

บทที่ 2

ความรู้พื้นฐานที่ใช้ในการทำเว็บไซต์

2.1) แนะนำระบบอินเทอร์เน็ต (Internet)

อินเทอร์เน็ต เป็นระบบเครือข่ายที่มีเครื่องคอมพิวเตอร์เชื่อมต่อกันมากที่สุดในโลก ภายในระบบเครือข่ายอินเทอร์เน็ตมีบริการต่างๆ เช่น บริการ e-mail หรือจดหมายอิเล็กทรอนิกส์ การให้บริการแบบ Gopher การให้บริการแบบ FTP การให้บริการแบบ WWW และอื่นๆ อีกมาก จุดเด่นที่ทำให้ระบบอินเทอร์เน็ตได้รับความนิยมอย่างมาก คือระบบอินเทอร์เน็ตนี้ใช้โปรโตคอลแบบ TCP/IP ทำให้เครื่องคอมพิวเตอร์ทุกเครื่องมีความสำคัญเท่ากันหมด เมื่อมีเครื่องใดเครื่องหนึ่งในระบบหยุดทำงานแล้ว จะมีเครื่องอื่นๆ ที่มาทำหน้าที่แทน ทำให้ระบบยังสามารถที่จะทำงานต่อไปได้

จุดเริ่มต้นของอินเทอร์เน็ตเกิดเมื่อประมาณปี 1964 หลังสงครามโลกครั้งที่ 2 ขณะนั้นอยู่ในช่วงสงครามเย็นระหว่างสหรัฐอเมริกากับรัสเซีย ทางรัฐบาลสหรัฐต้องการสร้างระบบเครือข่ายแบบใหม่เพื่อป้องกันการจู่โจมของรัสเซีย คือเป็นระบบเครือข่ายที่สามารถทำงานต่อไปได้ถึงแม้ว่าจะมีบางส่วนเสียหาย จากจุดนี้ทำให้มีการพัฒนาโปรโตคอลแบบ TCP/IP ขึ้นมา และเริ่มติดตั้งคอมพิวเตอร์ที่ทำงานแบบนี้เป็นเครื่องแรกในปี 1969 ที่มหาวิทยาลัย UCLA และอีก 3 ที่คือ สถาบันวิจัย Standford Research Institute , มหาวิทยาลัย Utha และมหาวิทยาลัย UCSB รวมเป็น 4 แห่ง เชื่อมโยงกันเป็นระบบเครือข่ายชื่อ ARPANET (Advanced Research Projects Argeny Network) โดยถูกใช้เป็นระบบเครือข่ายทดลองของกระทรวงกลาโหมสหรัฐอเมริกา หลังจากนั้นระบบเครือข่ายนี้ก็เปิดให้องค์กรหรือมหาวิทยาลัยต่างๆ ได้เข้ามามีส่วนร่วมอยู่เรื่อยๆ เมื่อมีผู้เข้าร่วมในเครือข่ายมากขึ้นทุกที นักพัฒนาโปรแกรมได้ช่วยกันเขียนโปรแกรมที่ใช้ในการติดต่อและส่งข้อมูลถึงกัน ตัวอย่างโปรแกรมเหล่านั้นได้แก่ e-mail Gopher WWW เป็นต้น

World Wide Web หรือเรียกย่อๆ ว่า WWW แปลความหมายได้ว่า สายใยกว้างไกลครอบคลุมทั่วโลก เป็นวิธีการเผยแพร่ข้อมูลข่าวสารที่มีขอบเขตครอบคลุมทั่วทั้งโลก ผู้ที่ใช้บริการนี้สามารถที่จะอ่านข้อมูลใดๆ ในโลกของเว็บได้ทั้งหมด ถ้าเปรียบระบบเครือข่ายเว็บเป็นห้องสมุดขนาดใหญ่สักห้อง ภายในมีหนังสือมากมาย ในโลกของเว็บก็จะมี **เว็บไซต์ (Web Site)** ทำหน้าที่เป็นแหล่งข้อมูลเปรียบเหมือนเป็นหนังสือหนึ่งเล่ม ข้อมูลแต่ละหน้าของเว็บไซต์เรียกว่า

เว็บเพจ (Web Page) ในเว็บไซต์ต่างๆ จะมีจำนวนเว็บเพจไม่เท่ากัน ขึ้นอยู่กับข้อมูลของเว็บไซต์นั้นๆ ว่ามีมากน้อยเพียงใด วิธีการเชื่อมโยงเว็บเพจหลายๆ หน้าเข้าด้วยกัน ต้องอาศัยเทคนิคที่เรียกว่า ไฮเปอร์เท็กซ์ (HyperText) คือการใช้ข้อความภายในเว็บเพจหน้าหนึ่งเป็นตัวเชื่อมโยงไปยังเว็บเพจหน้าอื่นๆ ในยุคแรกๆ วิธีการเชื่อมเว็บเพจเข้าด้วยกันยังใช้ข้อความเพียงอย่างเดียว แต่ในปัจจุบันสามารถจัดรูปเป็นตัวเชื่อมโยงได้ และยังมีวิธีการเชื่อมโยงแบบอื่นๆ ที่แบบขลอีกมากมาย การเชื่อมโยงไม่ได้ถูกจำกัดว่าต้องเชื่อมโยงเฉพาะเว็บเพจที่อยู่ในเว็บไซต์เดียวกันเท่านั้น จะเชื่อมข้ามประเทศหรือข้ามทวีปก็ได้ ด้วยวิธีการเชื่อมโยงแบบนี้ ถึงแม้ว่าทั่วโลกจะมีเว็บเพจอยู่หลายร้อยล้านหน้าก็สามารถที่จะเชื่อมโยงถึงกันได้

2.2) ภาษาที่ใช้ในการสร้างเว็บเพจ

2.2.1) ภาษา HTML

ภาษา HTML เป็นภาษาที่ใช้ในการสร้างเว็บเพจ โดยจะทำหน้าที่แปลคำสั่งต่างๆ ที่อยู่ในรูปของเท็กไฟล์ ให้เป็นรหัสแอสกี (ASCII) สาเหตุที่ภาษา HTML ถูกกำหนดให้แปลเป็นรหัสแอสกี เป็นเพราะว่าข้อความที่เป็นรหัสแอสกี เครื่องคอมพิวเตอร์ทุกเครื่องในโลกสามารถอ่านและแปลความหมายได้เหมือนกัน

ภาษา HTML ในเวอร์ชันแรกๆ มีวัตถุประสงค์เพื่อนำเสนอข้อมูลที่เป็นข้อความเป็นส่วนใหญ่ ต่อมาเมื่อความก้าวหน้าทางเทคโนโลยีมีมากขึ้นและราคาอุปกรณ์คอมพิวเตอร์มีราคาถูกลง ทำให้มีการใช้ข้อมูลแบบสื่อหลายทาง หรือที่เรียกทับศัพท์ว่า มัลติมีเดีย (Multimedia) มากขึ้น จนกระทั่งปัจจุบันเป็นส่วนที่ขาดไม่ได้ ดังนั้นภาษา HTML ในเวอร์ชันต่อมาจึงได้มีการเพิ่มชุดคำสั่ง หรือที่เราเรียกว่า แท็ก (Tag) ที่ทำงานกับไฟล์รูปหรือลูกเล่นต่างๆ ที่เกี่ยวกับรูปและข้อมูลมัลติมีเดียประเภทอื่นๆ เพิ่มมากขึ้น ในปัจจุบันภาษา HTML เป็นเวอร์ชัน 4.0

2.2.1.1) ส่วนประกอบของภาษา HTML

จะประกอบด้วย 3 ส่วนใหญ่ๆ ตามข้อกำหนดขององค์กร W3C (World Wide Web Consortium. ซึ่งเป็นองค์กรกลางในการจัดและกำหนดมาตรฐานของภาษา HTML ให้มีมาตรฐานกลางเดียวกันทั่วโลก) ดังนี้

- ส่วนประกาศเวอร์ชันของภาษา HTML

เป็นส่วนที่กำหนดวิธีการแปลข้อมูลต่างๆ ในเอกสาร HTML ทั้งหมด สาเหตุที่ต้องมีส่วนประกาศเวอร์ชันเนื่องมาจากภาษา HTML ในแต่ละเวอร์ชันมีแท็กและแอททริบิวต์ไม่เหมือนกัน คือในเวอร์ชันที่ใหม่กว่าอาจไม่มีแท็กและแอททริบิวต์บางส่วนที่เคยมีอยู่ในเวอร์ชันก่อนหน้า การที่ประกาศเวอร์ชันของภาษา HTML ที่ใช้ในการสร้างเว็บเพจ จะช่วยให้เว็บเบราว์เซอร์สามารถแปล

ความหมายและแสดงเว็บเพจได้อย่างถูกต้อง แต่ในปัจจุบันเว็บเบราว์เซอร์ที่ส่วนใหญ่นิยมใช้กัน ได้แก่ Microsoft Internet Explorer และ Netscape Communicator ในเวอร์ชันปัจจุบัน สามารถแปลความหมายของแท็กที่องค์กร W3C ประกาศยกเลิกการใช้ไปแล้วได้อย่างถูกต้อง ทำให้ส่วนนี้สามารถละเว้นได้ แต่ในอนาคตถ้าเว็บเบราว์เซอร์เวอร์ชันใหม่ๆ ของค่ายต่างๆ สนับสนุนเฉพาะภาษา HTML เวอร์ชันใหม่ การประกาศเวอร์ชันของภาษา HTML ในเอกสาร HTML จะเป็นส่วนจำเป็นที่ขาดไม่ได้ เพื่อให้เว็บเบราว์เซอร์สามารถแปลความหมายและแสดงผลเว็บเพจได้อย่างถูกต้อง การประกาศเวอร์ชันของภาษา HTML ในเวอร์ชัน 4.0 มีอยู่ 3 รูปแบบคือ

HTML 4.0 ชุดเข้มงวด

เป็นข้อกำหนดที่ไม่อนุญาตให้ใช้แท็กและแอททริบิวต์ที่กำลังจะถูกยกเลิกในอนาคต เช่น แท็ก <MENU> , <ISINDEX> ,แท็กที่เกี่ยวกับเฟรมทั้งหมด เราจะกำหนดที่บรรทัดแรกของเอกสาร HTML ดังนี้

```
<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
```

```
"http://www.w3.org/TR/REC-html40/Strict.dtd">
```

HTML 4.0 ชุดยืดหยุ่น

เป็นข้อกำหนดชุดเข้มงวดทั้งหมดและอนุญาตให้ใช้แท็กและแอททริบิวต์ที่ถูกประกาศว่าจะยกเลิกในอนาคตได้ เราจะทำการกำหนดที่บรรทัดแรกของเอกสาร HTML ดังนี้

```
<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
```

```
"http://www.w3.org/TR/REC-html40/Loose.dtd">
```

HTML 4.0 ชุดใช้เฟรม

เป็นข้อกำหนดชุดยืดหยุ่นทั้งหมดและเพิ่มการอนุญาตให้สามารถให้แท็กต่างๆที่เกี่ยวกับเฟรมทั้งหมด เราจะกำหนดที่บรรทัดแรกของเอกสาร HTML ดังนี้

```
<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
```

```
"http://www.w3.org/TR/REC-html40/Frameset.dtd">
```

ส่วนหัวเอกสาร

คือข้อความที่อยู่ภายใต้แท็ก <head> และ </head> ทั้งหมด ส่วนมากข้อความที่อยู่ภายใต้ส่วนหัวเอกสารจะเป็นข้อกำหนดต่างๆ ที่เกี่ยวกับเอกสาร HTML นั้นและการอ้างอิงถึงเอกสารอื่นๆ

ที่เกี่ยวข้องกับเอกสาร HTML นี้ แท็กที่สามารถใช้ในส่วนหัวเอกสารได้ มีดังนี้

แท็ก <TITLE>

ในภาษา HTML เวอร์ชัน 4.0 กำหนดว่าต้องมีแท็ก <TITLE> เพื่อกำหนดชื่อเรื่องของเว็บเพจนั้นๆ ข้อความที่ใช้เป็นชื่อเรื่องของเว็บเพจจะถูกโปรแกรม Search Engine นำไปใช้ทำสารบัญเว็บเพจ เพื่อให้บุคคลที่สนใจค้นพบข้อมูลได้โดยง่าย จึงจำเป็นต้องกำหนดชื่อเรื่องของเว็บเพจให้ตรงกับจุดประสงค์ของเว็บเพจหน้านั้นๆ โปรแกรม Search Engine จะได้จัดเว็บเพจหน้านั้นๆ ไปอยู่ในกลุ่มที่ถูกต้อง

รูปแบบ

<TITLE> ชื่อเรื่องของเว็บเพจ </TITLE>

แท็ก <META>

ตามข้อกำหนดมาตรฐานของภาษา HTML ไม่ได้กำหนดหน้าที่ของแท็ก <META> ไว้ตายตัว เพียงแต่กำหนดให้แท็ก <META> เป็นแท็กสำหรับใช้งานที่แท็กอื่นไม่สามารถทำได้ ด้วยเหตุนี้แท็กนี้จึงสามารถทำหน้าที่ได้หลายหน้าที่ ขึ้นอยู่กับว่าจะใช้แอททริบิวต์ตัวใดและกำหนดค่าต่างๆ ให้กับแอททริบิวต์แต่ละตัวอย่างไร แท็ก <META> มีแอททริบิวต์ 4 ตัว ดังนี้

HTTP-EQUIV

ใช้กำหนดข้อมูลที่จะนำไปสร้าง HTTP header ซึ่งเป็นข้อมูลสำคัญที่ใช้ในการรับส่งข้อมูลโดยใช้โปรโตคอล HTTP

NAME

ใช้ระบุชื่อคุณสมบัติของเอกสาร เช่น author , expiration , date และ keyword (คำสำคัญที่โปรแกรม Search Engine ใช้ในการตรวจสอบว่าเว็บเพจตรงกับความต้องการของผู้ค้นหาหรือไม่) เป็นต้น ซึ่งจะต้องส่งไปพร้อมกับค่าของคุณสมบัติที่ระบุไว้ในแอททริบิวต์ CONTENT

CONTENT

ใช้กำหนดค่าคุณสมบัติของเอกสารที่ต้องการส่งให้เว็บเซิร์ฟเวอร์หรือกำหนดค่าให้ชื่อเฉพาะที่กำหนดด้วยแอททริบิวต์ NAME

SCHEME

ใช้กำหนดรูปแบบในการตีความข้อมูลที่อยู่ในแอททริบิวต์ CONTENT เช่น <META SCHEME="9-digit-Zipcode" NAME="zip" CONTENT="02134-1078">

การใช้งานแท็ก <META> มีได้หลายแนวทาง แนวทางต่อไปนี้จะเป็นแนวทางที่นิยมใช้เป็นสากล

1. ใช้แท็ก <META> ในการกำหนดค่า Header Type

Header Type คือข้อมูลส่วนที่เว็บเซิร์ฟเวอร์จะส่งให้เว็บเบราว์เซอร์ทันทีเมื่อได้รับสัญญาณการร้องขอข้อมูลจากเว็บเบราว์เซอร์ ข้อมูลในส่วน Header Type นี้จะทำหน้าที่บอกเว็บเบราว์เซอร์ว่าเอกสาร HTML ที่กำลังดาวน์โหลดมานั้นมีคุณสมบัติอะไรบ้าง หลังจากนั้นเว็บเซิร์ฟเวอร์จึงจะส่งข้อมูลในเอกสาร HTML ไปให้เว็บเบราว์เซอร์ต่อไป

การกำหนด Header Type สามารถทำได้ 2 วิธี คือ

- ใช้วิธีการเขียนข้อความไว้ก่อนแท็ก <HTML>

```
Content-Type:text/html;charset=Windows-874
```

```
<HTML>
```

- ใช้แท็ก <META> เป็นตัวกำหนด

```
<HEAD>
```

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=Windows-874">
```

```
</HEAD>
```

2. ใช้แท็ก <META> เพื่อกำหนดคำสำคัญประจำเว็บเพจ เพื่อให้โปรแกรม Search Engine ค้นหาได้

การกำหนดคำสำคัญประจำเว็บเพจมักกำหนดด้วยแอททริบิวต์ NAME="KEYWORDS" หรือ NAME="DESCRIPTION" ตามตัวอย่างต่อไปนี้

```
<HEAD>
```

```
<TITLE> Computer Engineering Syllabus </TITLE>
```

```
<META NAME="KEYWORDS" CONTENT="Syllabus,Computer,Engineering">
```

```
<META NAME="DESCRIPTION" CONTENT="This is page of Computer Engineering Syllabus of Naresuan University">
```

```
</HEAD>
```

3. ใช้แท็ก <META> เพื่อกำหนดให้เว็บเบราว์เซอร์ทำการดึงข้อมูลจากเว็บเซิร์ฟเวอร์ เมื่อถึงเวลาที่กำหนดโดยอัตโนมัติ

เป็นการกำหนดให้เว็บเบราว์เซอร์ทำการดาวน์โหลดเอกสาร HTML จากเว็บเซิร์ฟเวอร์โดยไม่ต้องรอให้ผู้ป้อน URL ดังตัวอย่างต่อไปนี้

<HEAD>

<META HTTP-EQUIV="Refresh" CONTENT="5;

URL=http://www.microsoft.com/index.html">

</HEAD>

แท็ก <LINK>

แท็กนี้ใช้ในการอ้างอิงข้อมูลอื่นๆ ที่เกี่ยวข้องกับเอกสาร HTML ตัวปัจจุบัน ซึ่งการอ้างอิงเอกสารตัวอื่นนั้นมีจุดประสงค์อยู่ 2 ประการ คือ

1. อ้างอิงเอกสารข้อมูลตัวอื่น เพื่อกำหนดให้เว็บเบราว์เซอร์ทำการดึงข้อมูลในเอกสารนั้นมาใช้ประกอบกับข้อมูลในเอกสาร HTML ตัวปัจจุบัน เพื่อแสดงเป็นเว็บเพจ เช่น การอ้างอิงไฟล์ที่มีนามสกุล .css เพื่อนำข้อกำหนดต่างๆ ในไฟล์นั้นมาใช้ในการควบคุมการแสดงผลของเว็บเพจ
2. อ้างอิงเอกสารข้อมูลตัวอื่น เพื่อเป็นข้อมูลเพิ่มเติมแสดงความสัมพันธ์ของเอกสาร HTML ตัวปัจจุบันกับเอกสารที่อ้างอิง

- ส่วนตัวเอกสาร

เป็นส่วนที่เราต้องการแสดงในเว็บเบราว์เซอร์ซึ่งจะถูกอยู่ระหว่างแท็ก <BODY> และ </BODY> หรือในกรณีที่ใช้เฟรมส่วนตัวเอกสารก็จะอยู่ระหว่างแท็ก <FRAMESET> และ </FRAMESET>

ในเอกสาร HTML หนึ่งๆจะต้องมีส่วนประกอบอย่างน้อย 2 ส่วน คือ ส่วนหัวเอกสาร กับ ส่วนตัวเอกสาร

ตัวอย่าง

<DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.0 Frameset//En"

"http://www.w3.org/TR/REC-html40/frameset.dtd">

<html>

<head>

<link rel="stylesheet" href="Main.css" type="text/css">

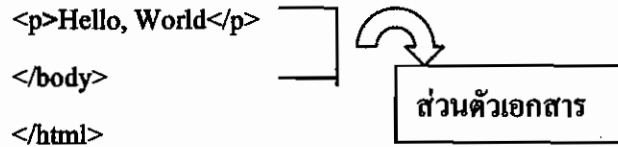
<title>หน้าแรกของโฮมเพจ</title>

</head>

<body>

ส่วนประกาศเวอร์ชัน

ส่วนหัวเอกสาร



2.2.1.2) การควบคุมการแสดงผลในเว็บเบราว์เซอร์ของเอกสาร HTML

2.2.1.2.1) การควบคุมการแสดงผลโดยใช้แท็กต่างๆ

ในภาษา HTML มีแท็กต่างๆ อยู่มากมาย เราสามารถใช้แท็กเหล่านี้ในการควบคุมรูปร่างหน้าตาของเว็บเพจที่จะปรากฏบนเว็บเบราว์เซอร์ การเลือกใช้แท็กขึ้นอยู่กับจุดประสงค์ของผู้ทำเว็บเพจว่าต้องการให้ผลลัพธ์ปรากฏที่เว็บเบราว์เซอร์อย่างไร แท็กที่ใช้ในการควบคุมการแสดงผลของเว็บเพจมีดังต่อไปนี้

- แท็กที่ใช้ในการจัดตำแหน่งของข้อมูลบนหน้าเว็บเพจ

แท็ก <DIV>

สำหรับแท็ก <DIV> นี้มักใช้ในการจัดกลุ่มของข้อความหรือรูปภาพให้อยู่ในตำแหน่งที่ต้องการ โดยการกำหนดค่าแอททริบิวต์ align ให้เป็น left, right หรือ center เมื่อเว็บเบราว์เซอร์พบแท็กนี้ก็จะทำการขึ้นย่อหน้าใหม่ให้ทันที

รูปแบบ

```
<DIV align="center">
```

ข้อความ , รูปภาพหรือตาราง

```
</DIV>
```

แท็ก <CENTER>

ใช้ในการจัดตำแหน่งของข้อความหรือรูปภาพให้อยู่ตรงกลางหน้าจอเบราว์เซอร์ แท็กนี้จะทำงานคล้ายกับแท็ก <DIV> แต่ในปัจจุบันแท็กนี้ไม่นิยมใช้เท่าไรเพราะว่าแท็ก <DIV> สามารถทำงานแทนได้หมดและสามารถจัดได้หลากหลายรูปแบบมากกว่า และในอนาคตทางองค์กร W3C จะประกาศยกเลิกใช้แท็กนี้

รูปแบบ

```
<CENTER>
```

ข้อความ , รูปภาพหรือตาราง

```
</CENTER>
```

**แท็ก **

เป็นแท็กที่ใช้ในการจัดแต่งข้อความสั้นๆ ที่อยู่ใต้การกำหนดรูปแบบของแท็กอื่นอยู่แล้ว เพื่อให้มีลักษณะที่แตกต่างจากรูปแบบเหล่านั้น โดยการกำหนดค่าแอททริบิวต์ style

รูปแบบ

 ข้อความ

เช่น Hello

- แท็กที่ใช้ในการสร้างตาราง

ในการทำเว็บเพจ เราไม่สามารถจัดข้อความต่างๆ ในเว็บเพจได้ด้วยการเว้นวรรคเหมือนในโปรแกรมเวิร์ด การจัดข้อความด้วยตารางเป็นอีกวิธีหนึ่งซึ่งช่วยให้การจัดเรียงข้อความหรือรูปภาพต่างๆ เป็นไปตามต้องการ และแสดงผลออกมาได้สวยงามเป็นระเบียบเมื่อดูด้วยเว็บเบราว์เซอร์

รูปแบบ

<TABLE>

<TR><TD> ข้อความหรือรูปที่จะใส่ในตาราง </TD></TR>

</TABLE>

แอททริบิวต์

ALIGN : ใช้กำหนดตำแหน่งของตารางบนเว็บเพจ มีค่าที่เป็นไปได้ 3 ค่า คือ left, right , center

BORDER : กำหนดความหนาของเส้นกรอบตาราง มีค่าเป็นหน่วยพิกเซล ถ้ากำหนดให้มีค่าเป็น 0 คือไม่แสดงเส้นกรอบของตาราง

BGCOLOR : กำหนดสีพื้นหลังของตาราง สามารถกำหนดค่าได้ 2 ลักษณะคือ เป็นชื่อสี หรือเป็นค่ารหัสสี RGB

CELLPADDING : กำหนดระยะห่างระหว่างเส้นกรอบของตารางกับข้อมูลที่อยู่ในแต่ละเซลล์ กำหนดค่าเป็นหน่วยพิกเซล

CELLSPACING : กำหนดค่าระยะห่างระหว่างเซลล์ข้อมูล กำหนดเป็นหน่วยพิกเซล

COLS : กำหนดจำนวนคอลัมน์ในแต่ละแถวของตาราง

WIDTH : กำหนดความกว้างของตาราง มีหน่วยเป็นพิกเซล

FRAME : ใช้กำหนดว่าต้องการแสดงเส้นกรอบนอกของตารางด้านใดบ้าง

Above ใส่เส้นกรอบด้านบน

Below ใส่เส้นกรอบด้านล่าง

Border ใส่เส้นกรอบทั้งสี่ด้าน
Box เหมือนกับ Border
Hsides ใส่เส้นกรอบเฉพาะด้านบนและด้านล่าง
Lhs ใส่เส้นกรอบเฉพาะด้านซ้าย
Rhs ใส่เส้นกรอบเฉพาะด้านขวา
Vsides ใส่เส้นกรอบเฉพาะด้านซ้ายและด้านขวา
Void ไม่ใส่เส้นกรอบด้านใดเลย

RULES : กำหนดว่าต้องการแสดงเส้นกรอบกับส่วนใดของตาราง

All ใส่เส้นกรอบทุกช่องในตาราง
Cols ใส่เส้นคั่นระหว่างคอลัมน์
Groups ใส่เส้นกรอบรอบกลุ่มของช่องในตารางที่กำหนดด้วยแท็กต่างๆ เช่น <THEAD> ,<TBODY> ,<TFOOT> หรือ<COLGROUPS>
none ไม่แสดงกรอบที่ใดเลย
rows ใส่เส้นคั่นระหว่างแถว

ตัวอย่าง

```

<TABLE border="1" cols="4" bgcolor="white">
<TR>
  <TD> HELLO </TD>
  <TD> SUNNY </TD>
  <TD> &nbsp; </TD>
  <TD> DOGS </TD>
</TR>
</TABLE>
  
```

ผลที่แสดงออกมาเมื่อดูด้วยเว็บเบราว์เซอร์ จะเป็นดังนี้

HELLO	SUNNY		DOGS
-------	-------	--	------

รูปที่ 2.1 การสร้างตาราง

การกำหนดค่าแอททริบิวต์ต่างๆ ไม่จำเป็นว่าเราต้องยกมาใช้ทั้งหมด จากตัวอย่างเรยกแอททริบิวต์มาใช้แค่เพียงบางตัวเท่านั้น และแอททริบิวต์บางค่าจะมีค่าที่เป็นค่า default ถึงแม้ว่าจะไม่ได้กำหนดค่าแอททริบิวต์ลงไปก็แท็ก เช่น FRAME จะมีค่า default เป็น border

- แท็กที่ใช้ในการสร้างเฟรม

ในบางครั้งที่เราต้องการให้เว็บเพจหน้าหนึ่งๆ แสดงเอกสาร HTML มากกว่า 1 ไฟล์ เราจะใช้แท็กที่เกี่ยวกับเฟรมเหล่านี้แบ่งหน้าเว็บเพจออกเป็นส่วน โดยที่แต่ละส่วนก็จะแสดงเอกสาร HTML ที่ต่างกัน แท็กที่เกี่ยวกับเฟรมนี้มีประโยชน์มากในกรณีที่มีเอกสาร HTML ที่เป็นเมนูต่างๆ ซึ่งเอกสาร HTML หน้าอื่นๆ ต้องมีเอกสาร HTML ตัวนี้ประกอบอยู่ด้วยเสมอ เราจะใช้เฟรมหนึ่งทำหน้าที่แสดงเอกสาร HTML ที่เป็นเมนู และให้อีกเฟรมหนึ่งแสดงเอกสาร HTML ตัวอื่นๆ เวลาที่เราต้องการดูเอกสาร HTML ตัวอื่นก็เพียงแค่โหลดเอกสารตัวนั้นมาโดยที่เราไม่ต้องโหลดเอกสาร HTML ที่เป็นเมนูนั้นใหม่ ทำให้เวลาที่ใช้ในการโหลดลดลง

รูปแบบ

<FRAMESET>

<FRAME>

</FRAMESET>

แอททริบิวต์

ROWS : ใช้กำหนดว่าจะให้มีการแบ่งพื้นที่ของหน้าเว็บเพจออกเป็นกี่แถว และในแต่ละแถวมีความสูงเท่าใด

COLS : ใช้กำหนดว่าจะให้มีการแบ่งพื้นที่ของหน้าเว็บเพจออกเป็นกี่คอลัมน์ และในแต่ละคอลัมน์มีความกว้างเท่าใด

การใช้แอททริบิวต์ทั้งสองตัวนี้มีข้อห้ามคือ ห้ามใช้ด้วยกัน ให้เลือกใช้ตัวใดตัวหนึ่งเท่านั้นในแต่ละแท็ก <FRAMESET>

แอททริบิวต์

FRAMEBORDER : ใช้กำหนดว่าจะแสดงขอบเฟรมหรือไม่ ค่า 0 คือการไม่แสดงขอบ ค่า 1 คือแสดงขอบเฟรม

MARGINHEIGHT : ใช้กำหนดระยะห่างระหว่างเส้นกรอบด้านบนของเฟรมกับตัวข้อมูล กำหนดค่าเป็นหน่วยพิกเซล

MARGINWIDTH : ใช้กำหนดระยะห่างระหว่างเส้นกรอบด้านซ้ายของเฟรมกับตัวข้อมูล กำหนดเป็นหน่วยพิกเซล

NAME : ใช้กำหนดชื่อให้กับเฟรมนั้นๆ เพื่อให้แท็กอื่นๆ สามารถอ้างถึงได้

NORESIZE : ใช้กำหนดว่าเมื่อเวลาผู้ใช้งานคลิกเมาส์ ผู้ใช้ไม่สามารถทำการย่อหรือขยายเฟรมได้ ซึ่งถ้าไม่กำหนดผู้ใช้สามารถใช้เมาส์ย่อหรือขยายขนาดของเฟรมได้

SCROLLING : กำหนดว่าต้องการแสดงแถบเลื่อนด้านข้างของเฟรมหรือไม่

Yes แสดงแถบเลื่อนเสมอ

No ไม่แสดงแถบเลื่อน

Auto แสดงแถบเลื่อนเมื่อข้อมูลล้นเฟรม

SRC : ใช้กำหนดชื่อเอกสาร HTML ที่ต้องการนำมาแสดงในเฟรมนั้นๆ เป็นแอททริบิวต์ที่ต้องมีเสมอ

แท็ก <NOFRAMES>

ใช้ในกรณีที่เว็บเบราว์เซอร์ของผู้ใช้ไม่สามารถแสดงเฟรมได้ โดยอาจให้แสดงเอกสาร HTML ที่ไม่มีการใช้เฟรมหรือแสดงข้อความบอกกับผู้ใช้ว่าเว็บเบราว์เซอร์ของผู้ใช้ไม่สามารถแสดงเฟรมได้

รูปแบบ

<NOFRAME> ข้อความหรือเอกสาร HTML ในแบบที่ไม่มีเฟรม </FRAME>

แท็ก <IFRAME>

ใช้ในการกำหนดเฟรมขึ้นมาให้อยู่กลางหน้าเว็บ เฟรมที่สร้างขึ้นมานี้จะมีลักษณะเหมือนรูปภาพสี่เหลี่ยมที่มีสกอร์บาร์อยู่ด้านข้าง สามารถวางที่ส่วนใดก็ได้ในหน้าเว็บเพจ

แอททริบิวต์

ALIGN : ใช้กำหนดการจัดวางของข้อความที่ต่อท้ายเฟรม

top ข้อความที่ต่อท้ายเฟรมจะจัดวางในระดับเดียวกับขอบด้านบนของเฟรม เมื่อข้อความยาวล้นจอ จะปัดมาขึ้นบรรทัดใหม่ได้เฟรม

middle ข้อความที่ต่อท้ายเฟรมจะจัดวางในระดับตรงกลางของเฟรม เมื่อข้อความยาวล้นจอ จะปัดมาขึ้นบรรทัดใหม่ได้เฟรม

bottom ข้อความที่ต่อท้ายเฟรมจะจัดวางในระดับเดียวกันกับขอบด้านล่างของเฟรม เมื่อข้อความยาวสั้นจอ จะปีดมาขึ้นบรรทัดใหม่ได้เฟรม

left ข้อความจะถูกจัดวางไว้ทางด้านซ้ายของเฟรม

right ข้อความจะถูกจัดวางไว้ทางด้านขวาของเฟรม

FRAMEBORDER : ใช้กำหนดว่าต้องการแสดงขอบของเฟรมหรือไม่

MARGINWIDTH : ใช้กำหนดระยะห่างระหว่างเส้นกรอบด้านซ้ายของเฟรมกับข้อมูล

MARGINHEIGHT : ใช้กำหนดระยะห่างระหว่างเส้นกรอบด้านบนของเฟรมกับข้อมูล

NAME : กำหนดชื่อให้เฟรม เพื่อใช้ในการอ้างอิง

SCROLLING : กำหนดว่าต้องการแสดงแถบเลื่อนหรือไม่

SRC : ใช้กำหนดไฟล์ข้อมูลที่ต้องการนำมาแสดงในเฟรม

WIDTH : กำหนดความกว้างของเฟรม เป็นหน่วยพิกเซล

HEIGHT : กำหนดความสูงของเฟรม เป็นหน่วยพิกเซล

ตัวอย่าง

```
<FRAMESET cols="150,*">
  <FRAME src="test.html" scrolling="auto" frameborder="0" name="left">
  <FRAME src="Hello.html" scrolling="auto" frameborder="0" name="right">
</FRAMESET>
<NOFRAMES >
<P> You browser can't show frame </P>
</NOFRAMES>
```

- แท็กที่ใช้ในการสร้างฟอร์ม

การสร้างฟอร์มเป็นวิธีที่ทำให้ผู้ใช้สามารถส่งข้อมูลต่างๆ ผ่านระบบอินเทอร์เน็ตไปยังผู้ที่ทำเว็บไซต์ได้ ไม่ว่าข้อมูลนั้นจะเป็นคำแนะนำที่มีต่อเว็บไซต์หรือเป็นข้อมูลของผู้ใช้เอง ในปัจจุบันเว็บไซต์ส่วนใหญ่จะมีฟอร์มไว้รับคำถามและข้อคิดเห็นต่างๆ จากผู้ใช้ การใช้ฟอร์มนี้ยังช่วยให้ผู้ใช้ได้มีส่วนร่วมในการใช้บริการเว็บไซต์มากขึ้นด้วย

แท็ก <FORM>

ใช้ครอบข้อมูลและแท็กที่ใช้สร้างฟิลด์รับข้อมูลและปุ่มต่างๆ ที่ประกอบกันเป็นฟอร์ม นอกจากนี้แท็กนี้ยังใช้เป็นตัวกำหนดว่าจะส่งข้อมูลให้กับเว็บเซิร์ฟเวอร์อย่างไร และเรียกใช้โปรแกรมตัวใดทำงาน

แอททริบิวต์

- ACCEPT-CHARSET** : กำหนดชุดของตัวอักษรที่ฟอร์มสามารถรับได้
- ACTION** : กำหนดชื่อไฟล์หรือโปรแกรมที่ทำหน้าที่ประมวลผลข้อมูลที่ผู้เล่นเว็บป้อนเข้ามา โดยที่โปรแกรมหรือไฟล์นั้นอยู่ที่ฝั่งเซิร์ฟเวอร์ และต้องกำหนดชื่อไฟล์ตามแบบ URL
- ENCTYPE** : กำหนดชนิดของไฟล์ที่จะถูกรับและส่งในแบบฟอร์มตามแบบ MIME
- METHOD** : กำหนดวิธีส่งไฟล์ให้เซิร์ฟเวอร์ มีได้ 2 แบบ คือ post และ get
- TARGET** : กำหนดชื่อเฟรมที่ต้องการแสดงผลลัพธ์ที่ได้จากการประมวลผลของเซิร์ฟเวอร์

แท็กที่สำคัญในการสร้างฟิลด์รับและส่งข้อมูลบนฟอร์ม

แท็ก <INPUT>

ใช้สร้างฟิลด์รับข้อมูลแบบต่างๆ ตามค่าที่กำหนดให้กับแอททริบิวต์ type

แอททริบิวต์

- NAME** : ใช้กำหนดชื่อของฟิลด์รับข้อมูล
- VALUE** : ใช้กำหนดค่าเริ่มต้นให้กับฟิลด์รับข้อมูล หรือเป็นข้อความบนปุ่มต่างๆ
- TYPE** : ใช้กำหนดชนิดของฟิลด์ มีค่าที่เป็นไปได้ดังนี้
- Text** : ฟิลด์รับข้อความแบบบรรทัดเดียว
 - Password** : ฟิลด์รับข้อมูลแบบบรรทัดเดียวคล้ายกับ text แต่ใช้กับการป้อนรหัสผ่าน โดยจะแสดง * แทนตัวอักษร
 - Hidden** : ฟิลด์นี้จะไม่แสดงบนเว็บเพจ แต่ข้อมูลที่เกี่ยวข้องภายในฟิลด์จะถูกส่งไปให้เว็บเซิร์ฟเวอร์ด้วย
 - Checkbox** : ฟิลด์ที่ให้ผู้ใช้งานทำการเลือกหรือไม่เลือกกรายการนั้น
 - Radio** : ฟิลด์ชนิดนี้จะมีหลายตัว แต่ผู้ใช้งานสามารถเลือกได้เพียงอันเดียว
 - File** : ฟิลด์ที่ใช้ใส่ชื่อไฟล์เพื่อส่งไปยังเว็บเซิร์ฟเวอร์พร้อมกับข้อมูลที่ผู้ใช้กรอกผ่านฟอร์ม
 - Image** : ใช้กำหนดรูปแทนปุ่มรับคำสั่ง

Button : ใช้สร้างปุ่มรับคำสั่ง เมื่อผู้ใช้ทำการคลิกที่ปุ่ม แอททริบิวต์ onclick จะเรียกสคริปต์หรือ โปรแกรมฝั่งเซิร์ฟเวอร์ทำงาน

Submit : ใช้สร้างปุ่มรับคำสั่ง submit เมื่อผู้ใช้คลิกที่ปุ่มนี้ เว็บบราวเซอร์ จะส่งข้อมูลทั้งหมดที่อยู่ในฟอร์ม ไปให้เซิร์ฟเวอร์

Reset : ใช้สร้างปุ่มรับคำสั่ง reset เมื่อผู้ใช้คลิกที่ปุ่มนี้ เว็บบราวเซอร์ จะล้างข้อมูลทั้งหมดที่อยู่ในฟอร์ม

DISABLE : ใช้ระงับการทำงานของฟิลด์ทุกชนิด โดยที่ผู้ใช้ไม่สามารถทำการแก้ไข ข้อมูลที่อยู่บนฟิลด์เหล่านั้นได้ โดยที่ฟิลด์เหล่านั้นยังคงปรากฏอยู่บน ฟอร์ม

LENGTH : ใช้กำหนดความยาวเป็นจำนวนตัวอักษรให้กับฟิลด์ข้อมูล

MAXLENGTH : กำหนดความยาวสูงสุดเป็นจำนวนตัวอักษรที่ฟิลด์นั้นจะรับได้

READONLY : กำหนดให้ฟิลด์นี้แสดงข้อมูลเพียงอย่างเดียว ไม่สามารถเพิ่มหรือแก้ไข ข้อมูลได้

แท็ก <TEXTAREA>

ใช้สร้างฟิลด์รับข้อมูลแบบหลายบรรทัด

แอททริบิวต์

COLS : กำหนดจำนวนตัวอักษรที่ผู้ใช้สามารถป้อนได้ในแต่ละบรรทัด

DISABLE : ห้ามไม่ให้ผู้ใช้ป้อนข้อมูลในฟิลด์นี้

NAME : กำหนดชื่อให้กับฟิลด์รับข้อมูล

READONLY : กำหนดให้ฟิลด์นี้แสดงข้อมูลเพียงอย่างเดียว

ROWS : กำหนดจำนวนบรรทัดที่ป้อนข้อความ

แท็ก <LABEL>

ใช้ใส่ข้อความกำกับฟิลด์รับข้อมูลต่างๆ

แอททริบิวต์

ACCESSKEY : กำหนดปุ่ม ในแป้นพิมพ์เพื่อให้เคอร์เซอร์เลื่อนไปยังฟิลด์หรือปุ่มภายใน ฟอร์มตามต้องการ เช่น ACCESSKEY="A" ผู้ใช้สามารถกด Ctrl+A หรือ Alt+A เพื่อให้เคอร์เซอร์เลื่อนไปยังฟิลด์หรือปุ่มที่ต้องการ

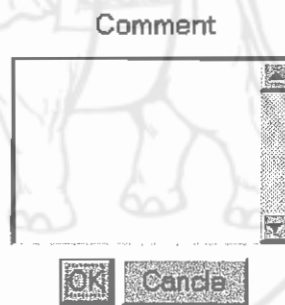
DISABLE : ห้ามไม่ให้ผู้ใช้คลิกข้อความ LABEL นี้ได้

FOR : กำหนดค่าให้เป็น ID ของฟิลด์ที่ต้องการให้คู่กับ LABEL นี้

ตัวอย่าง

```
<FORM action="test.asp" method="post" name="frmName">
<DIV align="center">
<LABEL> Comment </LABEL><BR>
<TEXTAREA name="comment" cols="30" rows="5" ></TEXTAREA><BR>
<INPUT type="submit" value="OK">
<INPUT type="reset" value="Cancel">
</DIV>
</FORM>
```

ผลที่ปรากฏบนเว็บเบราว์เซอร์จะเป็นดังรูป



รูปที่ 2.2 ฟอรั่ม

2.2.1.2.2) สไตลชีท (Style Sheet)

สไตลชีท คือ เอกสารที่เก็บรวบรวมคุณลักษณะของแท็กต่างๆในเว็บเพจ ซึ่งเว็บเพจหน้าอื่นๆ สามารถที่จะใช้ข้อกำหนดสไตลชีทนี้ร่วมกันได้ การแก้ไขหรือเพิ่มเติมคุณลักษณะก็สามารถทำได้อย่างสะดวก โดยการแก้ไขที่เอกสารสไตลชีทเพียงทีเดียว องค์กร W3C ได้สนับสนุนให้นักเขียนเว็บเพจหันมาใช้สไตลชีทในการเขียนเว็บเพจ เพราะนอกจากสามารถกำหนดคุณลักษณะต่างๆ ได้อย่างครอบคลุมแล้ว ยังสะดวกต่อการดูแลหรือปรับปรุงเพิ่มเติมในภายหลัง อีกทั้งยังช่วยลดความซ้ำซ้อนในการเขียนเว็บเพจจากเดิมที่ต้องกำหนดคุณสมบัติของแท็กต่างๆ ทุกครั้ง ก็เปลี่ยนมาเป็นกำหนดไว้ในสไตลชีทเพียงครั้งเดียว แล้วใช้ได้ตลอดเอกสาร HTML

รูปแบบคำสั่ง

```
<STYLE type="text/css">
```

```
<!--
```

ข้อกำหนดคุณลักษณะ

```
-->
```

```
</STYLE>
```

วิธีการเรียกใช้สไตลชีท

1.) **Embedding** คือ การกำหนดสไตลชีทไว้ในเอกสาร HTML โดยกำหนดสไตลชีทอยู่ระหว่างแท็ก <HEAD> และ </HEAD>

ตัวอย่าง

```
<html>
<head><title>Embedding Style Sheet</title>
<style type="text/css">
<!--
    p {font-family:"Ms Sans Serif"; font-size:20pt; font-weight:bold;
        text-align:center}
-->
</style>
</head>
<body><p>Hello, This is test page of Embedding Style Sheet</p></body>
</html>
```

เมื่อดูผลที่บราวเซอร์จะพบว่า ข้อความ "Hello, This is test page of Embedding Syle Sheet" ที่มีขนาด 20 พอยน์ เป็นตัวอักษรในตระกูล Ms Sans Serif ตัวหนา และจัดแสดงผลที่กลางหน้าเว็บบราวเซอร์

2.) **Linking** เป็นการกำหนดสไตลชีทแยกไว้จากเอกสาร HTML ต่างหาก ส่วนในเอกสาร HTML ใช้แท็ก <link> เพื่ออ้างถึงที่เก็บเอกสารที่เป็นสไตลชีท

ตัวอย่าง /* Test.css*/

```
<style type="text/css">
```



```

<!--
      p {font-family:"Ms Sans Serif"; font-size:20pt; font-weight:bold;
          text-align:center}
-->
</style>

```

```

/*Test.html*/
<html>
<head><title>Linking Style Sheet</title>
<link rel="stylesheet" href="c:\My Document\test.css" type="text/css">
</head>
<body><p>Hello, This is test page of Linking Style Sheet</p></body>
</html>

```

ผลลัพธ์ที่ได้จะเหมือนกับวิธี Embedding ทุกประการ โดยที่แท็ก <link> ต้องทำการใส่แอททริบิวต์ให้ถูกต้อง คือ rel="stylesheet" เป็นการบอกแก่เว็บเบราว์เซอร์ว่า เอกสารที่จะอ้างอิงเป็นสไตลชีท href="c:\My Document\test.css" เป็นการบอกตำแหน่งที่อยู่ของสไตลชีท ถ้าอยู่ในโฟลเดอร์เดียวกันกับเอกสาร HTML ก็ใส่เฉพาะชื่อ type="text/css" เป็นการบอกว่าสไตลชีทที่ใช้ เป็นไฟล์ประเภทข้อความที่มีนามสกุลเป็น .css

3.) **Import** เป็นการกำหนดสไตลชีทแยกจากเอกสาร HTML แล้วดึงข้อกำหนดดังกล่าวมาใช้ในเอกสาร HTML ด้วยแท็ก @import ซึ่งวิธีนี้สามารถนำข้อกำหนดสไตลชีทมากกว่า 1 เอกสารมาใช้พร้อมๆกันได้ ถ้ามีข้อกำหนดเกี่ยวกับแท็กซ้ำกัน จะถือข้อกำหนดเกี่ยวกับแท็กนั้นๆ ตามสไตลชีทที่นำเข้ามาหลังสุด

ตัวอย่าง

```

<html>
<head><title>Import Style Sheet</title>
<style type="text/css">
@import url(c:\My Document\test.css)
</style>
<body><p>Hello, This is test page of Import Style Sheet</p></body>

```

</html>

ผลที่ได้จะเหมือนกับสองแบบข้างต้น

4.) **Inline** ข้อกำหนดนี้จะมีผลเฉพาะในแท็กนั้นเท่านั้น ส่วนแท็กอื่นๆแม้ว่าจะเป็นแท็กเดียวกันก็จะมีผล การกำหนดในลักษณะนี้ถึงแม้ว่าในเอกสาร HTML นั้นมีการกำหนดสไตล์ชีทแบบอื่นอยู่แล้ว เว็บเบราว์เซอร์ก็จะแสดงผลตามคุณสมบัติที่กำหนดแบบ Inline ดังนั้นการกำหนดคุณสมบัติแบบนี้ จึงมักใช้เมื่อต้องการกำหนดคุณสมบัติของแท็กบางจุดเพื่อให้เห็นข้อมูลที่แตกต่างจากที่กำหนดในสไตล์ชีท

ตัวอย่าง

```
<html>
<head>
<title>Test Page</title>
</head>
<body>
<p style="color:Blue">This text is red</p>
<p>This is black</p>
</body>
</html>
```

เมื่อคุณคัดด้วยเบราว์เซอร์ จะพบว่า เฉพาะข้อความ “This text is red” เท่านั้นที่มีสีแดง ทั้งนี้เนื่องจากการใช้สไตล์ชีทแบบ Inline จึงมีผลเฉพาะแท็กที่มีการกำหนดสไตล์ชีทไว้

2.2.2) ภาษาจาวาสคริปต์ (Javascript)

เป็นภาษาที่นำมาใช้ในการตกแต่งและเพิ่มความน่าสนใจให้กับเว็บไซต์ เป็นภาษาที่ได้พัฒนามาจากภาษาจาวาอีกต่อหนึ่ง โดยมุ่งเน้นเพื่อมาใช้ในสื่อทางอินเทอร์เน็ตโดยเฉพาะ เป็นภาษาที่เน้นความเรียบง่ายในการเขียน สามารถเข้าใจได้ง่าย จาวาสคริปต์ได้ถูกออกแบบเพื่อมาเป็นส่วนเพิ่มขยายในภาษา HTML ทำให้เราสามารถควบคุมหน้าตาของเว็บเพจได้ง่ายและตรงกับความต้องการของผู้ออกแบบเว็บเพจมากยิ่งขึ้น นอกจากนี้การใช้ภาษาจาวาสคริปต์ในการทำเว็บไซต์ยังเป็นการลดภาระในการประมวลผลของเครื่องเซิร์ฟเวอร์ เพราะจาวาสคริปต์จะทำการประมวลผลที่เครื่องของผู้ใช้ ทำให้เซิร์ฟเวอร์มีเวลาทำงานบริการด้านอื่นๆมากขึ้น

หลักการเขียนภาษาจาวาสคริปต์

1.) ถ้าเราต้องการเขียนจาวาสคริปต์ไว้ในส่วนของ `<head>.....</head>` หรือส่วนของตัวภาษา HTML ซึ่งอยู่ภายใต้แท็ก `<body>.....</body>` จำเป็นต้องใช้แท็ก `<script>....</script>` ครอบส่วนที่เป็นคำสั่งภาษาจาวาสคริปต์เหล่านั้นไว้

2.) การระบุรุ่นของภาษาจาวาสคริปต์ สามารถทำได้โดยการใช้แอทริบิวต์ `language="รุ่นของจาวาสคริปต์"` เช่น `<script language="JavaScript1.1">.....</script>` เป็นการระบุว่า จะใช้ฟังก์ชันต่างๆของจาวาสคริปต์รุ่น 1.1 แต่โดยปกติถ้าเราไม่ระบุรุ่นของภาษาจาวาสคริปต์ เว็บเบราว์เซอร์จะถือว่าเป็นการใช้จาวาสคริปต์รุ่น 1.0 ซึ่งเว็บเบราว์เซอร์ส่วนใหญ่จะสนับสนุนภาษาจาวาสคริปต์รุ่นนี้เป็นอย่างดี ทั้งเว็บเบราว์เซอร์ของไมโครซอฟท์และของเนตสเคป

3.) จาวาสคริปต์จะตีความคำสั่งต่างๆตามการเว้นวรรคและการใช้เครื่องหมายอัฒภาค (;) ที่คั่นระหว่างข้อความ โดยไม่ถือการขึ้นบรรทัดใหม่เป็นเรื่องสำคัญ ดังนั้นถึงบางครั้งลักษณะการเขียนจะต่างกันแต่ก็สามารถทำงานได้เหมือนกัน

ตัวอย่าง

```
<script language="JavaScript">{
    document.bgColor="#3333FF";
    alert("Hello,World");
}
```

จะให้ผลการทำงานเหมือนกันกับการเขียนในรูปแบบต่อไปนี้

```
<script language="JavaScript">
    { document.bgColor="#3333FF"; alert("Hello,World"); }
```

4.) การเขียนจาวาสคริปต์ ควรที่จะใส่คอมเมนต์ทุกครั้งในส่วนที่เป็นคำสั่งต่างๆของจาวาสคริปต์ เพื่อป้องกันการแสดงผลที่ผิดพลาดของเบราว์เซอร์ที่ไม่รู้จักภาษาจาวาสคริปต์ ดังตัวอย่างที่แสดงไว้ข้างใต้

```
<script language="JavaScript">
<!-- //Start Hide from old browser
    คำสั่งต่างๆของจาวาสคริปต์
//End hide →
</script>
```

5.) การบันทึกไฟล์ภาษาจาวาสคริปต์สามารถบันทึกเป็นนามสกุล .html , .htm หรือ .js ก็ได้ ซึ่งเบราว์เซอร์สามารถแปลและแสดงผลได้ถูกต้องเหมือนกันหมด แต่ส่วนใหญ่เราจะบันทึกให้

อยู่ในรูปของ .html หรือ .htm โดยจาวาสคริปต์จะถูกครอบโดยแท็ก <SCRIPT> และ </SCRIPT> อยู่ในไฟล์เหล่านั้น

ตัวอย่างจาวาสคริปต์

```
<html>
<title>Transition menu</title>
<meta http-equiv="Content-Type" content="text/html; charset=window-874">
<script language="JavaScript">
<!--start hide from old browser
function Trans(state)
{
    var count=0;
    count=count+state;
    if (count==1)
        pic1.src="image/vivian40.jpg";
    else
        pic1.src="image/vivian45.jpg";
    pic1.filters[0].apply();
    pic1.filters[0].play(2);
}
//end hide ->
</script>
</head>
<body bgcolor="#000000">
<table border="0" width="300" height="40" align="center">
<tr><td><div align="center"><a href="#" onmouseover="Trans(1)">Vivian1</a>
</div></td>
<td><div align="center"><a href="#" onmouseover="Trans(0)">Vivian2</a>
</div></td></tr>
</table>
<table border="1" width="262" height="378" align="center">
```

```

<tr><td></td>
</tr>
</table>
</body>
</html>

```

2.2.3) ภาษา ASP (Active Server Page)

Active Server Page หรือ ASP เป็นโปรแกรมที่พัฒนาขึ้นโดยบริษัทไมโครซอฟท์ เพื่อใช้งานด้านอินเทอร์เน็ต โดย ASP จะทำหน้าที่แปลความหมายของข้อความที่เขียนด้วยภาษาสคริปต์ เช่น VBScript โดยที่ข้อความเหล่านั้นมี ASP Tag (<%.....%>) กำกับอยู่ ซึ่งเว็บเบราว์เซอร์ทั่วไปไม่สามารถแปลความหมายได้ ดังนั้นการแปลความหมายของไฟล์ที่มีนามสกุล .asp จะเกิดขึ้นที่เครื่องเซิร์ฟเวอร์เท่านั้น จึงเรียกการทำงานแบบนี้ว่า *Server Side* เว็บเซิร์ฟเวอร์ที่สามารถแปลความหมายของเอกสาร ASP ได้ ต้องเป็นเว็บเซิร์ฟเวอร์ของบริษัทไมโครซอฟท์เท่านั้น เช่น Internet Information Server , Personal Web Server เป็นต้น เมื่อเว็บเซิร์ฟเวอร์ทำการแปลความหมายของเอกสาร ASP แล้วจะได้ผลลัพธ์เป็นเอกสาร HTML หลังจากนั้นจึงส่งเอกสารดังกล่าวไปให้กับเว็บเบราว์เซอร์

ตัวอย่าง

```

<%
for n=1 to 3
response.write("Hello World" & n & "<br>")
next
%>

```

ผลที่ได้คือ

```

Hello World1
Hello World2
Hello World3

```

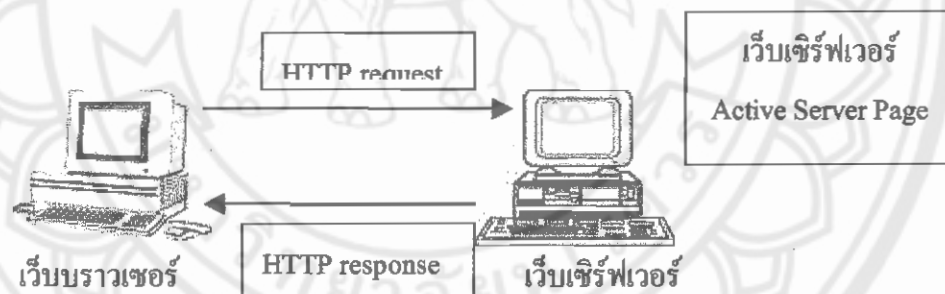
การเขียนภาษา ASP ทำได้ง่ายๆ บนโปรแกรม Text Editor ทั่วๆ ไป เช่น โปรแกรม NotePad , โปรแกรม WorldPad เป็นต้น แต่ในการทำโครงการนี้ ผู้ดำเนินโครงการได้ใช้โปรแกรม

Microsoft Visual Interdev เวอร์ชัน 6.0 เป็นเครื่องมือที่ช่วยในการเขียน ASP เพราะมีสิ่งอำนวยความสะดวกในการเขียนอยู่มากมาย ช่วยลดข้อผิดพลาดในการเขียนเอกสาร ASP ได้มาก

2.2.3.1) กระบวนการทำงานของ ASP

กระบวนการทำงานของ ASP จะมีขั้นตอนต่างๆ ดังต่อไปนี้

- 1.) เว็บเบราว์เซอร์ทำการร้องขอเอกสาร HTML ไปยังเว็บเซิร์ฟเวอร์โดยผ่านทางโปรโตคอล HTTP (HTTP request) โดยเอกสารที่ร้องขอไปจะเป็นไฟล์ที่มีนามสกุล .asp
 - 2.) เมื่อเว็บเซิร์ฟเวอร์ได้รับสัญญาณการร้องขอจากเว็บเบราว์เซอร์ ก็จะนำไฟล์ที่มีนามสกุลเป็น .asp ที่อยู่ที่เครื่องเซิร์ฟเวอร์นั้น ไปให้ตัวแปลภาษา ASP แปลความหมาย เมื่อแปลความหมายเสร็จแล้วจะได้เอกสาร HTML ออกมา
 - 3.) เว็บเซิร์ฟเวอร์ทำการส่งเอกสาร HTML ที่ได้จากการแปลของตัวแปลภาษา ASP นั้น กลับไปให้เว็บเบราว์เซอร์โดยผ่านทางโปรโตคอล HTTP เช่นกัน (HTTP response)
- การทำงานของ ASP จะเป็นดังรูป



รูปที่ 2.3 การทำงานของ ASP

2.2.3.2) ความรู้พื้นฐานเกี่ยวกับ VBScript

การเขียนเอกสาร ASP สามารถเขียนโดยใช้ภาษาสคริปต์ได้ 2 ภาษา คือ VBScript และ Jscript แต่ในที่นี้เราจะใช้ VBScript ในการเขียน เนื่องจากโครงสร้างของภาษาสามารถทำความเข้าใจได้ง่าย มีลักษณะคล้ายกับ Microsoft Visual Basic การทำงานของภาษาสคริปต์ที่นำมาใช้เขียนเอกสาร ASP นี้ ควรที่จะกำหนดให้ทำงานที่ฝั่งเซิร์ฟเวอร์ เพื่อที่เวลาดูด้วยเว็บเบราว์เซอร์ที่ไม่ใช่ของไมโครซอฟท์จะได้แสดงผลได้อย่างถูกต้อง การกำหนดชนิดของภาษาสคริปต์ที่ใช้ใน

TK
7685
A6
21510
2543

- 9 พ.ศ. 2544

4440097



การเขียนเอกสาร ASP ต้องกำหนดไว้ที่ส่วนหัวของเอกสาร แต่ถ้าไม่ระบุ ทางเว็บเซิร์ฟเวอร์จะถือว่าเป็นภาษา VBScript

```
<%@ Language="VBScript">
```

ชนิดของข้อมูลใน VBScript

ในภาษา VBScript จะมีชนิดข้อมูลอยู่เพียงชนิดเดียวคือ variant ซึ่งค่าข้อมูลจะขึ้นอยู่กับสถานการณ์ที่เราใช้เป็นข้อมูลชนิดอะไร เช่น เมื่อทำงานกับอักขรก็ใช้เป็นข้อมูลชนิดสตริง เมื่อทำงานกับข้อมูลที่เป็นตัวเลขก็ใช้เป็นจำนวนเต็มหรือจำนวนทศนิยมแบบยาว (Double) เป็นต้น

การประกาศตัวแปร

เนื่องจากใน VBScript มีชนิดของตัวแปรอยู่เพียงชนิดเดียว ดังนั้นเราจึงไม่ต้องประกาศชนิดของตัวแปร การประกาศตัวแปรจะใช้คำสั่ง Dim ดังตัวอย่าง

```
Dim first,second,third
```

หรือจะใช้การค่าให้ตัวแปรก็ได้ เช่น first=20

ถ้าต้องการสร้างตัวแปรอาร์เรย์ ก็สามารถทำได้ด้วยการประกาศตัวแปรดังตัวอย่างต่อไปนี้

```
Dim Check(10)
```

เป็นการสร้างตัวแปรแบบอาร์เรย์ที่มีสมาชิกจำนวน 10 ตัว คือ Check(0) จนถึง Check(9)

ค่าคงที่

เป็นการกำหนดค่าให้กับตัวแปร เพื่อให้เราสามารถนำตัวแปรนั้นไปใช้ในส่วนต่างๆ ของโปรแกรมต่อไป ถ้าหากมีการเปลี่ยนแปลงก็ทำการแก้ไขที่ค่าเริ่มต้นที่เดียว ก็จะมีผลกับทุกส่วนของโปรแกรมที่มีการอ้างถึงตัวแปรค่าคงที่ตัวนี้

รูปแบบ

```
const ตัวแปร 1, ตัวแปร 2, ....
```

ตัวแปร 1 = ค่าคงที่

ตัวแปร 2 = ค่าคงที่

เช่น

```
const pi, radias
```

```
pi=3.1416
```

```
radias=5
```

ฟังก์ชันและโพรซีเจอร์

ในภาษา VBScript จะมีทั้งฟังก์ชันและโพรซีเจอร์ให้ใช้งาน ส่วนที่แตกต่างกันระหว่างฟังก์ชันและโพรซีเจอร์คือ ฟังก์ชันจะใช้ในกรณีที่ต้องการให้มีการคืนค่ากลับเมื่อมีการเรียกให้

ฟังก์ชันทำงาน ส่วนโพรซีเจอร์จะไม่มีค่าคืนค่าใดๆ กลับมาหลังจากที่ทำงานเสร็จแล้ว การทำงานของฟังก์ชันหรือโพรซีเจอร์นั้นต้องระบุด้วยว่าให้สคริปต์ทำงานที่ฝั่งเซิร์ฟเวอร์(RunAT="server")

รูปแบบของฟังก์ชัน

Function ชื่อฟังก์ชัน (ตัวแปรที่รับค่าเข้ามาทำงาน)

การกระทำของฟังก์ชัน

end function

ในกรณีที่ต้องการให้มีการคืนค่ากลับมา ต้องกำหนดค่าที่จะคืนออกมานั้นให้กับตัวแปรที่มีชื่อเดียวกับชื่อฟังก์ชันที่เรียกทำงาน

รูปแบบของโพรซีเจอร์

Sub ชื่อ โพรซีเจอร์ (ตัวแปรที่ใช้รับค่าเข้ามาทำงาน)

การทำงานของโพรซีเจอร์

end Sub

2.2.3.3) ออบเจกต์สื่อสารของ ASP

1.) ออบเจกต์ ServerVariables

ออบเจกต์นี้จะถูกสร้างขึ้นเมื่อเว็บเซิร์ฟเวอร์ได้รับสัญญาณการร้องขอจากเว็บเบราว์เซอร์ ทางเว็บเซิร์ฟเวอร์จะทำการสร้างตัวแปรต่างๆ ขึ้นมาใช้จำนวนหนึ่ง เราเรียกตัวแปรเหล่านี้ว่า ServerVariables ซึ่งมีดังต่อไปนี้

AUTH_TYPE	เป็นวิธีที่เว็บเซิร์ฟเวอร์ใช้ตรวจสอบรายชื่อผู้ใช้สคริปต์ ในกรณีที่สคริปต์ดังกล่าวมีการจำกัดสิทธิการเรียกดู
CONTENT_LENGTH	ขนาดข้อมูลที่ส่งมาจากไคลเอนท์
CONTENT_TYPE	ชนิดของข้อมูลที่ส่งมาจากไคลเอนท์
GATEWAY_INTERFACE	หมายเลขรุ่น CGI/revision เช่น CGI/1.1
LOGON_USER	ชื่อของผู้ใช้ที่เป็นสมาชิกนของระบบปฏิบัติการ Windows NT
PATH_INFO	ชื่อ alias หรือ virtual directory ของเอกสาร
PATH_TRANSLATED	ที่อยู่จริงของเอกสาร
QUERY_STRING	ข้อมูลที่ส่งมาโดยวิธีการ get จะเป็นค่าที่อยู่ต่อท้ายชื่อเอกสาร โดยมีเครื่องหมาย ? กั้นอยู่
REMOTE_ADDR	IP address ของไคลเอนท์ที่เรียกมา

REMOTE_HOST	ชื่อเซิร์ฟเวอร์ของไคลเอนท์ที่เรียกเข้ามา
REQUEST_METHOD	วิธีการส่งข้อมูล มี 2 แบบคือ get และ post
SCRIPT_NAME	ชื่อเอกสาร ASP ที่กำลังแสดงอยู่
SERVER_NAME	ชื่อเครื่องเซิร์ฟเวอร์ที่ให้บริการ
SERVER_PORT	หมายเลขพอร์ตที่ใช้ในการติดต่อ ปรกติจะเป็นพอร์ต 80
SERVER_PROTOCOL	ชนิดของการเชื่อมต่อ เช่น HTTP/1.1
SERVER_SOFTWARE	ชื่อ โปรแกรมเว็บเซิร์ฟเวอร์
HTTP_ACCEPT_LANGUAGE	ภาษาที่ใช้อยู่ เช่น th
HTTP_CONNECTION	สภาพของการติดต่อระหว่างเว็บเซิร์ฟเวอร์กับเว็บเบราว์เซอร์
HTTP_USER_AGENT	ชนิดของเว็บเบราว์เซอร์ที่ทำการส่งสัญญาณเรียกเข้ามายังเว็บเซิร์ฟเวอร์

2.) ออบเจกต์ ClientCertificate

เป็นค่าที่นำมาตรวจสอบผู้ที่เรียกเข้ามายังเซิร์ฟเวอร์ เพื่อความปลอดภัยของข้อมูล จะใช้ได้ก็ต่อเมื่อเว็บเซิร์ฟเวอร์นั้นๆ รองรับระบบความปลอดภัยแบบ SSL3.0/PCT1 เท่านั้น และเมื่อเรียกใช้งานต้องระบุ URL ขึ้นต้นด้วย https:// แทน http://

3.) ออบเจกต์ Cookies

เป็นค่าของตัวแปรที่เก็บไว้ในเครื่องของไคลเอนท์ เมื่อมีการติดต่อกับเว็บเซิร์ฟเวอร์ โปรแกรมเว็บเซิร์ฟเวอร์จะทำการบันทึกค่าตัวแปรที่ใช้ในการติดต่อกันไว้ในเครื่องของไคลเอนท์

4.) อ็อบเจกต์ Form และ QueryString

เป็นอ็อบเจกต์ที่ทำงานคล้ายคลึงกัน คือการส่งข้อมูลจาก ฟังก์ชันไคลเอนท์ แต่วิธีที่สร้างอ็อบเจกต์ขึ้นมาแตกต่างกัน คือ

get จะเป็นแบบ QueryString

post จะเป็นแบบ Form

การส่งข้อมูลแบบ get จะสามารถส่งได้โดยไม่ต้องสร้างแบบฟอร์ม คือส่งในบรรทัดเดียวกับ URL ได้เลยหรือว่าจะส่งโดยการใช้ ฟอร์มก็ได้ ข้อมูลที่ส่งเป็นสามารถรับรู้โดยผู้อื่นได้ และมีขนาดข้อมูลได้ไม่เกิน 255 ตัวอักษร ดังตัวอย่างดังนี้

```
<form action="return.asp" method="get">
```

```
ชื่อ : <input type="text" name="name">
```

```
<input type="Submit" value="ส่ง">
```

```
<input type="Reset" value="ยกเลิก.">
```

```
</form>
```

เมื่อทำการใส่ข้อมูลใน ฟอรั่มแล้วด้วยวิธีการ get แล้ว ASP จะทำงานดังนี้

```
<%
  gname=Request.QueryString("name")
%>
```

การส่งข้อมูลแบบ post มีการทำงานที่เหมือนกันแต่ต้องอาศัยการส่งข้อมูลด้วยฟอรั่ม และข้อมูลที่ส่งไปจะไม่เป็นที่เปิดเผยมีความปลอดภัยมากกว่า และไม่จำกัดขนาดของข้อมูลที่จะส่งไปในแต่ละครั้ง จึงมักใช้ในการส่งข้อมูลที่มีจำนวนมากๆ และต้องการความปลอดภัยของการส่ง

2.2.3.4) การทำงานของ ASP กับเพิ่มข้อความ

การทำงานของ ASP กับเพิ่มข้อความ ใช้อ็อบเจกต์ FileSystem ในการจัดการเพิ่มข้อความ (Text File) ในการสร้างการเขียนและการอ่านเพิ่มข้อความ ด้วยเมธอด ดังนี้คือ

CreateTextFile คือการสร้างเพิ่มข้อมูลที่ต้องการขึ้นมา
รูปแบบ

```
[Object.]CreateTextFile( filename, overwrite, unicode)
```

โดยที่

filename	ชื่อไฟล์ข้อความที่จะสร้างขึ้นมา
overwrite	true คือการสร้างไฟล์ทับไฟล์เดิม ถ้ามีอยู่แล้ว false คือการสร้างไฟล์ขึ้นมาใหม่ ไม่ให้ทับกับของเดิม
unicode	true คือการสร้างเป็นไฟล์ข้อความชนิด unicode false คือการสร้างเป็นไฟล์ข้อความชนิด ASCII (เป็นค่า Default)

ตัวอย่าง

```
<%
set fs=CreateObject("Scripting.FileSystemObject")
set a=fs.CreateTextFile("D:\ASPDOC\text.txt",true)
a.WriteLine("Text File นี้สร้างจาก chap5-1.asp")
a.Close
%>
```

เป็นการสร้างเพิ่มข้อความที่ชื่อ text.txt ใน D:\ASPDOC\

OpenTextFile คือเมธอดที่ใช้ในการอ่านเพิ่มข้อความ ไม่ว่าจะเพิ่มข้อความที่สร้างด้วยเมธอด *CreateTextFile* หรือเป็นเพิ่มข้อมูลที่มีอยู่แล้ว

รูปแบบ

[Object.]OpenTextFile(filename,iomode,create,format)

โดยที่

filename ชื่อไฟล์ที่จะสร้างขึ้นใหม่หรือเปิดของเดิมที่มีอยู่แล้ว
 iomode 1= เปิดเพื่ออ่าน
 8=เปิดไฟล์เพื่อเขียนต่อเติม
 create true คือ เมื่อไม่พบ ให้สร้างไฟล์ชื่อนั้นขึ้นมา
 false คือ เมื่อไม่พบ ไม่ต้องสร้างไฟล์ขึ้นมาใหม่ (เป็นค่า Default)
 format ถ้าไม่ระบุ จะเป็นประเภท ASCII

ตัวอย่าง

```
<%
ForReading=1
set fs=CreateObject("Scripting.FileSystemObject")
set a=fs.OpenTextFile("D:\ASPDOC\text.txt",ForReading)
response.write(a.ReadLine())
a.close
%>
```

เป็นการเปิดเพิ่มข้อความชื่อ text.txt เพื่อทำการอ่าน โดยเป็นไปตามการกำหนดค่า iomode คือ ForReading=1 ค่า iomode เป็น 1 เพื่อให้ทำการอ่านอย่างเดียว แต่ถ้ากำหนดให้เป็น ForAppend= 8 จะทำการเขียนต่อจากข้อมูลเดิม

2.2.3.5) ASP กับการทำงานกับฐานข้อมูล

เมื่อทำการสร้างเพิ่มฐานข้อมูลขึ้นมาแล้วเก็บไว้ที่ฝั่งเซิร์ฟเวอร์ การจะเข้าถึงข้อมูลในแฟ้มนี้ได้จะต้องใช้ ASP ทำการติดต่อกับ ODBC ที่ฝั่งเซิร์ฟเวอร์ ด้วยการสร้าง อ็อบเจกต์เพื่อทำการติดต่อฐานข้อมูลด้วยคำสั่งดังนี้

```
<%
set Conn=Server.CreateObject("ADODB.Connection")
Conn.open "EakWork1","",""
set RS=Conn.Execute("Select * from Customers")
```

%>

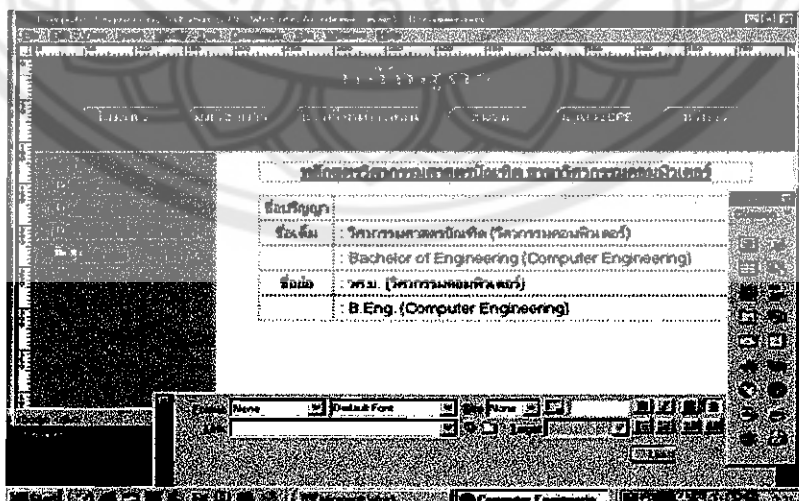
อ็อบเจกต์ที่ชื่อ Conn ถูกสร้างขึ้น และอ็อบเจกต์ Conn ทำการเปิด DSN ที่ชื่อ EakWork1 แล้วก็ สามารถติดต่อกับแฟ้มฐานข้อมูลที่ DSN ติดต่ออยู่ได้ ด้วยคำสั่ง SQL

2.3) เครื่องมือที่ช่วยในการเขียนเว็บเพจ

ในปัจจุบันนี้ได้มีเครื่องมือที่สามารถเขียนโฮมเพจได้ง่ายและสะดวกรวดเร็วมากยิ่งขึ้น โดยที่เราไม่ต้องทำการเขียนโค้ด HTML เองทั้งหมดหรือเครื่องมือบางตัว ผู้ใช้ไม่ต้องเขียนโค้ดเองเลย โปรแกรมจะทำการสร้างโค้ดขึ้นมาให้อัตโนมัติ ในปัจจุบันโปรแกรมประเภทหลังนี้ได้รับความนิยมเป็นอย่างมาก เนื่องจากผู้ใช้ไม่ต้องมาเขียนโค้ดเองและสามารถเห็นรูปร่างหน้าตาของเว็บเพจไปพร้อมๆ กับการออกแบบ เราเรียกโปรแกรมที่มีการทำงานแบบนี้ว่า WYSIWYG (What You See Is What You Get) โปรแกรมลักษณะนี้ที่เป็นที่รู้จักแพร่หลายมีอยู่ด้วยกันหลายโปรแกรม ทางคณะผู้ดำเนินโครงการได้เลือกนำมาใช้ 1 โปรแกรมคือ Dreamweaver เวอร์ชัน 3

2.3.1) Macromedia Dreamweaver 3.0

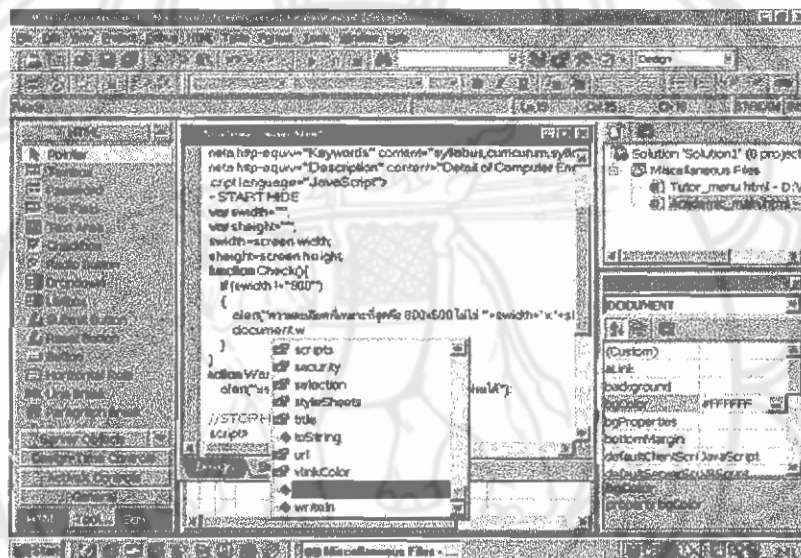
เป็นเครื่องมือที่ใช้ในการทำเว็บไซต์ที่ได้รับความนิยมมากตัวหนึ่ง เนื่องมาจากการทำงานในลักษณะ WYSIWYG(What You See Is What You Get) ผู้ใช้เพียงแค่พิมพ์ข้อความลงไป นำรูปภาพมาวาง ทำการตกแต่งและกำหนดคุณสมบัติของส่วนต่างๆของเอกสาร HTML โดยผู้ใช้ไม่ต้องเขียนโค้ดภาษา HTML เลย อีกทั้งยังมีเครื่องมือที่ช่วยในการตรวจสอบลิงค์และตรวจสอบการแสดงผลกับเว็บเบราว์เซอร์รุ่นต่างๆ อีกด้วย



รูปที่ 2.4 Macromedia Dreamweaver 3.0

2.3.2) Microsoft Visual Interdev 6.0

เป็นเครื่องมือที่ใช้ในการเขียนจาวาสคริปต์และ ASP มีสิ่งอำนวยความสะดวกต่าง ๆ มากมาย ใช้งานคล้ายกับ Microsoft Visual Basic มีส่วนของตาราง property ต่างๆ เมื่อพิมพ์ชื่อคำสั่งหรือตัวแปรจะมีการขึ้น property ของออบเจกต์นั้นๆให้ ทำให้ผู้ใช้สามารถเขียนโปรแกรมได้ถูกต้องมากยิ่งขึ้น เมื่อเขียนเสร็จแล้วต้องการดูหน้าตาของเว็บเพจก็สามารถใช้ Quick View ในการดูหน้าตาของเว็บได้โดยทันที นอกจากนี้ยังสามารถทำงานในลักษณะ WYSIWYG ได้อีกด้วย



รูปที่ 2.5 Microsoft Visual Interdev 6.0

2.4) กราฟิกและภาพเคลื่อนไหวที่ใช้ในการทำเว็บไซต์

การทำงานด้านกราฟิกและภาพเคลื่อนไหวในการตกแต่งเว็บเพจเราจะใช้โปรแกรมโฟโต - ชีปในการตกแต่งภาพ ใช้โปรแกรมอิมเมจลิคคิในการสร้างภาพเคลื่อนไหวและปรับภาพเป็น ส่วนๆ ที่เล็กลงเพื่อนำมาใช้กับเว็บไซต์ ดังหัวข้อต่อไปนี้

2.2.1) โปรแกรมโฟโตชีป เวอร์ชัน 5.5 (Photoshop 5.5) และการทำงาน

การทำงานในด้านกราฟิกอาศัยโปรแกรมโฟโตชีปช่วยการสร้างภาพเพื่อใช้ในเว็บเพจซึ่ง จะใช้เป็นโปรแกรมที่ใช้ในการสร้างภาพ และตกแต่งภาพให้ดูสวยงามและเหมาะสำหรับการนำไป ใช้ในเว็บเพจ โปรแกรมนี้เป็นโปรแกรมที่มีประสิทธิภาพและใช้งานได้ไม่ยาก สามารถนำมาสร้าง ภาพเพื่อใช้ในการตกแต่งเว็บเพจ การทำพื้นหลัง ปุ่มได้ตอบ ภาพประกอบ ฯลฯ



รูปที่ 2.6 โปรแกรมโฟโตช้อป

โปรแกรมโฟโตช้อปทำงานในลักษณะเป็นเลเยอร์ เหมือนกับการนำแผ่นใสหลายๆแผ่นมาวางซ้อนกัน โดยที่แผ่นใสแต่ละแผ่นจะมีองค์ประกอบภาพที่แตกต่างกัน เมื่อนำมารวมกันจะได้ภาพๆเดียว โปรแกรมสามารถปรับรูปแบบไฟล์ภาพ เพื่อนำไปใช้งานได้หลายๆแบบ โดยเฉพาะงานด้านเว็บเพจ โดยสามารถปรับเป็นรูปแบบต่างๆได้ดังนี้ เช่น GIF , JPEG ,PNG,การบันทึกไฟล์ภาพสำหรับงานที่ใช้บนเว็บ โดยเฉพาะและยังมีโปรแกรมอิมเมจสเกดดีที่ใช้นำมาทำงานร่วมกับโปรแกรมโฟโตช้อป ในการทำภาพเคลื่อนไหวเพื่อใช้ในงานด้านเว็บเพจ



เลเยอร์

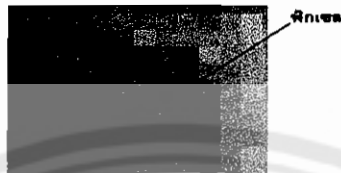
รูปที่ 2.7 เลเยอร์

การประมวลผลภาพ

ภาพบนคอมพิวเตอร์แบ่งออกเป็น 2 ประเภทคือ ภาพทางกายภาพ(Physical Image) คือรูปที่เราเห็นโดยทั่วไป เช่น รูปภาพถ่าย อีกประเภทหนึ่งคือ ภาพดิจิทัล(Digital Image) หรือคอมพิวเตอร์กราฟิก(Computer Graphic) คือรูปที่ใช้ในการประมวลและเก็บในคอมพิวเตอร์

ภาพที่ปรากฏบนจอคอมพิวเตอร์นั้นเกิดจากจุดสีสี่เหลี่ยมเล็กๆที่เรียกว่า พิกเซล (Pixel) ซึ่งเป็นองค์ประกอบที่เล็กที่สุดของภาพ มาประกอบกันเป็นภาพ วิธีการประมวลผลภาพของ

คอมพิวเตอร์มี 2 ลักษณะคือ การประมวลผลแบบเวกเตอร์ (Vector) และ การประมวลผลแบบบิตแมป (Bitmap)



รูปที่ 2.8 พิกเซล

การประมวลผลภาพแบบเวกเตอร์ คือ การประมวลผลโดยอาศัยการคำนวณทางคณิตศาสตร์ซึ่งเครื่องคอมพิวเตอร์จะเก็บข้อมูลเป็นสูตรทางคณิตศาสตร์ ดังนั้นภาพที่ประมวลผลแบบนี้จึงสามารถเคลื่อนย้ายที่หรือย่อขยายขนาดภาพ ลักษณะของภาพได้โดยที่ไม่เสียรูปทรงเชิงเรขาคณิต

การประมวลผลภาพแบบบิตแมป คือ การประมวลผลโดยอาศัยการอ่านค่าสีในแต่ละพิกเซล ซึ่งเหมาะกับภาพที่มีบริเวณสีที่ใกล้เคียงกัน เช่น ภาพถ่าย การประมวลผลแบบนี้เครื่องคอมพิวเตอร์จะเก็บข้อมูลเป็นข้อมูลดิบ คือ ค่า 0 และ 1 ใช้การแสดงผลเป็นแต่ละพิกเซล โดยแต่ละพิกเซลจะมีการเก็บค่าสีที่เจาะจงในแต่ละตำแหน่ง

ชนิดของไฟล์ที่ใช้ในงานกราฟิก

.PSD เป็นไฟล์พื้นฐานของโปรแกรมโฟโตช้อป ใช้บันทึกขั้นตอนการทำงานและแก้ไขงานที่ทำบนโปรแกรมโฟโตช้อปได้

.BMP เป็นไฟล์มาตรฐานของระบบปฏิบัติการวินโดวส์

.TIF เป็นไฟล์ที่ใช้ได้ทั้งเครื่อง พีซี และ แม็กอินทอร์ช ใช้ในงานที่เกี่ยวข้องกับสิ่งพิมพ์

.GIF เป็นไฟล์ที่นิยมใช้ในเครือข่ายอินเทอร์เน็ต เพราะง่ายต่อการบีบอัดข้อมูล และรวดเร็วในการส่งผ่านข้อมูล

.JPG เป็นไฟล์ที่มีการบีบอัดข้อมูลให้มีขนาดเล็ก และนิยมใช้ในอินเทอร์เน็ต

.PCT เป็นไฟล์ที่ใช้กับงานพิมพ์บนเครื่อง แม็กอินทอร์ช เท่านั้น

.PNG เป็นไฟล์ที่ถูกพัฒนามาจากแฟ้มข้อมูล .GIF แต่มีข้อดีคือมีการสูญเสียข้อมูลในการบีบอัดข้อมูล

การปรับภาพเพื่อนำเสนอบนเว็บด้วยโปรแกรมโฟโตช้อป 5.5

ภาพที่ใช้แสดงบนเว็บจะต้องมีขนาดเล็กดังนั้นรูปแบบเพิ่มข้อมูลที่ใช้จะต้องมีการบีบอัดข้อมูลซึ่งได้แก่ รูปแบบไฟล์ GIF, JPEG และ PNG

หลักการทั่วไปสำหรับการเลือกใช้ภาพคือ

.JPEG สำหรับภาพถ่าย หรือ ภาพวาดที่มีจำนวนสีมาก ๆ

.GIF หรือ .PNG-8 สำหรับภาพที่มีลักษณะเป็นลายเส้นหรือภาพที่ต้องการสร้างเป็นภาพโปร่งใส เช่น ภาพการ์ตูน

.PNG-24 สำหรับภาพถ่ายที่ต้องการให้มีลักษณะโปร่งแสงแต่ไฟล์ที่ได้หลังการบีบอัดจะมีขนาดใหญ่กว่าไฟล์ JPEG แต่มีคุณภาพมากกว่า

เมื่อต้องการภาพออกมาในรูปแบบใด โปรแกรมโฟโตช้อปสามารถที่จะบันทึกไฟล์นั้นให้เป็นรูปแบบตามที่ต้องการได้โดยใช้การทำบันทึกสำหรับการแสดงบนเว็บ (Save For Web) โดยสามารถกำหนดคุณลักษณะของรูปแบบไฟล์ที่เราจะบันทึกได้ เช่น ขนาดของไฟล์ คุณภาพข้อมูล ความเร็วในการขนถ่ายข้อมูล ฯลฯ และคุณผลลัพธ์ที่ปรากฏในบราวเซอร์ได้ในขณะที่ทำงาน หลังจากการกำหนดคุณสมบัติของไฟล์ตามความต้องการแล้วทำการบันทึกสำหรับการแสดงบนเว็บ (Save for web) ยังสามารถบันทึกภาพให้อยู่ในรูปเอกสาร HTML เพื่อนำเสนอบนเว็บ วิธีการปรับภาพมีขั้นตอนดังนี้

1.) ทำการเปิดเพิ่มภาพที่ต้องการด้วยโปรแกรมโฟโตช้อป หลังจากนั้นคลิกเมาส์ที่คำสั่ง File>Save For Web

2.) จะปรากฏหน้าต่างการทำงานของ Save For Web ขึ้น องค์ประกอบของหน้าต่าง Save For Web จะประกอบด้วย

Tab ใช้ในการกำหนดมุมมองของภาพแบ่งออกเป็น 4 รูปแบบมุมมอง คือ

- Original คือมุมมองภาพต้นฉบับที่ยังไม่ได้มีการแต่งภาพใดๆ
- Optimized คือมุมมองภาพที่ได้หลังการปรับแต่ง
- 2-Up คือมุมมองแสดงภาพเปรียบเทียบกัน ระหว่างสองแบบข้างต้น
- 4-Up คือมุมมองแสดงการเปรียบเทียบภาพ 4 ลักษณะ

ในการทำงานจะทำงานที่ มุมมอง Original แล้วเมื่อทำการปรับแต่งภาพเสร็จแล้วจึงค่อยปรับไปเป็นมุมมองแบบ Optimized

3.) ทำการเลือกรูปแบบไฟล์ว่าจะทำการแสดงเป็นแบบใดบนเว็บ และกำหนดลักษณะคุณสมบัติของไฟล์โดยคลิกเมาส์ที่ **Setting** แล้วทำการปรับเปลี่ยนข้อกำหนดต่างๆในที่นี้เราจะกำหนดให้เป็นไฟล์แบบ **JPEG** และมีคุณสมบัติดังนี้

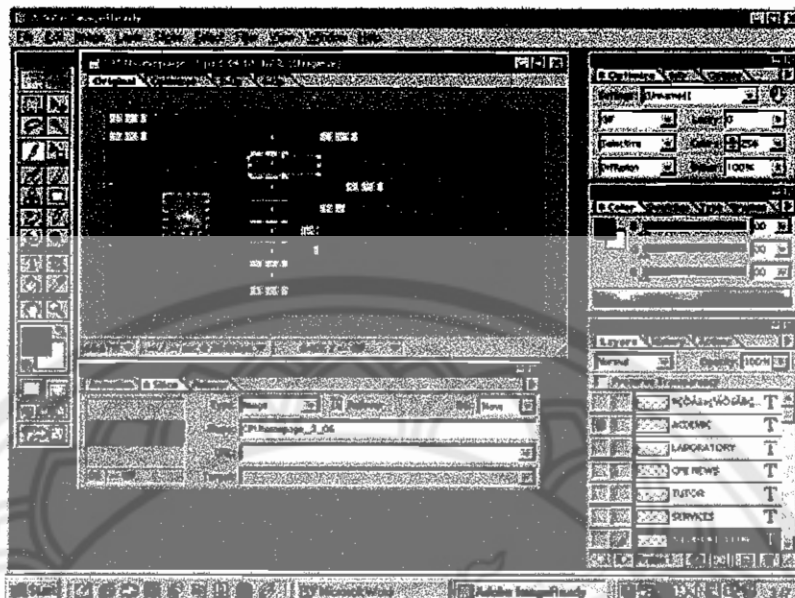
Quality = 30

- คลิกเมาส์ที่ **Optimized** เพื่อให้มีการปรับค่าให้เหมาะกับรูปแบบไฟล์ **JPEG**
 - คลิกเมาส์ที่ **Progressive** เพื่อให้เมื่อทำการแสดงภาพที่โปรแกรมแสดงเว็บจะแสดงภาพที่มีความละเอียดน้อยๆก่อนแล้วค่อยๆชัดขึ้นจนสมบูรณ์
 - กำหนดความเบลอของภาพที่ **Blur**
- 4.) ทำการบันทึกภาพเพื่อการแสดงบนเว็บคลิกเมาส์ที่ **OK** แล้วจึงทำการเลือกตำแหน่งที่เก็บเพิ่มแล้วคลิกเมาส์ที่ **Save HTML File**
- 5.) ทดลองแสดงภาพบนเว็บเบราว์เซอร์

2.4.2) โปรแกรมอิมเมจลีดดี 2.0 (ImageReady 2.0) และการทำงาน

การสร้างภาพเคลื่อนไหว

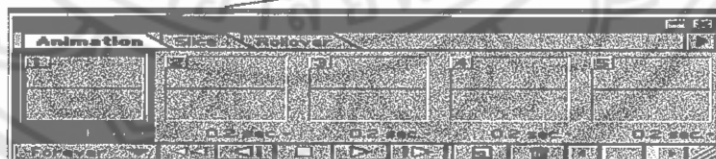
อิมเมจลีดดี 2.0 เป็นโปรแกรมตกแต่งภาพสำหรับไปใช้งานบนเว็บ และรวมถึงปุ่มกด ป้ายโฆษณา หรือที่เรียกว่าแบนเนอร์ สามารถสร้างภาพเคลื่อนไหวเพื่อใช้บนเว็บได้อีกด้วย สามารถบันทึกไฟล์ภาพที่จะแสดงบนเว็บไปเป็นรูปแบบไฟล์ที่ต้องการได้ บันทึกเป็นเอกสาร **HTML** ที่นำไปสร้างเว็บเพจได้และยังสามารถทำงานร่วมกับโปรแกรมโฟโตช้อปได้เป็นอย่างดี นอกจากนี้ยังสามารถแบ่งรูปออกเป็นส่วนย่อยๆ เพื่อให้โหลดได้เร็วขึ้น ซึ่งในทุกวันนี้ความเร็วในการโหลดข้อมูลเป็นสิ่งหนึ่งที่ดึงดูดให้ผู้คนเข้ามาเยี่ยมชมเว็บไซต์มากขึ้น



รูปที่ 2.9 โปรแกรมอิมเมจลิตตี

การสร้างภาพเคลื่อนไหวคือการนำเอาภาพนิ่งรูปแบบ GIF หลากๆภาพมาสร้างเป็นภาพเคลื่อนไหวที่เรียกว่า "GIF Animation" หลักการคือ การเปลี่ยนจากภาพหนึ่งไปเป็นอีกภาพหนึ่งไปทีละเฟรม (Frame) การเตรียมภาพนิ่งเพื่อที่จะนำมาทำภาพเคลื่อนไหวสามารถสร้างขึ้นมาได้จะโปรแกรมอิมเมจลิตตีเอง หรือจากโปรแกรมโฟโตช้อปแล้วนำมาทำงานที่โปรแกรมอิมเมจลิตตีก็ได้ โดยการตกแต่งภาพในแต่ละเลเยอร์เพื่อทำให้ภาพเคลื่อนไหวได้อย่างนุ่มนวล

Animation Palette



รูปที่ 2.10 Animation Palette

เมื่อได้ภาพนิ่งมาแล้วเราจะใช้การปรับแต่งที่โปรแกรมอิมเมจลิตตี เพื่อปรับแต่งภาพทีละเฟรมโดยการทำงานของ ตัวจัดการภาพเคลื่อนไหว (Animation Palette) เมื่อสร้างภาพแต่ละเฟรมจนครบแล้วจะต้องอาศัยการกำหนดเวลาหน่วงในการแสดงผลที่ ตัวจัดการภาพเคลื่อนไหว และ

สามารถดูการเคลื่อนไหวก่อนได้ หรือจะดูการเคลื่อนไหวที่แสดงบนเว็บที่เบราว์เซอร์ก็ได้โดยที่เรา
ยังคงทำงานอยู่ ดังขั้นตอนการสร้างต่อไปนี้

เริ่มการทำภาพเคลื่อนไหวจะต้องมีภาพนิ่งก่อน ซึ่งภาพนิ่งนี้สามารถทำในโปรแกรมโฟโต-
ช้อปโดยภาพที่ทำได้มีเลขเอร์ของภาพที่ต้องการให้เปลี่ยนไปเมื่อกำหนดค่าเฟรมเป็นค่าต่างๆ แล้ว
จึงทำการปรับแต่งภาพในแต่ละเฟรมใน โปรแกรมอิมเมจลิตคิดตามขั้นตอนต่อไปนี้

1.) เปิดภาพที่ต้องการพร้อมทั้งเปิดหน้าต่างจัดการภาพเคลื่อนไหว (Animation Palette) ใน
หน้าต่างนี้คลิกปุ่มการสร้างเฟรมใหม่แล้วทำการปรับแต่งภาพในเฟรมนี้ โดยปรับไปตามการเคลื่อนไหว
ที่เราต้องการ ทำเช่นนี้กับเฟรมอื่นๆอีกจนครบทั้งหมดของเฟรมการเคลื่อนไหว

2.) กำหนดเวลาหน่วง (Delay Time) ในการแสดงภาพเคลื่อนไหว โดยคลิกที่ปุ่มได้เฟรม
เพื่อกำหนดเวลาหน่วงของแต่ละเฟรมนั้นๆ

3.) ดูภาพเคลื่อนไหวที่สร้างขึ้นได้ที่หน้าต่างจัดการภาพเคลื่อนไหวนี้เอง โดยภาพจะเลื่อน
แสดงไปที่เฟรมตามขนาดของเวลาหน่วงที่ได้กำหนดเอาไว้

4.) ดูภาพเคลื่อนไหวที่สร้างในการแสดงผลบนเว็บได้ที่ คำสั่ง File>Preview In นอกจาก
ภาพเคลื่อนไหวที่แสดงอยู่บนเว็บแล้วยังแสดงโค้ดภาษา HTML ให้อีกด้วย

5.) ไฟล์ที่ได้เป็นไฟล์รูปแบบ GIF หลากๆภาพที่แสดงออกมาในเวลาที่แตกต่างกัน

การใช้ สไลซ์ (Slice)

ปัญหาที่สำคัญของการนำภาพขนาดใหญ่ขึ้นมาแสดงบนเว็บคือ การวางตำแหน่งของภาพ
ซึ่งต้องอาศัยการสร้างตารางย่อยๆ ในเว็บเพจจำนวนมากเพื่อใส่ภาพที่ทำการแบ่งเอาไว้เป็นส่วนๆ
ซึ่งต้องใช้เวลานานมาก อีกทั้งภาพที่มีขนาดใหญ่ยังโหลดได้ช้า



รูปที่ 2.11 สไลซ์

สไลซ์ เป็นความสามารถในโปรแกรมอิมเมจลิคคี่ ที่สามารถลดความยุ่งยากนี้ได้โดยการแบ่งภาพเป็นส่วนเล็กๆ เมื่อทำการบันทึกไฟล์ภาพพร้อมเอกสาร HTML ที่ใช้แสดงภาพบนเว็บเพจ โปรแกรมอิมเมจลิคคี่ก็จะทำการแบ่งภาพเป็นส่วนย่อยๆ สำหรับแสดงในบราวเซอร์ให้โดยอัตโนมัติ คุณสมบัติที่สำคัญของ สไลซ์ คือ การที่สามารถปรับภาพในแต่ละส่วนย่อยที่กำหนดเป็น สไลซ์ ได้แตกต่างกัน เช่น สไลซ์หนึ่งใช้รูปแบบ GIF ส่วนอีกสไลซ์หนึ่งในภาพเดียวกันใช้รูปแบบ JPEG ซึ่งสามารถปรับภาพให้เหมาะสมที่สุดในการแสดงบนเว็บได้

คุณสมบัติที่สำคัญของสไลซ์ อีกอย่างคือ สามารถกำหนดให้ สไลซ์ เป็นตัวเชื่อมโยงข้อมูลอื่นในเว็บเพจหรือต่างเว็บเพจกันได้ที่เราเรียกว่า ลิงค์ วิธีการสร้าง สไลซ์ มีดังนี้

- 1.) ทำการสร้างสไลซ์โดยใช้เมาส์คลิกที่เครื่องมือสไลซ์ทำการลากเมาส์กำหนดพื้นที่สไลซ์บนภาพ จะปรากฏกรอบสี่เหลี่ยมสีเหลืองขึ้นมา
- 2.) ทำการปรับแต่ง เคลื่อนย้ายสไลซ์ที่สร้างขึ้นได้ด้วยการไปคลิกเมาส์ที่เครื่องมือการเลือกสไลซ์ (Slice Selection Tool) แล้วปรับให้มีขนาดและตำแหน่งที่เหมาะสม
- 3.) ทำการกำหนดการเชื่อมโยง โดยไปคลิกเมาส์ที่เครื่องมือการเลือกสไลซ์และใช้มันไปคลิกเลือกสไลซ์ที่ต้องการ
- 4.) คลิกที่ Tab สไลซ์ เพื่อแสดงหน้าต่างการกำหนดคุณสมบัติสไลซ์ (Slice Palette) กำหนดค่าตำแหน่งข้อมูลที่เราต้องการเชื่อมโยง (URL)
- 5.) ดูการทำงานของสไลซ์ที่สร้างขึ้นในเว็บบราวเซอร์ โดยการคลิกเมาส์ที่ File>Preveiw In > Internet Explorer

การสร้างอิมเมจแมป

อิมเมจแมป (Image map) คือ ภาพในเว็บเพจที่ประกอบด้วยพื้นที่ที่ถูกกำหนดเป็นลิงค์เชื่อมโยงกับข้อมูลในเว็บ โดยพื้นที่ที่ถูกกำหนดให้เป็นลิงค์จะถูกเรียกว่า ฮอตสปอต(Hotspot)พื้นที่อิมเมจแมปมีความคล้ายกับการทำสไลซ์มาก แต่อิมเมจแมปมีข้อดีที่สามารถกำหนดพื้นที่ที่ทำเป็นลิงค์ได้หลายรูปแบบเช่น วงกลม หรือพื้นที่หลายเหลี่ยม แต่ในขณะที่สไลซ์กำหนดได้แก่พื้นที่รูปสี่เหลี่ยมเท่านั้น

หลักการ สร้างอิมเมจแมปคือการสร้างพื้นที่ส่วนที่เรากำหนดให้เป็นลิงค์บนภาพหรือฮอตสปอต ให้เป็นเลขอร์แยกออกมาจากภาพหลัก และกำหนดให้เลขอร์นั้นเป็นอิมเมจแมปแล้วไปกำหนดคุณสมบัติของเลขอร์ที่กำหนดเป็นฮอตสปอตให้ลิงค์ไปที่ที่ต้องการ โดยการกำหนดตำแหน่งที่อยู่ของเว็บ URL (Uniform Resource Locator)

หลังจากที่เราได้เปิดดูการทำงานของ อิมเมจแมป ในบราวเซอร์โดยใช้คำสั่ง Preview แล้ว

หากต้องการนำอิมเมจแมบไปใช้ในการสร้างเว็บเพจ ต่อไปต้องทำการบันทึกอิมเมจแมบนี้เป็นเอกสาร HTML โดยการเลือกที่ File<Save Optimized As ขั้นตอนการสร้างอิมเมจแมบมีดังนี้

1.) เปิดไฟล์ภาพที่ต้องการทำเป็นพื้นหลังออกมา ใช้เครื่องมือสำหรับการเลือกพื้นที่ทำการเลือกพื้นที่ที่ต้องการทำเป็นบริเวณเชื่อมโยง แล้วทำการคัดลอกภาพของพื้นที่ที่ทำการเลือกเอาไว้ด้วย กด Ctrl+C

2.) นำภาพที่คัดลอกมาวางในเลเยอร์ใหม่ที่สร้างขึ้นมาแล้ววางลง ณ ตำแหน่งเดิมเหมือนกับการซ้อนภาพเดิมเอาไว้

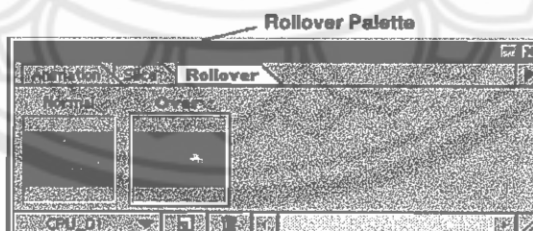
3.) ดับเบิลคลิกที่เลเยอร์นี้จะได้นหน้าต่างกำหนดคุณสมบัติ (Layer Option) กำหนดค่าที่ส่วน Shape เพื่อกำหนดรูปร่างของตัวเชื่อมโยงและค่าต่างๆ ที่ใช้ในการกำหนดการเชื่อมโยง

- Polygon คือรูปหลายเหลี่ยม
- Rectangle คือรูปสี่เหลี่ยม
- Circle คือรูปทรงกลม
- Tolerance กำหนดว่าเมาส์เข้าใกล้เท่าใดถึงจะทำงาน
- URL กำหนดตำแหน่งข้อมูลในเว็บที่ต้องการลิงค์ไป

4.) ตรวจสอบผลการทำอิมเมจแมบในการแสดงบนเว็บที่ File>Preview In >Internet Explore

การสร้างปุ่มตอบสนองผู้ใช้ (Rollover)

โรโอเวอร์ เป็นการสร้างปุ่มกด สัญลักษณ์ภาพ หรือภาพสำหรับแสดงบนเว็บที่มีลูกเล่นที่พิเศษ โดยเมื่อเราเลื่อนเมาส์ไปอยู่เหนือตำแหน่งของโรโอเวอร์ ก็จะเกิดการเปลี่ยนแปลงเช่น ปุ่มเปลี่ยนสี ภาพมีข้อความบางข้อความปรากฏขึ้น เป็นต้น



รูปที่ 2.12 Rollover

สิ่งที่สำคัญของการสร้างปุ่มด้วยการทำโรโอเวอร์ คือ การกำหนดสถานะของการแสดงการทำงานที่จะปรากฏเมื่อมีการกระทำของเมาส์คอปุ่มต่างๆ ซึ่งต้องไปกำหนดการทำงานที่ตัวจัดการ

การตอบสนองของปุ่ม(Rollover Palette) เพื่อทำการกำหนดสถานะของการทำงานของปุ่ม(Rollover State) ขั้นตอนการทำโรโอเวอร์มีดังนี้

สร้างเพิ่มภาพขึ้นมาใหม่โดยคลิกเมาส์ที่ New

1.) คลิกเมาส์ที่เครื่องมือสร้างปุ่ม (Button Tool) แล้วสร้างปุ่มขึ้นมากำหนดสีให้เป็นสีแดง

2.) ทำการคัดลอกปุ่มที่สร้างมาวางลงในเลเยอร์ใหม่ กำหนดปุ่มให้เป็นสีเหลือง

3.) กำหนดพื้นที่สไลซ์รอบปุ่มสีแดง

4.) ไปที่หน้าต่างการกำหนดลักษณะของปุ่ม(Rollover Palette) กำหนดสถานะการทำงานของ

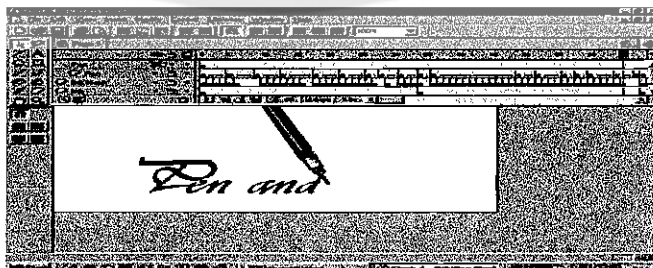
ของปุ่ม จากการทำงานแบบปกติ(Normal) ไปเป็นการตอบสนองแบบต่างๆคือ

- Over แสดงเมื่อเมาส์อยู่เหนือปุ่ม
- Down แสดงเมื่อคลิกเมาส์ค้างไว้
- Click แสดงเมื่อคลิกเมาส์
- Out แสดงเมื่อเลื่อนเมาส์ออก
- Up แสดงเมื่อปล่อยปุ่มเมาส์
- Custom กำหนดเงื่อนไขเองด้วยการเขียนจาวาสคริปท์
- None ไม่กำหนดการทำงาน

5.) คูการทำงานของปุ่มในโปรแกรมเว็บเบราว์เซอร์ จะปรากฏปุ่มที่มีสีต่างไปจากเดิมเมื่อนำเมาส์มากระทำการใดๆตามที่เรากำหนดเงื่อนไขเอาไว้

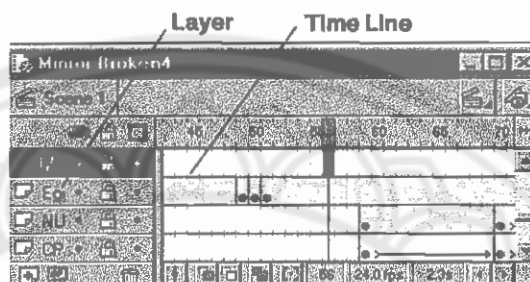
2.4.3)โปรแกรมเม็คโคมีเดียแฟลช 4 (Macromedia Flash 4) และการทำงาน

โปรแกรมแฟลชเป็นโปรแกรมไม่เน้นการเขียนโปรแกรมที่ยุ่งยาก แต่ก็สามารถเขียนคำสั่งได้โดยการ โปรแกรมคำสั่งอยู่ภายในองค์ประกอบของภาพที่เรียกว่า วัตถุ(Object) โปรแกรมนี้เน้นเกี่ยวกับการทำภาพเคลื่อนไหวในลักษณะของการ์ตูน ภาพเคลื่อนไหวที่ได้จะมีขนาดเล็กมาก จึงทำให้ได้รับความนิยมอย่างรวดเร็วในหมู่นักเขียนเว็บเพจ



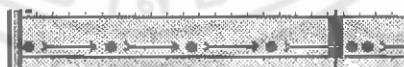
รูปที่ 2.13 Macromedia Flash 4

หลักการการทำงานของโปรแกรมแฟลชที่สำคัญก็คือ การลำดับการแสดงวัตถุ (Object) ที่อยู่บนเส้นเวลา (Time line) โดยวัตถุจะปรากฏบนเวที (Stage) ตามเวลาที่กำหนด และเนื่องจากวัตถุที่ปรากฏเป็นภาพแบบเวกเตอร์จึงทำให้ประหยัดเนื้อที่และสามารถดัดแปลงรูปทรงได้โดยที่ไม่สูญเสียความละเอียดของภาพไป



รูปที่ 2.14 เลเยอร์และเส้นเวลา

ภาพเคลื่อนไหวในโปรแกรมแฟลช มี 2 ประเภทใหญ่ๆคือ “Cell Animation” และ “Tweened Animation” ประเภทแรกเป็นการสร้างภาพเคลื่อนไหวทีละเฟรม แล้วนำมาแสดงในระยะเวลาใกล้เคียงกันเพื่อให้เกิดภาพที่เคลื่อนไหวได้ ในขณะที่ประเภทหลังจะเป็นการแสดงผลเฟรมหลักๆ แล้วให้โปรแกรมแฟลชสร้างภาพที่เหลือให้ ดังนั้นการสร้างภาพเคลื่อนไหวประเภท “Tweened Animation” จึงประหยัดเนื้อที่มากกว่า ลักษณะการใช้งานของทั้งสองประเภทก็แตกต่างกัน เช่น การเคลื่อนที่ของปีกของนกที่บินอยู่ ต้องอาศัยการทำ “Cell Animation” แต่ลักษณะของการเคลื่อนที่ของวัตถุจากจุดหนึ่งไปยังอีกจุดหนึ่ง ต้องอาศัยการทำ “Tweened Animation”



รูปที่ 2.15 Tweened Animation

หลักการสำคัญของการสร้างภาพเคลื่อนไหวคือการใช้งานเลเยอร์และเฟรมให้สัมพันธ์กับเส้นเวลา ขั้นตอนการสร้างภาพเคลื่อนไหวแบบ Motion Tweened ดังนี้

เมื่อต้องการสร้างภาพที่เคลื่อนที่จากเฟรมที่ 1 ไปยังเฟรมที่ 25 ทำได้ดังนี้

- 1.) คลิกเมาส์ขวาที่เฟรมที่ 1 แล้วเลือก Create Motion Tween
- 2.) ทำเช่นเดียวกับเฟรมที่ 1 ในเฟรมที่ 25
- 3.) จะปรากฏแถบสีฟ้าจากเฟรมที่ 1 ถึงที่ 25

4.) คู่มือการทำงานได้ที่ Window Controller

5.) คู่มือการทำงานของภาพเคลื่อนไหวได้ด้วยการกด Enter หรือเลือกที่ Publish Preview เพื่อดูบนเว็บเบราว์เซอร์

ขั้นตอนการสร้างภาพเคลื่อนไหวแบบ Shape Tweened

เมื่อต้องการทำการแปลงรูปหนึ่งให้กลายเป็นอีกรูปหนึ่ง

1.) เลือกรูปใดๆจาก Libraries คลิกมาวางที่เฟรมที่ 1

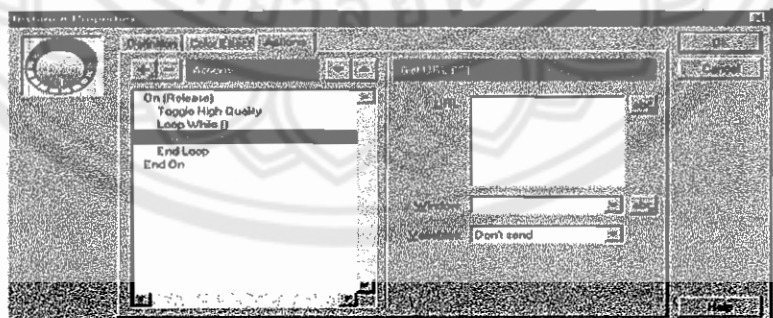
2.) ดับเบิลคลิกที่เฟรมที่ 1 จะได้ คุณสมบัติเฟรม (Frame Properties) คลิกที่แถบ Action เพื่อกำหนดให้เป็นแบบ Shape

3.) ไปที่เฟรมที่ 25 ทำการใส่เฟรมหลักลงไป (Insert Key Frame)

4.) ดับเบิลคลิกที่เฟรมที่ 25 ได้ คุณสมบัติเฟรม (Frame Properties) คลิกที่ Action เลือก Shape

5.) ที่เฟรมที่ 25 ทำการเปลี่ยนแปลงรูปไปตามที่เราต้องการ หลังจากที่ทำเสร็จแล้วกด Enter เพื่อดูผลที่เกิดขึ้น

โปรแกรมแฟลชสามารถนำเสียงมาใช้ประกอบการแสดงภาพเคลื่อนไหวได้โดยการกำหนดการใส่เสียงที่เฟรมที่ต้องการให้มีเสียงเกิดขึ้น นอกจากนี้ โปรแกรมแฟลชยังสามารถสร้างปุ่มควบคุมการทำงานของภาพเคลื่อนไหวและสามารถสร้างปุ่มการตอบสนองต่อเมาส์ของผู้ใช้ในการลิงค์ไปยังที่เว็บเพจต่างๆได้ โดยการกำหนดการควบคุมการทำงานที่คำสั่ง Get URL ที่ Instance Properties



รูปที่ 2.16 Instance Properties

เมื่อทำการสร้างงานเสร็จแล้ว โปรแกรมแฟลช สามารถนำเสนอผลงานให้เป็นไฟล์ในรูปแบบต่างๆ โดยการทำการนำออกไฟล์ (Export) ไปเป็นรูปแบบไฟล์ต่างๆดังนี้

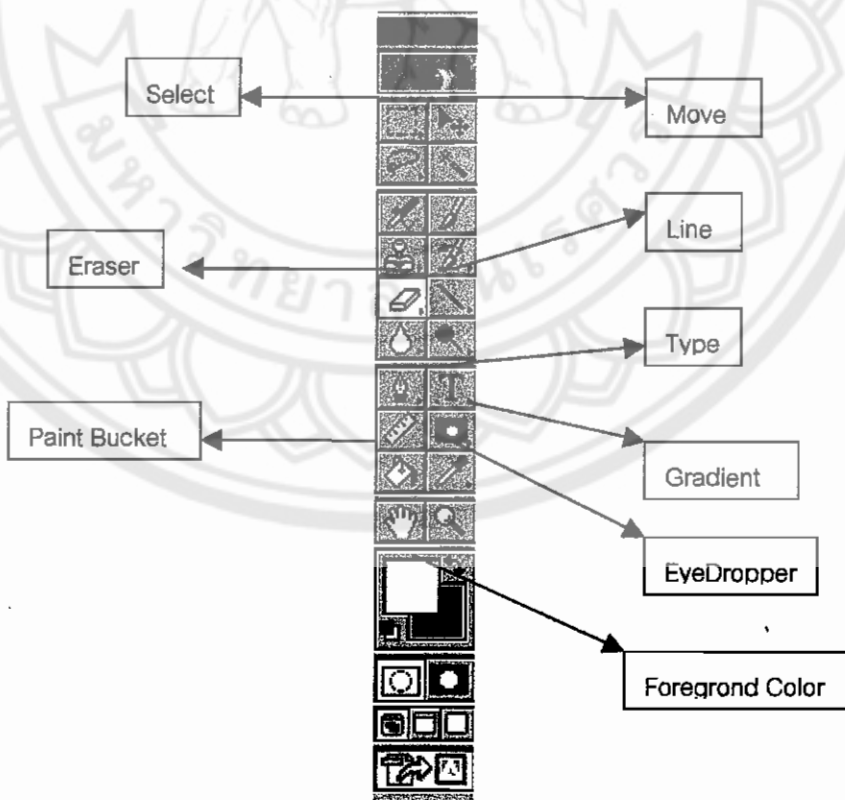
- ไฟล์ภาพยนต์ ในรูปแบบต่างๆ
- ไฟล์ที่ใช้เล่นกับ Flash Player (.SWF)
- ไฟล์ที่ใช้กับ โปรแกรมเดี่ยว(Standalone Player.EXE)
- ไฟล์เอกสาร HTML ที่แสดงภาพเคลื่อนไหวบนเว็บเพจ

การทำการนำออกไฟล์(Export File) สามารถกำหนดคุณสมบัติของไฟล์นั้นๆ ได้เช่น กำหนดคุณภาพ ,ความเร็วในการขนถ่ายข้อมูล , ขนาดของการแสดงภาพเคลื่อนไหว ฯลฯ ในการทำภาพเคลื่อนไหวในโปรแกรมเฟลชสามารถตรวจสอบการแสดงผลภาพเคลื่อนไหวที่แสดงบนเว็บได้ โดยการทำการคำสั่ง Preview และเลือกเปิดดูด้วยเว็บเบราว์เซอร์ ได้ตามต้องการ

2.5 เทคนิคที่ใช้ในการสร้างและตกแต่งภาพให้ดูสวยงาม

เทคนิคต่างๆที่ใช้นำมาจาก โปรแกรมที่กล่าวมาข้างต้นทั้งหมด สามารถนำมาใช้กับการทำงานจริงๆในการสร้างเว็บ และเป็นเทคนิคที่สามารถสร้างสรรค์ชิ้นงานได้อย่างหลากหลายไม่มีขีดจำกัด

การใช้ Tool ของโปรแกรมโฟโตช้อป



รูปที่ 2.17 Tools ของโฟโตช้อป

การใช้ Gradient Tool จากโปรแกรม โฟโตช้อป คือการกำหนดโทนการไล่สีของภาพหรือของวัตถุ เพื่อให้ภาพดูมีลวดลายที่สวยงาม และทำให้วัตถุสีที่สวยงามตามความต้องการของการออกแบบ

การใช้ Select Tool เพื่อทำการเลือกบริเวณที่เราต้องการบนภาพ

การใช้ Move เพื่อทำการเคลื่อนย้ายสิ่งที่เราต้องการจัดวางตำแหน่งหรือ กำจัดออก

การใช้ Eraser เพื่อทำการลบส่วนที่ไม่ต้องการของภาพออก

การใช้ Line เพื่อสร้างรูปทรงเชิงเส้นที่เราต้องการ

การใช้ Type เพื่อพิมพ์ อักษร หรือข้อความลงบนภาพ

การใช้ Paint Bucket เพื่อทำการเทสีลงตรงบริเวณที่ต้องการ

การใช้ Eyedropper เพื่อค้นหาสีที่เราต้องการ

การใช้ Foreground Color เพื่อทำการเปลี่ยนสี

การใช้งาน History ของโปรแกรมโฟโตช้อป เพื่อสามารถย้อนกลับไปแก้ไขความผิดพลาดที่เกิดขึ้นแล้วได้

การทำงานกับเลเยอร์อาศัยเทคนิคการทำ Move Layer เป็นการเคลื่อนย้ายเลเยอร์หนึ่งไปไว้ที่ตำแหน่งใหม่

การทำ Duplicate Layer เป็นการสร้างเลเยอร์ขึ้นใหม่ที่เหมือนกับเลเยอร์เดิมไว้ที่ตำแหน่งบนของเลเยอร์เดิม

การทำ Layer Effects เป็นการเปลี่ยน รูปแบบของเลเยอร์ ใหม่ ขึ้นอยู่กับการเลือกรูปแบบนั้นๆ เช่น Drop Shadow , Bevel and Emboss

การเรียงลำดับ เลเยอร์ เป็นการกำหนด การแสดงของภาพว่าต้องการให้ส่วนใดปรากฏเป็นเช่นไร โดยการซ้อนกันของภาพที่อยู่ในแต่ละเลเยอร์

การทำ Merge Down เป็นการรวมเลเยอร์ที่อยู่ล่างเลเยอร์ปัจจุบันหนึ่งเลเยอร์

การสร้างภาพโดยแยกแต่ละส่วนของภาพให้อยู่ในแต่ละเลเยอร์จะทำให้ การจัดการเรื่องการทำ

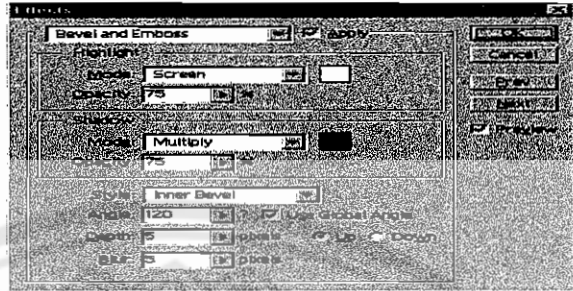
โลโอเวอร์ และการทำ ภาพเคลื่อนไหวใน โปรแกรม อิมเมจลิคิตีทำได้ง่าย

การใช้ Filter ของโปรแกรมโฟโตช้อป และเลเยอร์ Effects

การทำ Blur filter ทำภาพให้ดูนุ่มขึ้น

การทำ Reder Lens Flare และ Lighting Effects เป็นการใส่แสงลงบนภาพ

การทำ Layer Effect Bevel and Emboss เพื่อทำให้ภาพดูนุ่มขึ้นอย่างเป็นธรรมชาติ



รูปที่ 2.18 Bevel and Emboss

C/C++ programming

รูปที่ 2.19 ตัวอย่างการทำ Bevel and Emboss

มหาวิทยาลัยพระนคร