับไปให้ผู้ใช้งาน วิธีการทางานข้างต้นเป็นวิธีการแบบการร้องขอและการตอบรับ (Request and Response) ซึ่งผู้ใช้งานจะต้องรอระหว่างที่เซิร์ฟเวอร์ประมวลผลอยู่ ถือเป็นหลักการการทางานแบบช่วงเวลาตรงกัน (Synchronous) แต่การทางานของเว็บแอปพลิเคชันที่ใช้เทคนิคเอแจ็กซ์จะเป็นการทางานแบบช่วงเวลาไม่ตรงกันหรือการติดต่อสื่อสารแบบไม่ต่อเนื่อง โดยเซิร์ฟเวอร์จะทาการส่งผลลัพธ์ไปยังเว็บเพจให้ผู้ใช้งานทันที โดยไม่ต้องรอให้ประมวลผลเสร็จก่อน เว็บเพจของผู้ใช้งานจะทาการดึงข้อมูลในส่วนต่าง ๆ ทีหลัง



**ภาพที่ 2.2 หลักการทางานของเอแจ็กซ์**

จากภาพที่ 2.2 เปรียบเทียบหลักการทางานระหว่างเว็บแอปพลิเคชันทั่วไปและเว็บแอปพลิเคชันที่ใช้เทคโนโลยีเอแจ็กซ์ จะเห็นได้ว่าการทางานของเทคโนโลยีเอแจ็กซ์ ช่วงเวลาของผู้ใช้ที่ใช้งานไม่สอดคล้องกันกับเซิร์ฟเวอร์