

14 หัวข้อแนวทางในการพัฒนาเว็บไซต์ให้ทุกคนเข้าถึงได้ จาก W3C

1. กำหนดคำอธิบายความหมายแทนภาพหรือกราฟิกที่ไม่ใช่ข้อความตัวอักษร

1.1) ข้อความอธิบายรูปภาพกราฟิกต่างๆในหน้าเว็บ ซึ่งจะต้องใช้ Attribute "alt" เพิ่มคำอธิบายความหมายของภาพ หรือ ใช้ Attribute "longdesc" เพื่อเป็นการเชื่อมโยงไปยังหน้าใหม่ในกรณีที่ต้องการอธิบายความหมายของภาพนั้นอย่างละเอียด

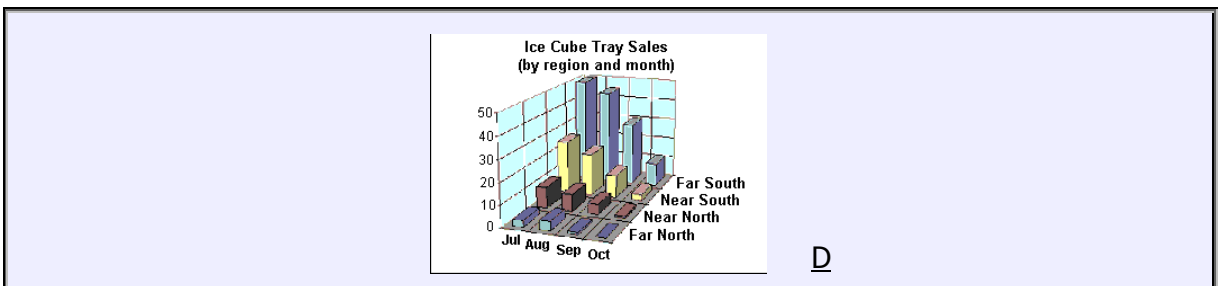
ตัวอย่าง :- ต้องการแสดงข้อมูลภาพกราฟิกบนเว็บไซต์ดังนี้



Code HTML

```

```



Code HTML

```

```

จะเห็นได้ว่า Attribute longdesc จะเป็นการเชื่อมโยงไปยัง ไฟล์ graph1.htm ซึ่งเป็นหน้าเว็บที่จะอธิบายความหมายอย่างละเอียดของกราฟ

1.2) การใช้ Image map ทางฝั่ง Server โดยต้องใช้ Attribute "ISMAP" (ปกติไม่ค่อยใช้) ถ้ามีการเชื่อมโยงไปยังส่วน A, B, C, D และ E ของเว็บไซต์คุณ ให้เพิ่มคำอธิบายสำหรับการเชื่อมโยงในแต่ละส่วนดังตัวอย่าง

ตัวอย่าง :- ต้องการแสดงภาพกราฟิก 1 ภาพ แต่ต้องการให้สามารถเชื่อมโยงไปได้ในหลายๆส่วนไปยังหน้าอื่น



Code HTML

```
<a href="img/imgmap1.map">
```

```
</a><br/>
```

[Section A | Section B | Section C | Section D | Section E]

1.3) การนำเสนอข้อมูลที่เรียกว่า Multimedia หรือ ข้อมูลที่มีทั้งภาพและเสียงนั้น ต้องเตรียมข้อมูล Text เพื่อให้โปรแกรมประเภทอ่านหน้าจอ สามารถอ่านข้อมูลได้ โดย W3C แนะนำควรมี 3 รูปแบบคือ Quicktime ของ Apple, SMIL (Synchronized Multimedia Integration Language) ของ W3C และ SAMI ของ Microsoft

1.4) การนำเสนอข้อมูลแบบ Multimedia จะต้องมีความสัมพันธ์ กันของบทบรรยายกับภาพและเสียง ซึ่งจะต้องตรงกัน เช่น ต้องการนำเสนอส่วนหนึ่งของบทภาพยนตร์ ต้องสร้างบทบรรยายที่เรียกว่า Caption เข้าไปในไฟล์ Vedio นั้นด้วย โดยใช้ในรูปแบบที่ W3C แนะนำ คือ Quicktime ,SMIL หรือ SAMI

1.5) การใช้ Image Map ทางฝั่ง Client โดยใช้ area ในการแบ่งออกเป็นส่วนของ การเชื่อมโยงในภาพ ต้องกำหนด "alt" ของการส่วนนั้นๆ ด้วย

ตัวอย่าง :-

การใช้ area ในการสร้าง ImageMap เพื่อเชื่อมโยงไปยังหน้าอื่น



Code HTML

```

<map name="map1">
  <area coords="0,0,39,39" href="a.htm" alt="เชื่อมโยงไปยัง section A">
  <area coords="40,0,79,39" href="b.htm" alt="เชื่อมโยงไปยัง section B ">
  <area coords="80,0,120,39" href="c.htm" alt="เชื่อมโยงไปยัง section C ">
  <area coords="121,0,160,39" href="d.htm" alt="เชื่อมโยงไปยัง section D ">
</map>
<map name="map2">
  [ <a href="a.htm">Section A</a>
  | <a href="b.htm">Section B</a>
  | <a href="c.htm">Section C</a>
  | <a href="d.htm">Section D</a> ]
</map>
```

2. การเลือกใช้สีในเว็บไซต์

2.1) การเลือกใช้สีของตัวหนังสือหรือข้อความที่ใช้อธิบายแสดงบนหน้าเว็บไซต์ ให้ใช้สีที่เห็นได้ชัดเจน

2.2) การเลือกใช้สีของพื้นหลังให้มีความแตกต่างจากสีของตัวอักษร ซึ่งถ้าเป็นภาพกราฟิกควรปรับความเข้มของแสงให้เหมาะสม แต่ถ้าเป็นตัวอักษร ควรใช้ Style Sheet ในการกำหนดสีตัวอักษร

ตัวอย่าง :-

สีตัวอักษร – ตัวอย่างที่ 1	สีตัวอักษร – ตัวอย่างที่ 2
แสดงรายการสินค้า	แสดงรายการสินค้า
<i>เครื่องพิมพ์</i>	<i>เครื่องพิมพ์</i>
Laser	Laser
Model 1 - 4 ppm - 2 bin	Model 1 - 4 ppm - 2 bin
Model 2 - 10 ppm - 3 bin	Model 2 - 10 ppm - 3 bin
Ink Jet	Ink Jet
<i>Scanners</i>	<i>Scanners</i>

จากตัวอย่างจะเห็นได้ว่า ตัวอย่างที่ 1 ทางด้านซ้ายแสดงผลข้อมูลได้ชัดเจนกว่า การแสดงผลข้อมูลทางด้านขวา เนื่องจาก สีของพื้น กับสีของตัวอักษรมีความแตกต่างกันอย่างชัดเจน ไม่เหมือนตัวอย่างที่ 2 ทางด้านขวามือ ซึ่งการแสดงผลของข้อมูล เห็นไม่ชัดเนื่องจากใช้ สีพื้นและสีตัวอักษร ใกล้เคียงกัน

3. เลือกใช้การ Markup ข้อความให้เหมาะสมกับงาน

3.1) ความเหมาะสมในการเลือกใช้ภาษาในการ Markup ซึ่งจะดีกว่าการนำเสนอจาก Image เช่น การใช้ ภาษา MathML ในการ Markup การนำเสนอข้อมูลเกี่ยวกับ สูตรทางคณิตศาสตร์ เป็นต้น

ตัวอย่าง :- ต้องการแสดงผลข้อมูลของสูตรทางคณิตศาสตร์ ในหน้าเว็บไซต์

$$\int_0^t \frac{dx}{x}$$

Code MathML

```
<semantics>
  <mrow>
    <msubsup>
      <mo>∫</mo>
      <mn>0</mn>
      <mi>t</mi>
    </msubsup>
    <mfrac>
      <mrow>
        <mo>d</mo>
        <mi>x</mi>
      </mrow>
      <mi>x</mi>
    </mfrac>
  </mrow>

<annotation-xml encoding="MathML-Content">
  <apply>
    <int/>
    <bvar><ci>x</ci></bvar>
```

```

<lowlimit><cn>0</cn></lowlimit>
<uplimit><ci>t</ci></uplimit>
<apply>
  <divide/>
  <cn>1</cn>
  <ci>x</ci>
</apply>
</apply>
</annotation-xml>
</semantics>

```

ข้อมูลของภาษา MathML สามารถดูได้จาก <http://www.w3.org/TR/REC-MathML/>

3.2) ความถูกต้องของเอกสารที่สร้าง ได้รูปแบบตามมาตรฐานของเอกสาร หรือไม่ กล่าวคือให้บอกว่า เอกสารที่สร้างนี้เป็นไปตามมาตรฐานและรุ่นใดของภาษาที่ใช้

Code HTML

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
หรือ
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

```

โดยที่ต้องประกาศไว้ที่ส่วนบนสุดในส่วนของ Code

3.3) ใช้ Style Sheet ในการกำหนดควบคุมการจัดตำแหน่งของรูปแบบตัวอักษร ขนาดตัวอักษร และ สีของตัวอักษรที่ต้องการนำเสนอ

ตัวอย่าง :- ต้องการกำหนดรูปแบบของตัวอักษรโดยใช้ Style Sheet ตามตัวอย่างนี้

หน้าที่ของการ Protection ในระบบคอมพิวเตอร์ คือ การเตรียมการทำงานสำหรับบังคับให้ใช้ ควบคุมดูแล ความฉลาด (Policy) ของการใช้ทรัพยากร

Code HTML

```

<STYLE TYPE="text/css">
  .para { font-size: 130%; color: #ff0000 }
  .linetwo { font-size: 70% ; font-style: italic ; color: #0000ff }
</STYLE>
<P CLASS= "para">หน้าที่ของการ Protection ในระบบคอมพิวเตอร์ คือ <SPAN CLASS=
"linetwo">การเตรียมการทำงานสำหรับบังคับให้ใช้ควบคุมดูแล ความฉลาด (Policy) ของการใช้
ทรัพยากร</SPAN></P>

```

3.4) การใช้หน่วยวัดของค่าต่างๆ ใน Style Sheet ให้กำหนดเป็นแบบที่สามารถแปรผันได้ตามขนาด ของ การแสดงผลของ Browser เช่น การใช้ค่าความยาวเป็น เปอร์เซนต์ แทนการใช้หน่วยเป็น 'pt' หรือ 'cm' ซึ่งจะทำให้เวลาที่ Browser ประมวลผล ทำให้ค่าที่ได้ เป็นไปตามสัดส่วนที่กำหนด ซึ่งทำให้ได้ผลที่ เหมือนกัน ในทุกๆ Browser

ตัวอย่าง :- ต้องการกำหนดขนาดของหัวข้อใน Stlye Sheet

ตัวอย่างหัวข้อใหญ่ :

Code HTML

```
<STYLE TYPE="text/css">
  h1{font-size: 150%}
</STYLE>
<h1>ตัวอย่างหัวข้อใหญ่ :</h1>
```

3.5) การสร้างหัวข้อ จากการใช้ Tag <h > ควรใช้ให้มีขนาดลดหลั่นกันลงมาตามลำดับ โดยหัวข้อใหญ่จะต้องอยู่ก่อนหัวข้อที่เล็กกว่า

ตัวอย่าง :- การกำหนดหัวข้อของเอกสารบนเว็บไซต์

```
ตัวอย่างหัวข้อใหญ่ : (<H1>)
  ตัวอย่างหัวข้อย่อยที่ 1 : (<H2>)
    ตัวอย่างหัวข้อย่อยที่ 1.1 (<H3>)
      ส่วนเนื้อหาในหัวข้อย่อย ซึ่งอาจจะใช้ <p> ในการทำ Markup ก็ได้
    ตัวอย่างหัวข้อย่อยที่ 1.2 (<H3>)
  ตัวอย่างหัวข้อย่อยที่ 2 : (<H2>)
    ตัวอย่างหัวข้อย่อยที่ 2.1 (<H3>)
```

Code HTML

```
<h1>ตัวอย่างหัวข้อใหญ่ : </h1>
<h2>ตัวอย่างหัวข้อย่อยที่ 1 : </h2>
<h3>ตัวอย่างหัวข้อย่อยที่ 1.1 </h3>
<p>ส่วนเนื้อหาในหัวข้อย่อย ซึ่งอาจจะใช้ Tag P ในการทำ Markup ก็ได้</p>
<h3>ตัวอย่างหัวข้อย่อยที่ 1.2 </h3>
<h2>ตัวอย่างหัวข้อย่อยที่ 2 : </h2>
<h3>ตัวอย่างหัวข้อย่อยที่ 2.1 </h3>
```

3.6) การสร้างรายการหัวข้อหรือ List Item ต่างๆให้ใช้ Tag ของ HTML เพื่อสะดวกในการ Navigate

ตัวอย่าง :- การสร้างรายการแบบมีลำดับให้กับข้อมูลในเว็บไซต์ดังนี้

```
อุปกรณ์ต่อพ่วงคอมพิวเตอร์มีดังนี้
1. เมาส์
2. คีย์บอร์ด
3. เครื่องพิมพ์
```

Code HTML

```
<ol>อุปกรณ์ต่อพ่วงคอมพิวเตอร์มีดังนี้
  <li>เมาส์</li>
  <li>คีย์บอร์ด</li>
  <li>เครื่องพิมพ์</li>
</ol>
```

3.7) การสร้าง Markup คำพูดของบุคคลที่อ้างถึงที่มีเครื่องหมายคำพูดให้ใช้ Tag <q>
ตัวอย่าง :-

เมื่อเข้านี้เค้าพูดว่า "สวัสดีครับ พ่อ แม่ พี่ น้อง"

Code HTML

```
<p>เมื่อเข้านี้เค้าพูดว่า  
  <q lang="en-th">สวัสดีครับ พ่อ แม่ พี่ น้อง</q>  
</p>
```

และถ้ามีการอ้างถึงคำพูดของบุคคลอื่นแต่ไม่มีเครื่องหมายคำพูดให้ใช้ Tag <quotation>
ตัวอย่าง

ประโยคแรกที่เค้าพูดคือ สวัสดีครับ พ่อ แม่ พี่ น้อง!

Code HTML

```
<p>ประโยคแรกที่เค้าพูดคือ  
  <quotation>สวัสดีครับ พ่อ แม่ พี่ น้อง!</quotation>  
</p>
```

4. การเลือกใช้ภาษาที่แสดงต้องบอกให้ชัดเจน

4.1) การใช้ภาษา ให้กำหนดภาษาที่ใช้ หรือในกรณีที่มีการสลับภาษาก็ต้องบอกโดยใช้ Attribute "lang=" "
ตัวอย่าง :-

เมื่อวานมีชาวต่างชาติพูดกับฉันว่า I love you.

Code HTML

```
<p>เมื่อวานมีชาวต่างชาติพูดกับฉันว่า<span lang="en">I love you</span></p>
```

4.2) การใช้คำย่อ ให้ใช้ abbr หรือ acronym และ Attribute "title" ในการบอกถึงชื่อเต็มของคำย่อ หรืออักษรย่อ นั้น ๆ ด้วย

ตัวอย่าง :- ต้องการอธิบายคำที่ใช้ในการย่อคำจะเป็นดังนี้

W3C

Code HTML

```
<acronym title="World Wide Web Consortium">W3C</acronym>
```

4.3) การระบุถึงภาษาหลักที่ใช้ในเอกสารนี้ โดยใช้ attribute lang ใน Tag <html>
ตัวอย่าง :- การกำหนดภาษาหลักที่ใช้ในเว็บไซต์

Code HTML

```
<html lang="th">
```

5. การสร้างตารางให้มีความเป็นระเบียบเรียบร้อย

5.1) การกำหนดข้อมูลในตาราง นั้นในแต่ละแถว แต่ละคอลัมน์ ต้องให้มีชื่อ Header ซึ่งต้องใช้ attribute headers ให้กับแถวหรือคอลัมน์นั้นๆ ด้วย เพราะโปรแกรมอ่านหน้าจอจะสามารถที่จะอ่านหัวข้อแล้วตามด้วยข้อมูลในคอลัมน์หรือแถวได้

	header ของ คอลัมน์ 1	header คอลัมน์ 2
header ของแถว 1	คอลัมน์ 1 แถว 1	คอลัมน์ 2 แถว 1
header ของแถว 2	คอลัมน์ 1 แถว 2	คอลัมน์ 2 แถว 2

ตัวอย่าง :- การกำหนดหัวข้อในแต่ละคอลัมน์

ชื่อ	อายุ	สถานที่เกิด	สถานที่ทำงาน
น้ำหนึ่ง	29	กรุงเทพฯ	เนคเทค
ยุทธการ	28	หนองคาย	เนคเทค

ซึ่งลักษณะการอ่านของโปรแกรมอ่านหน้าจอจะ เป็นดังนี้
ชื่อ น้ำหนึ่ง ,อายุ 29 , สถานที่เกิด กรุงเทพฯ , สถานที่ทำงาน เนคเทค
ชื่อ ยุทธการ, อายุ 28 , สถานที่เกิด หนองคาย, สถานที่ทำงาน เนคเทค

Code HTML

```
<tr>  
  <th id="t1">ชื่อ</th>  
  <th id="t2">อายุ</th>  
  <th id="t3" abbr="Type">สถานที่เกิด</th>  
  <th id="t4">สถานที่ทำงาน</th>  
</tr>  
<tr>  
  <td headers="t1">น้ำหนึ่ง</td>  
  <td headers="t2">29</td>  
  <td headers="t3">กรุงเทพฯ</td>  
  <td headers="t4">เนคเทค</td>  
</tr>  
<tr>  
  <td headers="t1">ยุทธการ</td>  
  <td headers="t2">28</td>  
  <td headers="t3">หนองคาย</td>  
  <td headers="t4">เนคเทค</td>  
</tr>
```

5.2) ข้อมูลในตารางที่มีระดับของหัวข้อมากกว่า 2 ระดับ ต้องให้มีความสัมพันธ์กันระหว่างข้อมูลและหัวข้อของแต่ละคอลัมน์ โดยใช้ Attribute (TBODY, THEAD, SCOPE, HEADERS, etc.)

ตัวอย่าง :-

ตาราง ค่าใช้จ่ายในการเดินทาง (บาท)

สถานที่, วันที่	ค่าอาหาร 3 มื้อ	ค่าที่พัก 1 คืน	ค่า ยานพาหนะ	รวมทั้งสิ้น
เชียงใหม่				
25ม.ค.46	100	1000	800	
26ม.ค.46	100	1000	700	
รวม	200	2000	1500	3700
เชียงราย				
27ม.ค.46	100	1200	1000	
28ม.ค.46	100	1200	500	
รวม	200	2400	1500	4100
รวมทั้งสิ้น	400	4400	3000	7800

จากตาราง ถ้าจะทำให้ โปรแกรมอ่านหน้าจ้ออ่านได้อย่างถูกต้อง โดยให้อ่านดังนี้
ตาราง ค่าใช้จ่ายในการเดินทาง (บาท)

สถานที่: เชียงใหม่ ,วันที่ :25ม.ค.46 ,ค่าอาหาร:100 ,ค่าที่พัก:1000 ,ค่าพาหนะ :800

สถานที่: เชียงใหม่ ,วันที่ :26ม.ค.46 ,ค่าอาหาร:100 ,ค่าที่พัก:1000 ,ค่าพาหนะ :700

สถานที่: เชียงใหม่ ,รวม ,ค่าอาหาร:200 ,ค่าที่พัก:2000 ,ค่าพาหนะ :1500 ,รวมทั้งสิ้น :3700

สถานที่: เชียงราย ,วันที่ :27ม.ค.46 ,ค่าอาหาร:100 ,ค่าที่พัก:1200 ,ค่าพาหนะ :1000

สถานที่: เชียงราย ,วันที่ :28ม.ค.46 ,ค่าอาหาร:100 ,ค่าที่พัก:1200 ,ค่าพาหนะ :500

สถานที่: เชียงราย ,รวม ,ค่าอาหาร:200 ,ค่าที่พัก:2400 ,ค่าพาหนะ :1500 ,รวมทั้งสิ้น :4100

สถานที่:รวมทั้งสิ้น:ค่าอาหาร:400, ค่าที่พัก:4400, ค่าพาหนะ:3000, รวมทั้งสิ้น:7800

Code HTML

```
<TABLE BORDER="1" CELLPADDING=2 CELLSPACING=3>
<CAPTION>ค่าใช้จ่ายในการเดินทาง (บาท)</CAPTION>
<THEAD>
<TR>
<TH><P><SPAN ID="t1-r1-l1">สถานที่</SPAN><BR/>
<SPAN ID="t1-r1-l2"> วันที่</SPAN></P></TH>
<TH SCOPE="column"><ABBR="ค่าอาหาร">ค่าอาหาร 3 มื้อ</ABBR></TH>
<TH SCOPE="column"><ABBR="ค่าที่พัก">ค่าที่พัก 1 คืน</ABBR></TH>
<TH SCOPE="column"><ABBR="ค่าพาหนะ">ค่ายานพาหนะ</ABBR></TH>
<TH SCOPE="column">รวม</TH>
</TR>
</THEAD>
<TBODY>
<TR>
<TH SCOPE="rowgroup" HEADERS="t1-r1-l1">เชียงใหม่</TH>
</TR>
<TR>
<TD SCOPE="row" HEADERS="t1-r1-l2"> 25ม.ค.46</TD>
<TD>100</TD>
<TD>1000</TD>
```



```

        <TD>800</TD>
    </TR>
    <TR>
        <TD SCOPE="row" HEADERS="t1-r1-l2"> 26ม.ค.46</TD>
        <TD>100</TD>
        <TD>1000</TD>
        <TD>700</TD>
    </TR>
    <TR>
        <TD SCOPE="row">รวม</TD>
        <TD>200</TD>
        <TD>2000</TD>
        <TD>1500</TD>
        <TD>3700</TD>
    </TR>
</TBODY>

<TBODY>
    <TR>
        <TH SCOPE="rowgroup" HEADERS="t1-r1-l1">เชียงใหม่</TH>
    </TR>
    <TR>
        <TD SCOPE="row" HEADERS="t1-r1-l2"> 27ม.ค.46</TD>
        <TD>100</TD>
        <TD>1200</TD>
        <TD>1000</TD>
    </TR>
    <TR>
        <TD SCOPE="row" HEADERS="t1-r1-l2"> 28ม.ค.46</TD>
        <TD>100</TD>
        <TD>1200</TD>
        <TD>500</TD>
    </TR>
    <TR>
        <TD SCOPE="row">รวม</TD>
        <TD>200</TD>
        <TD>2400</TD>
        <TD>1500</TD>
        <TD>4100</TD>
    </TR>
</TBODY>
<TBODY>
    <TR>
        <TH SCOPE="row">รวมทั้งสิ้น</TH>
        <TD>400</TD>
        <TD>4400</TD>
        <TD>3000</TD>
        <TD>7800</TD>
    </TR>
</TBODY>
</TABLE>

```

5.3) ไม่ควรใช้ตารางในการสร้างโครงร่าง (Layout) เว้นแต่ว่าต้องการใช้ตารางทำข้อมูลให้อยู่ในแนวเดียวกัน

5.4) ถ้ามีการใช้ตารางในการสร้างโครงร่าง Layout ก็อย่าไปใช้ Markup เกี่ยวกับโครงสร้างในการจัดรูปแบบของข้อมูล ซึ่งก็หมายถึงว่า ถ้าต้องการที่จะใช้ Table ในการจัดโครงสร้างก็ให้คิดว่า ควรจะใช้ Style Sheet (CSS) ในการจัดการรูปแบบข้อมูลจะดีกว่า

ตัวอย่าง :- กรณีที่ต้องการสร้างข้อมูลในตารางให้เป็นตัวหนาควรใช้ดังนี้

แสดงตัวอักษรแบบ ตัวหนา	16
-----------------------------------	----

Code HTML

```
<TD><DIV style="font-weight: bold">แสดงตัวอักษรแบบตัวหนา</DIV></TD>
```

ไม่ให้ใช้วิธีนี้

```
<TH>แสดงตัวอักษรแบบตัวหนา</TH>
```

เพราะการใช้ Tag <TH> ใน HTML จะหมายถึงการกำหนดชื่อหัวข้อของคอลัมน์ หรือแถวในตารางนั้นๆ

5.5) การใช้สร้างตารางในเว็บไซต์ควรจะใช้ Attribute Summary เพื่อใช้ในการอธิบายข้อมูลในตาราง ซึ่งโปรแกรมอ่านหน้าจอก็จะสามารถอ่านคำอธิบายความหมายของตารางนี้ได้ ซึ่งโดยปกติแล้ว Browser ทั่วไปจะไม่แสดงคำอธิบายนี้ให้เห็น

ตัวอย่าง :- การสร้างตารางดังตัวอย่างให้มีคำอธิบายความหมายข้อมูลในตาราง ได้ดังนี้

ชื่อ	อายุ	สถานที่เกิด	สถานที่ทำงาน
น้ำหนึ่ง	29	กรุงเทพฯ	เนคเทค
ยุทธการ	28	หนองคาย	เนคเทค

Code HTML

```
<TABLE border="1" summary="เป็นตารางที่แสดงชื่อ,อายุ,สถานที่เกิด และสถานที่ทำงานของพนักงาน">
```

```
... สร้างข้อมูลตาราง ...
```

```
</TABLE>
```

5.6) การใช้ Attribute "abbr" ในการสร้างคำย่อส่วนข้อมูลในหัวตาราง (Tag <TH>) ซึ่งใช้ประโยชน์ในการย่อคำ ทำให้ตัวอ่าน (Screen Reader) อ่านเฉพาะคำที่ย่อได้

ตัวอย่าง :- จากตัวอย่างตารางในข้อ 5.5 ถ้าต้องการให้ โปรแกรมอ่านหน้าจ้อ่านคำย่อก็จะได้ดังนี้

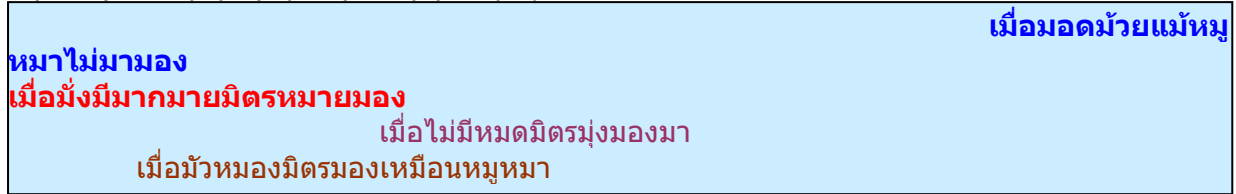
Code HTML

```
<TH id="t3" abbr="เกิด">สถานที่เกิด</TH>
```

เมื่อโปรแกรมอ่านหน้าจ้อ่านจะได้เป็น "เกิด" ซึ่งจะอ่านแทนคำว่า "สถานที่เกิด" จะทำให้ประหยัดเวลาในการอ่านเพราะยิ่งถ้าเป็นคำยาวเกินไป ก็จะไม่เสียเวลาในการรอให้โปรแกรมอ่านหน้าจ้อให้จบ ทั้งที่ผู้ใช้เข้าใจถึงความหมายจากคำย่อนั้นอยู่แล้ว (ระวังการย่อคำต้องให้ผู้ใช้เข้าใจตรงกันกับคำเต็มด้วย)

6. สำหรับเทคโนโลยีใหม่ๆ ที่ใช้แสดงบนหน้าเว็บไซต์ ต้องมีทางเลือกให้สำหรับกรณีที่ Browser ไม่สามารถแสดงผลได้ด้วย

6.1 การใช้ Style Sheet ในการจัดการกับข้อมูลตัวอักษร ต้องแน่ใจว่าถ้า Browser ไม่สนับสนุนเทคโนโลยี CSS ต้องสามารถแสดงผลข้อมูลได้ถูกต้องเหมือนเดิม โปรแกรมอ่านหน้าจอสามารถอ่านข้อมูลได้ด้วย
ตัวอย่าง :- การแสดงผลของข้อความต่อไปนี้



Code HTML

ในส่วนการกำหนด Style Sheet

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

```
<!--
```

```
.part1 /* เมื่อมั่งมีมากมายมิตรหมายมอง */ { color: red; font-size: 14pt;  
padding-left: 0; margin-top: 40px;  
font-family: copperplate gothic bold, fantasy, sans-serif }  
.part2 /* เมื่อมัวหมองมิตรมองเหมือนหมูหมา */ { color: brown; font-size: 10pt;  
padding-left: 100px; margin-top: 30px;  
font-family: times new roman, desdemona, serif }  
.part3 /* เมื่อไม่มีหมดมิตรมุ่งมองมา */ { color: purple; font-size: 18pt;  
padding-left: 200px; margin-top: -60px;  
font-family: desdemona, times new roman, serif }  
.part4 /* เมื่อมอดม้วยแม่หมูหมาไม่มามอง */ { color: blue; font-size: 24pt;  
padding-left: 350px;  
margin-top: -100px; margin-bottom: 100px;  
font-family: fantasy, copperplate gothic bold, sans-serif }
```

```
-->
```

```
</STYLE>
```

ในส่วน Body แสดงผลข้อมูล

```
<DIV class= "part1">เมื่อมั่งมีมากมายมิตรหมายมอง</DIV>
```

```
<DIV class= "part2">เมื่อมัวหมองมิตรมองเหมือนหมูหมา </DIV>
```

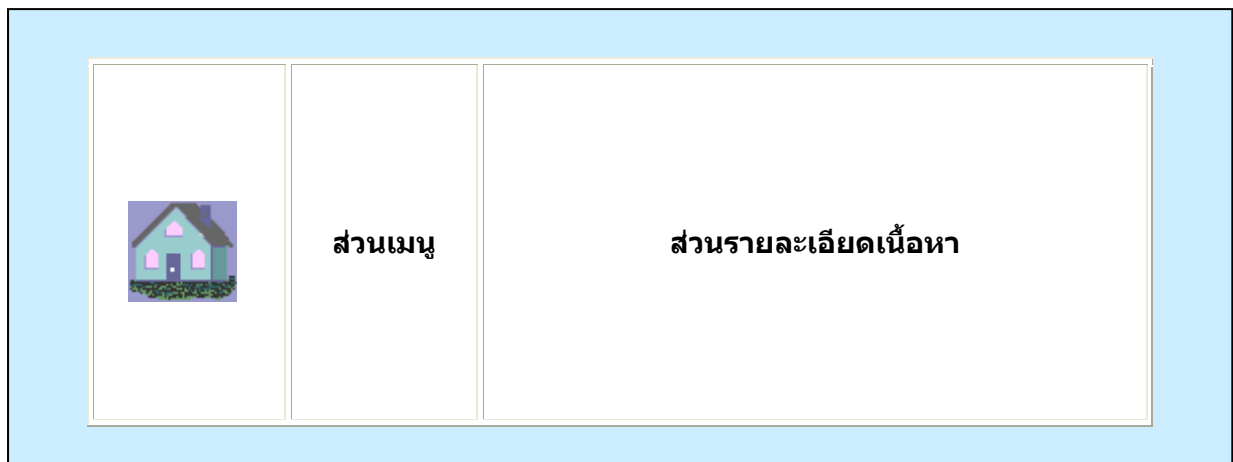
```
<DIV class= "part3">เมื่อไม่มีหมดมิตรมุ่งมองมา </DIV>
```

```
<DIV class= "part4">เมื่อมอดม้วยแม่หมูหมาไม่มามอง</DIV>
```

ซึ่งถ้า Browser ที่ไม่สนับสนุน Style Sheet แล้วก็ยังแสดงผลได้ด้วยข้อความเดิมได้ จะทำให้โปรแกรมอ่านหน้าจอยังสามารถอ่านข้อมูลได้เหมือนเดิม คือ

เมื่อมั่งมีมากมายมิตรหมายมอง
เมื่อมัวหมองมิตรมองเหมือนหมูหมา
เมื่อไม่มีหมดมิตรมุ่งมองมา
เมื่อมอดม้วยแม่หมูหมาไม่มามอง

6.2) ในการสร้าง Frame อย่าลืมการกำหนดคำอธิบายของภาพ ให้กับภาพที่ใส่ไว้ในไฟล์ที่อยู่ในแต่ละ Frame ด้วย
ตัวอย่าง :-



Code HTML

```
...
<FRAME name="picture_frame" src="home.html">
...
```

จะเห็นว่ามีการสร้าง Frame ออกเป็น 3 ส่วน คือส่วนที่เป็นรูปภาพ ส่วนเมนู และส่วนรายละเอียดเนื้อหา และที่สำคัญของข้อนี้คือ ในส่วนที่เป็นแสดงภาพกราฟิกนั้น ต้องระบุในไฟล์ home.html ด้วยว่า

```
<IMG src="home.gif" alt="ภาพแสดงรูปบ้าน">
```

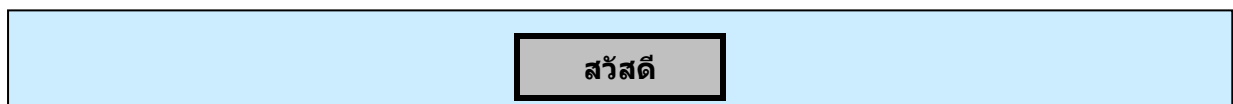
6.3) สำหรับการใส่ Script หรือ Applet บนเว็บไซต์ ต้องกำหนดคำอธิบายไว้ในกรณีที่ Script หรือ Applet ไม่สามารถ ทำงานได้ หรือ Browser ไม่สนับสนุนการทำงานของ Script หรือ Applet นั้นๆ และควรที่จะหลีกเลี่ยงการสร้าง link จากการใส่ javascript แทนการใช้แสดง URL ตรงๆ

ตัวอย่างที่ไม่ควรทำ :-

```
<A href="javascript:">...</A>
```

6.4) การใช้ Script หรือ Applet ในการรับค่าข้อมูล Input จาก Event ต่างๆ ก็ต้องแน่ใจว่าต้องเป็น Event ที่รับจาก อุปกรณ์ในการ Input นั้นๆได้ และถ้าไม่สามารถใช้ Event นั้นๆได้ต้องมี Tag <Noscript> แสดงคำอธิบายแทนการใช้ Event นั้นๆ

ตัวอย่าง :- มีปุ่ม สวัสดี ต้องการรับค่าข้อมูล Input จาก Mouse ซึ่งจะทำให้การประมวลผลเมื่อ ปุ่มนี้ถูก Focus และ ใช้ Event ของ Mouse "OnMouseOver" ก็จะปรากฏ กล่องข้อความว่า "สวัสดี"



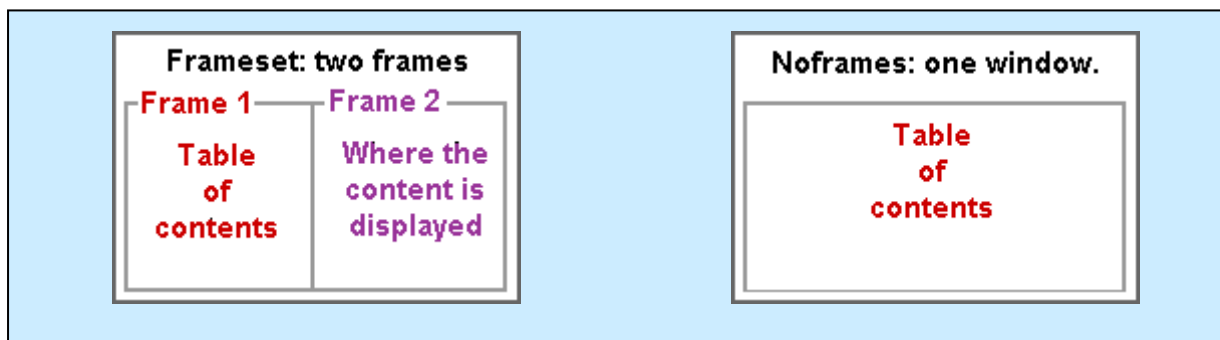
และถ้า Browser ไม่สนับสนุน Script นี้ เมื่อเรามี Tag <Noscript> อธิบายก็จะปรากฏข้อความขึ้นเหมือนกัน

Code HTML

```
<BUTTON ONMOUSEOVER="pop()"> สวัสดี</BUTTON>.
<SCRIPT language="JavaScript" type="text/javascript">
function pop() { alert ("สวัสดี") }
</SCRIPT>
<NOSCRIPT>สวัสดี</NOSCRIPT>
```

6.5) ในส่วนที่เป็น Dynamic Content ที่ต้องมีการเปลี่ยนแปลงของข้อมูลนั้น ได้อำนวยความสะดวกให้กับคนพิการได้สามารถเข้าถึงข้อมูลข่าวสารได้หรือไม่

ตัวอย่าง :- กรณีใช้ Frame ต้องใช้ NoFrame ในการอธิบายข้อมูลถ้า Browser ไม่สนับสนุน Frame



Code HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<HTML>
<HEAD>
<TITLE>ไฟล์เริ่มต้น.html</TITLE>
</HEAD>
<FRAMESET cols="50%, 50%" title="ข้อมูลเอกสารทั้งหมด">
<FRAME src="main.html" title="ส่วนเมนูหลัก">
<FRAME src="table_of_contents.html" title="ส่วนข้อมูลรายละเอียด">
<NOFRAMES>
<A href="table_of_contents.html">ส่วนข้อมูลรายละเอียด</A>
<!-- เป็นการแสดงข้อมูลในหน้าข้อมูลรายละเอียด -->
</NOFRAMES>
</FRAMESET>
</HTML>
```

จากตัวอย่างจะเห็นได้ว่าถ้า Browser ไม่สนับสนุน Frame ก็ยังสามารถแสดงข้อมูลได้อย่างถูกต้อง กล่าวคือ จะยังสามารถแสดงข้อมูลรายละเอียดของ ไฟล์ table_of_contents.html ได้

ตัวอย่าง :- กรณีใช้ Script ต้องใช้ Noscript ในการอธิบายข้อมูลถ้า Browser ไม่สนับสนุน Script

Code HTML

```
<SCRIPT type="text/tcl">
... เขียนสคริปต์แสดงผลคะแนน กีฬา...
</SCRIPT>
<NOSCRIPT>
<P>รายงานผลตารางคะแนนนี้ล่าสุด</P>
<DL>
<DT>Chelsea 36, Arsenal 35, ManchesterUnited 34.
<DD><A href="Scoretoday.html">ผลการแข่งขัน Chelsea vs. ManchesterUnited </A>
...ผลคู่อื่น... </DL>
</NOSCRIPT>
```

7. ผู้ใช้สามารถควบคุมการเคลื่อนที่ได้

7.1) ส่วนที่เป็นการแสดงจากพื้นกระพริบนั้น สามารถให้ผู้ใช้ควบคุมการกระพริบนั้นได้ กล่าวคือ สามารถควบคุมไม่ให้กระพริบหรือหยุดการกระพริบไปเลย บางคนซึ่งมีปฏิกิริยากับความไวของแสง อาจทำให้เกิดโรคมบางอย่างขึ้น เช่น โรคลมบ้าหมู เป็นต้น ช่วงเวลาการกระพริบของแสงหรือการเคลื่อนไหวนั้นๆปกติจะอยู่ที่ 4-59 ครั้งต่อวินาที (Hz) และค่าการกระพริบที่ดีที่สุดจะอยู่ที่ 20 ครั้งต่อวินาที(Hz)

7.2) ส่วนที่แสดงการกระพริบของตัวอักษร สามารถให้ผู้ใช้ควบคุมการกระพริบนั้นได้ หรือหยุดการกระพริบได้ ถ้าต้องการเน้นค่าโดยต้องการให้ตัวอักษรนั้นกระพริบ ควรใช้ CSS ,**'text-decoration: blink' attribute.** ไม่ให้ใช้ Tag <Blink> และ <MARQUEE>

7.3) การใช้ Script หรือ Applet ที่มีการเขียนให้เกิดมีการเคลื่อนที่หรือเคลื่อนไหวอะไรได้ในหน้าเว็บ ควรต้องมีกลไกในการควบคุมให้ผู้ใช้สามารถหยุดการเคลื่อนไหวหรือเคลื่อนที่ของหน้าเว็บนั้นๆ ได้

7.4) การ Refresh ของหน้าเว็บซึ่งจะต้องมีการโหลดใหม่ตามเวลาที่กำหนดนั้น ผู้ใช้สามารถควบคุมให้หยุดการกระทำนั้นได้ ซึ่งการกระทำดังกล่าวเป็นการ Update ข้อมูลขึ้นมาใหม่ในหน้าเว็บนั้นๆ ซึ่งก็เป็นหน้าที่ ที่ผู้ใช้ควรจะเป็นคน Reload ข้อมูลเอง โดยที่จะมีการเตือนจากเว็บไซต์ก็ได้

7.5) ไม่ทำการเรียกหน้าเว็บไซต์ขึ้นมา แล้วมีการไปเรียกไปยังหน้าเว็บใหม่โดยที่ผู้ใช้ไม่รู้ URL ซึ่งเป็นการเรียกหน้าใหม่โดยอัตโนมัติ

<META http-equiv="refresh" content="5;http://....." >

ซึ่งวิธีนี้ไม่ควรใช้เพราะจะทำให้ผู้ใช้สับสน และการทำวิธีนี้ไม่ใช้มาตรฐานของการสร้างเว็บไซต์ด้วยในการเก็บค่า history ของ Browser ก็มีผลทำให้ผิดพลาดเหมือนกัน

8. การรองรับความสามารถ ในการเข้าถึงข้อมูลได้จาก User interface

8.1) การเขียนโปรแกรมด้วย Script หรือ Applet นั้นๆ ให้พิจารณาด้วยว่าคนพิการสามารถเข้าถึงหรือใช้ได้กับเทคโนโลยีของคนพิการหรือไม่ ซึ่งถ้าเป็น ฟังก์ชันพื้นฐานของอุปกรณ์ Input ที่สำคัญๆ ก็ต้องให้คนพิการเข้าถึงได้เหมือนกัน

ตัวอย่าง :- ในการเขียน Script โดยให้มีการรับค่าจากการใช้ Mouse ซึ่งจะใช้ Event ของ Mouse จำเป็นต้องเพิ่มการรับค่าจากการใช้ Event ของ keyboard ด้วย

Event Mouse ใช้ "OnMouseDown" ใน Event Keyboard ใช้ "OnKeyDown"

Event Mouse ใช้ "OnMouseUp" ใน Event Keyboard ใช้ "OnKeyUp"

Event Mouse ใช้ "OnClick" ใน Event Keyboard ใช้ "OnKeyPress"

เป็นต้น

ที่สำคัญอย่าใช้ Event ของ Mouse ในการรับค่า Input ในทางเดียว ต้องใช้ Event ของ Keyboard ในการรับค่าควบคู่กันไปด้วย ซึ่งต้องเทียบลักษณะการรับค่าจาก Event ให้มีลักษณะที่ใกล้เคียงกันด้วย

9. การออกแบบอุปกรณ์ที่ใช้เฉพาะกับบุคคล

9.1) การใช้ Image map ทางฝั่ง Client แทนการใช้จากทางฝั่ง Server ในกรณีที่บริเวณของภาพกราฟิกนั้นไม่สามารถที่จะใช้การกำหนดการเชื่อมโยงเป็นรูปทรงเรขาคณิตได้

ตัวอย่าง :- ต้องการสร้าง link ใน Object เป็นแบบ image map

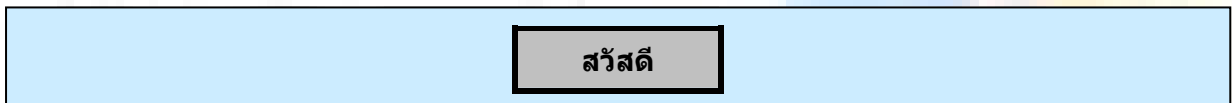


Code HTML

```
<OBJECT data="imgmap4.gif" type="image/gif" usemap="#map1">
  <MAP name="map1">
    <P>Navigate the site. </P>
    <A href="a.htm" shape="rect" coords="0,0,39,39">[Section A]</A>
    <A href="b.htm" shape="rect" coords="40,0,79,39">[Section B]</A>
    <A href="c.htm" shape="rect" coords="80,0,120,39">[Section C]</A>
    <A href="d.htm" shape="rect" coords="121,0,160,39">[Section D]</A>
  </MAP>
</OBJECT>
```

9.2) ในการออกแบบ Interface ต้องแน่ใจว่าทุกๆ Element นั้นสามารถที่จะให้ Keyboard ควบคุม การเข้าถึงข้อมูลได้เหมือนกับใช้ Mouse ซึ่งอาจจะเป็น การใช้ Keyboard Shortcut ในการเลือก หรือจะเป็น การกำหนด Tab order ในการควบคุมการโฟกัสของ Control ต่างๆ โดยการกด Tab เพื่อให้เลือกไปที่ละ Control

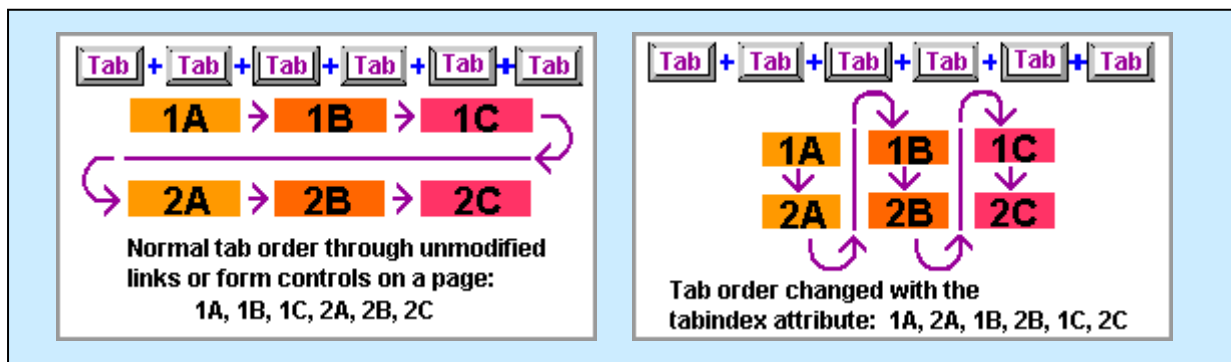
9.3) ในการกำหนดมีการรับ Input จาก Event ของ Mouse ก็ให้มีการรับ Input จาก Event ของ Keyboard ที่มีลักษณะคล้ายๆกันด้วยซึ่งก็จะเป็นดังนี้ "OnMouseDown" ใช้กับ "OnKeyDown", "OnMouseUp" ใช้กับ "OnKeyUp" และ "OnClick" ใช้กับ "OnKeyPress" เป็นต้น ตัวอย่าง :- สร้างปุ่มโดยใช้ Javascript เพื่อรับค่า Input จาก Mouse และ Keyboard



Code HTML

```
<BUTTON ONCLICK="pop()" ONKEYPRESS="pop()"> สวัสดี</BUTTON>.
<SCRIPT language="JavaScript" type="text/javascript">
  function pop() { alert ("สวัสดี") }
</SCRIPT>
<NOSCRIPT>สวัสดี</NOSCRIPT>
```

9.4) ในการกำหนดลำดับของ Tab ซึ่งใช้ในการโฟกัส ตัว Control หรือ Object ต่างๆ โดยให้ เรียงลำดับของ tabindex ให้เหมาะสม ซึ่งการกำหนดเป็น Attribute Tabindex จะใช้ได้กับ Tag <A> <AREA> <BUTTON> <INPUT> <SELECT> และ <TEXTAREA>



ภาพซ้ายมือ เป็นจำลองการเกิดลำดับของ Tab ที่เกิดโดยปกติ ส่วนภาพทางด้านขวามือ จะเป็นการจำลอง การกำหนดลำดับของ Tab ใหม่โดยใช้ Attribute Tabindex

ตัวอย่าง :- การใช้ Attribute Tabindex กำหนดลำดับการ Tab

(6th tab stop)
คุณชอบสีอะไร ?
แดง (3rd tab stop)
เขียว (4th tab stop)
น้ำเงิน (5th tab stop)
ชื่อบริษัท (2nd tab stop)
 (7th tab stop)

[Skip the form to find accessible alternatives.](#) (1st tab stop)

Code HTML

```
<FORM METHOD="POST">  
  <INPUT TYPE="SUBMIT" VALUE="Enter" tabindex="6">คุณชอบสีอะไร ?<BR>  
  แดง <INPUT TYPE="CHECKBOX" NAME="CBVar1" VALUE="CB4"  
  tabindex="3"><BR>  
  เขียว <INPUT TYPE="CHECKBOX" NAME="CBVar2" VALUE="CB5"  
  tabindex="4"><BR>  
  น้ำเงิน <INPUT TYPE="CHECKBOX" NAME="CBVar3" VALUE="CkB6"  
  tabindex="5"><BR>  
  ชื่อบริษัท <INPUT TYPE="TEXT" NAME="Com" tabindex="2"><BR>  
  <INPUT TYPE="RESET" VALUE="Undo" tabindex="7">  
  <A HREF="form-alt.htm" tabindex="1">Skip the form to find accessible  
  alternatives</A>.<BR>  
</FORM>
```

และในความเป็นจริงแล้ว การแสดงผลของข้อมูลก็ต้อง เรียงลำดับไว้ด้วยดังนี้

[Skip the form to find accessible alternatives.](#) (1st tab stop)

ชื่อบริษัท (2nd tab stop)
คุณชอบสีอะไร ?
แดง (3rd tab stop)
เขียว (4th tab stop)
น้ำเงิน (5th tab stop)
 (6th tab stop)
 (7th tab stop)

9.5) การจัดเตรียม Keyboard Shortcut (คีย์ลัด) สำหรับการโฟกัสไปยังตัว Control นั้นๆ

ตัวอย่าง :- การกำหนด Shortkey ให้กับตัว Control ใน Form ซึ่งจะกำหนดให้ดังนี้

a – Skip then form to find accessible alternatives

u – Undo
s – Submit
n – ชื่อบริษัท
r,g,b – สีแดง, สีเขียว, สีน้ำเงิน

[Skip the form to find accessible alternatives. \(1st tab stop\)](#)

ชื่อบริษัท (2nd tab stop)

คุณชอบสีอะไร ?

แดง (3rd tab stop)

เขียว (4th tab stop)

น้ำเงิน (5th tab stop)

(6th tab stop)

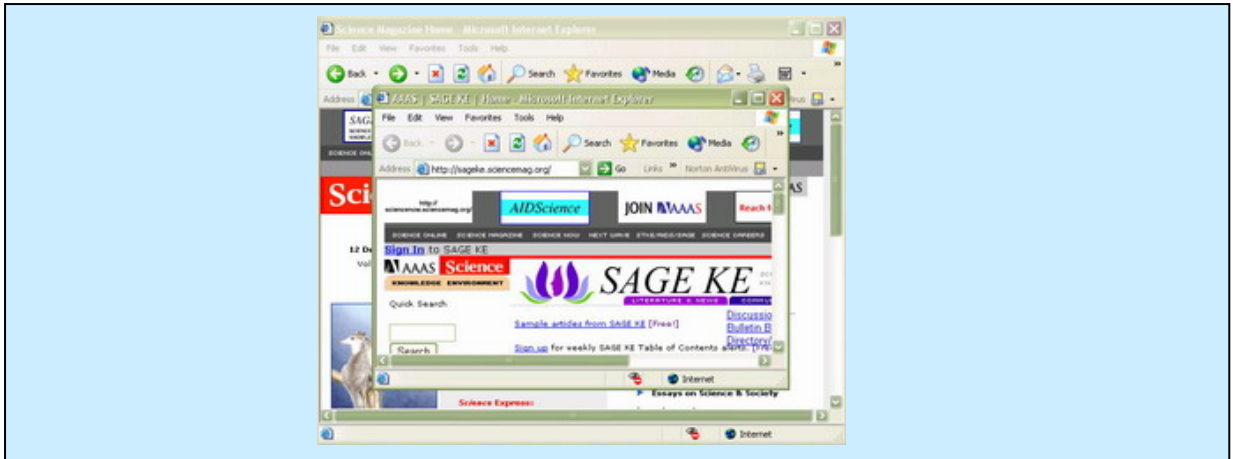
(7th tab stop)

Code HTML

```
<FORM METHOD="POST">  
  <A HREF="form-alt.htm" tabindex="1" ACCESSKEY=a>Skip the form to find  
accessible  
alternatives</A><BR>  
  ชื่อบริษัท <INPUT TYPE="TEXT" NAME="Com" tabindex="2" ACCESSKEY=n><BR>  
  คุณชอบสีอะไร ?<BR/>  
  แดง <INPUT TYPE="CHECKBOX" NAME="CBVar1" VALUE="CB4"  
  tabindex="3" ACCESSKEY=r><BR/>  
  เขียว <INPUT TYPE="CHECKBOX" NAME="CBVar2" VALUE="CB5"  
  tabindex="4" ACCESSKEY=g><BR/>  
  น้ำเงิน <INPUT TYPE="CHECKBOX" NAME="CBVar3" VALUE="Cb6"  
  tabindex="5" ACCESSKEY=b><BR/>  
  <INPUT TYPE="SUBMIT" VALUE="Enter" tabindex="6" ACCESSKEY=s>  
  <INPUT TYPE="UNDO" VALUE="Undo" tabindex="7" ACCESSKEY=u>  
</FORM>
```

10. ในช่วงเวลาระหว่างการรอการทำงานต่อ

10.1) การเชื่อมโยงไปยังหน้าใหม่จะต้องไม่เปิดวินโดวส์ใหม่ขึ้นมา เพราะผู้ใช้จะสับสนถ้าหากเปิดหน้าใหม่โดย มีการเปิด windows ใหม่ของ Browser ขึ้นมา เนื่องจากปกติผู้ใช้จะใช้วิธีการเลือกปุ่ม Back หรือ Forward จาก Browser ในกรณีที่ต้องการกลับไปยังหน้าเดิมที่เคยเปิดมาแล้ว หรือไปยังหน้าต่อไปจากที่เคยเปิดมาแล้ว ซึ่ง Browesr จะเก็บข้อมูลไว้ใน Cache เป็น History อยู่แล้ว และการที่เปิด Windows ใหม่ขึ้นมาจะทำให้ผู้ใช้ไม่สามารถกลับไปยังหน้าเดิมได้ หรือกระทำไต่ยากกว่าเดิม
ตัวอย่าง :- แสดงการเปิดหน้าใหม่ โดยเปิด Windows ใหม่ ซึ่งลักษณะนี้ไม่ควรทำ เนื่องจากการ Focus ของ windows จะถูกเปลี่ยนไปทำให้การค้นหา หรือ การ Navigate ผิดพลาดได้



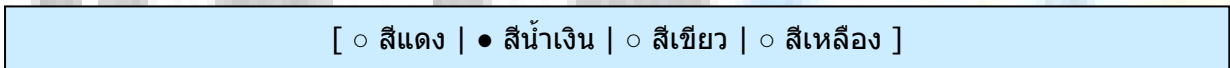
Code HTML

เปิดหน้าต่างใหม่

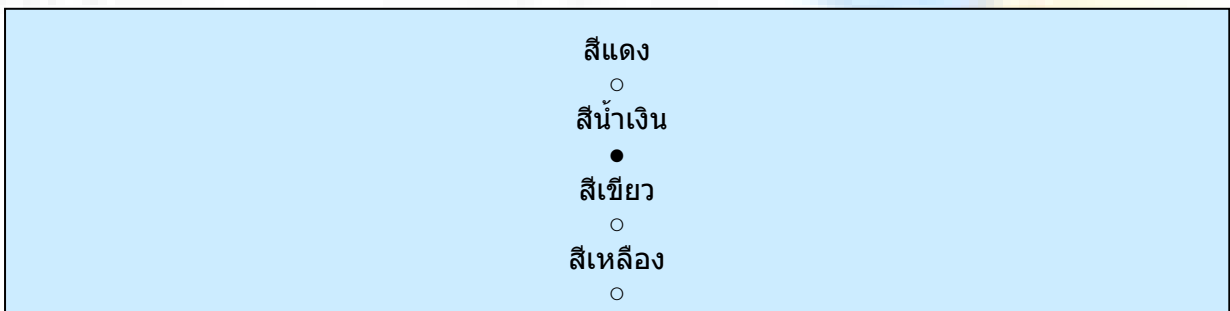
ลักษณะอย่างนี้ของ CodeHTML จะทำให้การเชื่อมโยงไปยังหน้าต่างใหม่นั้น ได้ทำการเปิด Windows ใหม่เพราะการใช้ Attribute TARGET = "_blank" ดังนั้นการเปิดหน้าต่างของการเชื่อมโยงใหม่ที่ถูกต้องการที่จะเปิดไว้ใน Window เดิม โดยให้ใช้ TARGET= "_self" หรือ TARGET= "ชื่อเดิมของ Window"

10.2) ในการใช้ Control ต่างๆ ใน Form ต้องให้ Label กับตัว Control มีความสัมพันธ์กัน เพื่อจะทำให้ผู้ใช้ไม่สับสน เพราะในการใช้โปรแกรมอ่านหน้าจอ อ่านข้อมูลในหน้าเว็บ ทำให้การอ่านข้อมูลจะอ่านไม่ตรงกันเกิดความสับสนได้

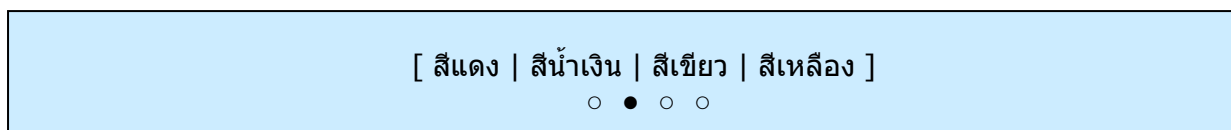
ตัวอย่างที่ 1 จะแสดงให้เห็นถึงการ ใช้ Control กับ Label ที่อยู่ในบรรทัดเดียวกัน



ตัวอย่างที่ 2 จะแสดงให้เห็นถึงการ ใช้ Control กับ Label ที่ต้องแสดงให้อยู่ต่างบรรทัดกัน ซึ่งก็ยังคงคำนึงถึงการอ่านของโปรแกรมอ่านหน้าจอ เหมือนกัน เพื่อไม่ให้ผู้ใช้เกิดความสับสน



ตัวอย่างที่ 3 จะแสดงให้เห็นถึง วิธีที่ไม่ควรใช้ในการใช้ Control



ซึ่งวิธีนี้จะทำให้โปรแกรมอ่านหน้าจออ่าน Label จนจบบรรทัดก่อนที่จะมาอ่าน ค่าของ Control จึงทำให้ผู้ใช้เกิดความสับสนได้

10.3) การตารางในการจัดข้อมูล ควรจัดให้ข้อความที่ต้องการให้โปรแกรมอ่านหน้าจอ อ่านต่อกันในเรื่องเดียวกันนั้น ควรให้มีการจัดอยู่ในคอลัมน์เดียวกัน เพราะโดยปกติของ ตัวอ่านหน้าจอ จะอ่านจากแถวเดียวกันก่อนไปทีละคอลัมน์ในแถวนั้นๆ แล้วค่อยไปเริ่มอ่านที่แถวใหม่ ในคอลัมน์แรก ต่อไป ดังนั้นจึงควรที่จะทำข้อความที่ต้องการอ่านต่อกันไป ให้อยู่ในคอลัมน์เดียวกัน

ตัวอย่าง

บทที่ 1	บทที่ 2	บทที่ 3
การสร้างตาราง.....	การสร้างเฟรม.....	การสร้างฟอร์ม.....
.....
.....
.....

การจัดข้อมูลในลักษณะนี้จะทำให้โปรแกรมอ่านหน้าจออ่านได้ดังนี้
 บทที่ 1 การสร้างตาราง..... , บทที่ 2 การสร้างเฟรม , บทที่ 3 การสร้างฟอร์ม.....
 ซึ่งถ้าเรากำหนดค่าในตารางเป็นแบบนี้

บทที่ 1	บทที่ 2	บทที่ 3
การสร้างตาราง.....	การสร้างเฟรม.....	การสร้างฟอร์ม.....
.....
.....
.....

การอ่านค่าข้อมูลในตารางจะเป็นดังนี้
 บทที่ 1 , บทที่ 2 , บทที่ 3, การสร้างตาราง , การสร้างเฟรม..... , การสร้างฟอร์ม
 ซึ่งทำให้ไม่ตรงกับข้อมูลที่ต้องการแสดง เพราะการลำดับของการอ่านเปลี่ยนไป

10.4) การสร้าง Form ที่มีการใช้ Control สำหรับการเติมข้อความในช่องว่าง ให้กำหนดค่าเริ่มต้นให้กับตัวควบคุมนั้นด้วย ไม่ควรเป็นช่องว่างเปล่าๆ

ตัวอย่าง

The image shows a web form with the following elements: a text input field labeled 'Your name' with an asterisk, a dropdown menu with the text 'ให้กรอกข้อมูลที่นี่', a label 'Feedback' next to the dropdown, and a green button labeled 'Submit this form.'

Code HTML

```
Your name <INPUT TYPE="TEXT" NAME="Name" SIZE="50" VALUE="* ">
Feedback <TEXTAREA NAME="TextArea1" ROWS="4" COLS="50">
ให้กรอกข้อมูลที่นี่ </TEXTAREA>
<INPUT TYPE="submit" VALUE="Submit this form.">
```

จากตัวอย่างจะเห็นว่า ในช่องของ Your name มีค่าเริ่มต้นเป็น "*" และในช่องของ Feedback มีค่าเริ่มต้นเป็น "ให้กรอกข้อมูลที่นี่" ซึ่งการกำหนดให้มีค่าเริ่มต้น เพราะมี Browser บางตัวไม่สามารถใช้ TAB จาก Keyboard เพื่อให้ Browser ที่ไม่สนับสนุนการใช้ TAB จากคีย์บอร์ด ในการนำไปสู่ของการกรอกข้อมูล สามารถให้โปรแกรมอ่านหน้าจออ่านแล้วผู้ใช้เข้าใจได้

10.5) การจัดทำมีการเชื่อมโยง (Link) อยู่ติด หรือใกล้ๆกัน ควรจะให้ มีสัญลักษณ์ หรืออะไรบางอย่าง แบ่ง ระหว่าง การเชื่อมโยง ทั้งสองนั้น ตัวอย่างเช่น เพิ่มตัวอักษร " | "(Pipe) และ ช่องว่าง เพื่อแบ่งให้ผู้ใช้รู้ว่ามีการเชื่อมโยงมากกว่าหนึ่ง

ตัวอย่าง

[[Section A](#) | [Section B](#) | [Section C](#) | [Section D](#) | [Section E](#)]

Code HTML

```
[ <A HREF="a.htm">Section A</A> | <A HREF="b.htm">Section B</A> | <A HREF="c.htm">Section C</A> | <A HREF="d.htm">Section D</A> | <A HREF="e.htm">Section E</A> ]
```

11. ใช้เทคโนโลยีและทำตามคำแนะนำของ W3C

11.1) การใช้เทคโนโลยีของ W3C ให้เหมาะสมกับงาน และใช้ใน Version สุดท้ายที่ Browser สนับสนุนการใช้งานซึ่งมีดังนี้

MathML ให้ใช้สำหรับการเขียนสูตรทางคณิตศาสตร์

HTML, XHTML, XML ให้ใช้สำหรับงานด้านโครงสร้างเอกสารการนำเสนอ

RDF ให้ใช้กับงานโครงสร้างข้อมูล

SMIL เหมาะสำหรับงานประเภทการนำเสนอในรูปแบบ Multimedia

CSS and XSL ให้ใช้ในการกำหนด Style Sheet

XSLT ใช้ในการสร้างการปรับเปลี่ยนลำดับโครงสร้างข้อมูลหรือเอกสาร และ PNG เหมาะสำหรับงาน

ด้านกราฟิก

11.2) หลีกเลี่ยงการใช้ Element และ Attribute ที่ไม่สนับสนุนความสามารถด้าน Accessible

ตัวอย่าง

Element ที่ทำหน้าที่เกี่ยวกับโครงสร้าง

- ประเภทแสดงรายการเช่น DIR และ Attribute ที่อยู่ใน Element ประเภทแสดงรายการนี้ คือ start และ value
- ประเภทการเชื่อมโยง เช่น APPLET
- ประเภทฟอร์มและตัวควบคุมคีย์บอร์ด เช่น ISINDEX

Element ที่ทำหน้าที่เกี่ยวกับการแสดงผลของตัวอักษร

- ประเภทการจัดหน้า เช่น CENTER และ Attribute ที่ส่วนมากใช้คือ align, valign, clear, nowrap
- ประเภทการเว้นช่องว่าง เช่น Attribute hspace, vspace, compact, type
- ประเภทรูปแบบตัวอักษร เช่น Element STRIKE, S, U, FONT, BASEFONT และ Attribute face, size
- ประเภทการกำหนดสีทั้งหมด เช่น Attribute background, bgcolor, color, text, link, alink, vlink
- การกำหนด Attribute noshade ของ Element HR

11.3) สำหรับเอกสารที่แสดงในหน้าเว็บ ที่มีหลายภาษาในการแสดง ต้องมีวิธีการที่จะบอกให้ผู้ใช้ทราบ โดยการเพิ่มคำอธิบายให้กับเอกสารเหล่านั้นนั้นข้อความที่บอกถึง การเลือกภาษาของเอกสารใน

หน้าเว็บได้ โดยอาจจะมีการเชื่อมโยงไปยังเอกสารหน้าใหม่ที่เป็นอีกภาษาหนึ่งก็ได้ และใน HTML4.0 จะมี Attribute "lang = " เพื่อใช้ในการบอกให้รู้ว่าขณะนี้ใช้ภาษาหลักใดในการอ่าน หรือ การใช้ Attribute hreflang กับ Element <A> และ <LINK> ในการบอกให้โปรแกรมอ่านหน้าจอรู้อาษาที่จะใช้อ่านคือ ภาษาใด

Code HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML lang="fr">
<HEAD>
<TITLE>Un document multilingue</TITLE>
</HEAD>
<BODY>
...Interpreted as French...
<P lang="es">...Interpreted as Spanish...
<P>...Interpreted as French again...
<P>...French text interrupted by<EM lang="ja">some
Japanese</EM>French begins here again...
</BODY>
</HTML>
```

11.4) สดท้ายแล้วหากพยายามถึงที่สุดแล้วยังไม่สามารถที่จะทำให้หน้าเว็บนั้นสามารถให้คนพิการเข้าถึงได้ ก็ต้องทำการเพิ่มหน้าเว็บนี้ใหม่โดยใช้เทคโนโลยีตาม W3C ที่แนะนำเพื่อเป็นทางเลือกให้กับคนพิการ ให้สามารถเข้าถึงข้อมูลได้ ซึ่งจะมีเนื้อหาที่เหมือนกันกับแบบเดิม

12. การกำหนดความหมายให้กับหัวข้อต่างๆ

12.1) การตั้งชื่อให้กับเฟรมในแต่ละเฟรม โดยใช้ Attribute title เพื่อให้สะดวกในการเข้าใจถึงความหมายของเฟรมได้ง่าย

ตัวอย่าง แสดงเว็บไซต์ที่ใช้เฟรมในการออกแบบและจัดข้อมูลเอกสาร

เฟรมแสดง Banner ส่วนหัวของเว็บไซต์	
เฟรมแสดง สารบัญ หัวข้อที่1 หัวข้อที่2 หัวข้อที่3	เฟรมแสดงข้อมูลของรายละเอียดจากสารบัญ
เฟรมแสดง ส่วนท้ายของเว็บไซต์	

จากตัวอย่าง จะเห็นได้ว่าในตัวอย่างแบ่งหน้าเว็บไซต์ออกเป็น 4 เฟรม โดยแต่ละเฟรมจะทำหน้าที่แสดงข้อมูลแตกต่างกันไป ซึ่งก็ต้องบอกให้ทราบว่าแต่ละเฟรมคืออะไร

Code HTML

```
<FRAMESET ROWS="20%,*,30%">
<FRAME SRC="f1.htm" title=" Banner ส่วนหัวของเว็บไซต์">
<FRAMESET COLS="20%,*,">
<FRAME SRC="f2.htm" title="แสดงสารบัญ">
<FRAME SRC="f3.htm" title="แสดงรายละเอียดจากสารบัญ">
</FRAMESET>
```

<FRAME SRC="f4.htm" title="แสดงส่วนท้ายของเว็บไซต์">
</FRAMESET>

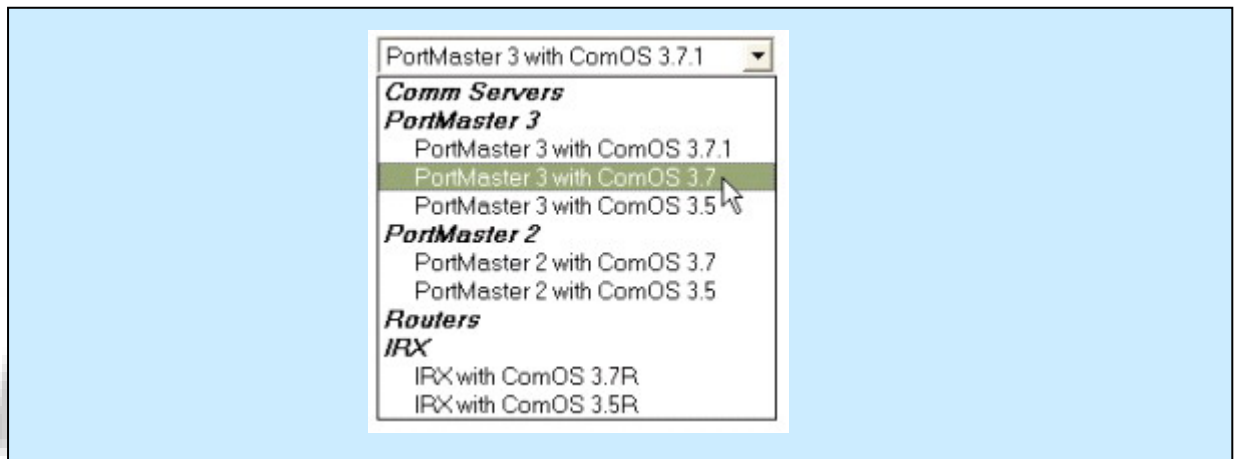
12.2) หากการอธิบายความหมายของในแต่ละเฟรม โดยใช้ Attribute title ยังไม่เพียงพอ อาจจะใช้ Attribute longdesc ใน link ไปยังคำอธิบายที่สามารถอธิบายได้มากกว่าใน Attribute title ซึ่งโดยปกติแล้ว ใน Attribute title จะสามารถบรรจุตัวอักษรได้ 256 ตัวอักษร เพราะฉะนั้นถ้าต้องการคำอธิบายให้มากกว่าต้อง ใช้ Attribute longdesc

Code HTML

<FRAME src="f3.htm" longdesc="f3desc.htm" title="แสดงข้อมูลรายละเอียดจากสารบัญ">

12.3) การสร้างรายการเลือก ให้แบ่งกลุ่มของรายการให้เหมาะสม โดยอาจจะใช้ Element OPTGROUP ในการกำหนดกลุ่มของรายการ ดังตัวอย่าง

ตัวอย่าง

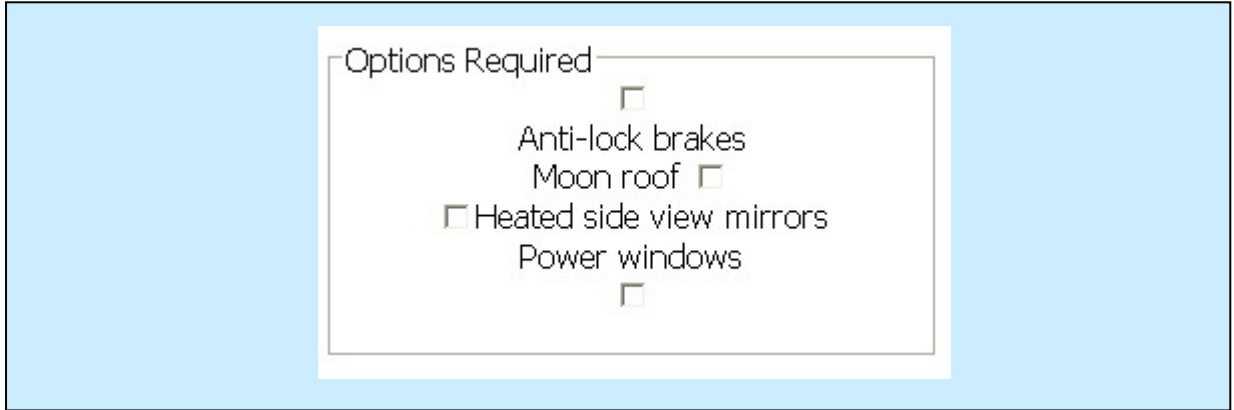


Code HTML

```
<SELECT name="ComOS">
<OPTGROUP label="Comm Servers">
  <OPTGROUP label="PortMaster 3">
    <OPTION label="3.7.1" value="pm3_3.7.1"> PortMaster 3 with ComOS 3.7.1
    <OPTION label="3.7" value="pm3_3.7"> PortMaster 3 with ComOS 3.7
    <OPTION label="3.5" value="pm3_3.5"> PortMaster 3 with ComOS 3.5
  </OPTGROUP>
  <OPTGROUP label="PortMaster 2">
    <OPTION label="3.7" value="pm2_3.7">PortMaster 2 with ComOS 3.7
    <OPTION label="3.5" value="pm2_3.5">PortMaster 2 with ComOS 3.5
  </OPTGROUP>
</OPTGROUP>
<OPTGROUP label="Routers">
  <OPTGROUP label="IRX">
    <OPTION label="3.7R" value="IRX_3.7R">IRX with ComOS 3.7R
    <OPTION label="3.5R" value="IRX_3.5R"> IRX with ComOS 3.5R
  </OPTGROUP>
</OPTGROUP>
```

12.4) ความสัมพันธ์ของ LABEL กับ Control ให้อยู่ในตำแหน่งที่เหมาะสม เพราะถ้าผู้ใช้ โปรแกรมในการอ่านหน้าจอ แล้วตำแหน่งของ Label กับ Control ไม่สัมพันธ์กันจะทำให้ไม่สามารถ หรือเข้าใจได้ ยาก ดังตัวอย่าง

ตัวอย่าง



Code HTML

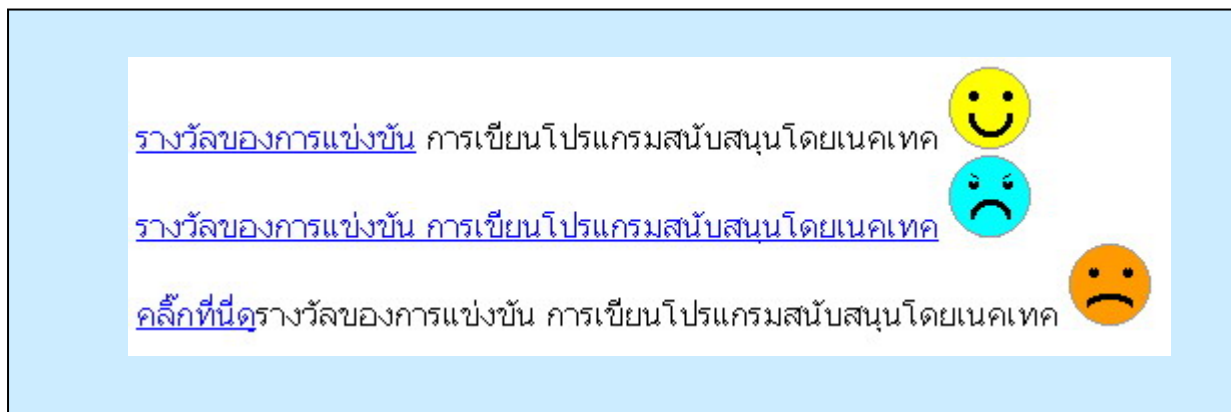
```
<FIELDSET>
  <LEGEND align="top"> Options Required</LEGEND><BR>
  <INPUT TYPE="CHECKBOX" NAME="opt1" ID="opt1" VALUE="abs"><BR>
  <LABEL for="opt1">>Anti-lock breaks</LABEL>
  <LABEL for="opt2">Moon roof </LABEL>
  <INPUT TYPE="CHECKBOX" NAME="opt2" ID="opt2" VALUE="mr"><BR>
  <INPUT TYPE="CHECKBOX" NAME="opt3" ID="opt3" VALUE="hm">
  <LABEL for="opt3">Heated mirrors</LABEL>
  <LABEL for="opt4">Power windows</LABEL><BR>
  <INPUT TYPE="CHECKBOX" NAME="opt4" ID="opt4" VALUE="pw">
</FIELDSET>
```

จากตัวอย่างข้างต้นจะเห็นได้ว่า ลักษณะการออกแบบ แบบนี้จะมีผลต่อผู้ใช้ในการใช้โปรแกรมอ่านหน้าจอ เพราะจะทำให้ผู้ใช้เข้าใจยาก

13. จัดโครงสร้างการไหลของข้อมูล ให้ถูกต้องและชัดเจน

13.1 ในการสร้างการเชื่อมโยงไปยังหน้าใหม่หรือข้อมูลอื่นโดยใช้ข้อความในการแสดงการเชื่อมโยง ให้ใช้ข้อความที่รวบรัดเข้าใจได้ง่าย ในการสื่อความหมายถึงสิ่งที่จะเชื่อมโยงไป หรือ ถ้าจะใช้ Attribute Title ในการอธิบายถึงสิ่งที่จะเชื่อมโยงไปก็ได้

ตัวอย่าง



จากตัวอย่างจะเห็นได้ว่าในส่วนของบรรทัดแรก จะเป็นการกำหนดข้อความในการเชื่อมโยงที่กระชับ รวบรัด ส่วนในบรรทัดที่สองจะเป็นการกำหนดข้อความของการเชื่อมโยงที่ มากเกินจำเป็น ส่วนในบรรทัดที่สามเป็นการกำหนดข้อความการเชื่อมโยงที่ไม่ถูกต้องเพราะ ผู้ใช้จะไม่รู้เลยว่าจะถูกเชื่อมโยงไปที่ไหนถ้าไม่

เข้าไปก่อน และถ้าเป็นกรณีเช่นนี้ควรที่จะเพิ่มส่วนการอธิบายการเชื่อมโยงข้อความนั้นๆด้วย Attribute Title ดังตัวอย่าง Code

Code HTML

```
...  
<a href="next.html" title="เชื่อมโยงไปยังรางวัลของการแข่งขัน">คลิกที่นี่</a>รางวัล  
ของการแข่งขัน การเขียนโปรแกรมสนับสนุนโดยเนคเทค  
...
```

13.2 การเตรียมข้อมูลสำหรับการค้นหาของข้อมูลในสวนสาระสำคัญ (Keyword) บน Internet ที่เรียกว่า Metadata ซึ่งจะมีลักษณะดังนี้

Code HTML

```
<head>  
  <meta name="keywords" content="PWD, Web accessibility, Assitive  
  Technology Center" />  
</head>
```

จากตัวอย่างจะเห็นได้ว่าเป็นการเตรียมข้อมูลให้ค้นหา ในส่วนของคำว่า "PWD", "Web", "Accessibility" "Assitive Technology Center" โดยถ้ามีใครต้องการหาข้อมูลที่มีความเกี่ยวข้องกับคำเหล่านี้จาก เว็บไซต์ประเภท Search Engine เช่น www.google.com หรือ www.yahoo.com เป็นต้น ก็จะพบกับเว็บไซต์นี้ได้

13.3 เตรียมการแสดงผลแผนผังของเว็บไซต์ (Site Map) หรือสารบัญ ให้กับผู้ใช้ จุดประสงค์เพื่อให้ผู้ใช้สามารถรู้ได้ว่า หัวข้อของข้อมูลต่างๆ มีอย่างไรบ้าง เพื่อสะดวกในการค้นหา

ตัวอย่าง :- แผนผังแสดงหัวข้อต่างๆ ภายในเว็บไซต์ทั้งหมด ของเว็บไซต์เนคเทค (ที่ตัดมาแสดงให้เห็นส่วนหนึ่ง) และแผนผังของเว็บไซต์ที่แสดงหัวข้อของหน้าต่างๆที่มีในเว็บไซต์นี้



โดยการใช้ Attribute Title ใน Tag <A> ช่วยในการบอกถึงระดับของหัวข้อต่างๆ ก็ได้ เช่น

Code HTML

```
<A HREF="a.htm" TITLE="level 1">ยินดีต้อนรับ</A>
```


บ้าน (main page)

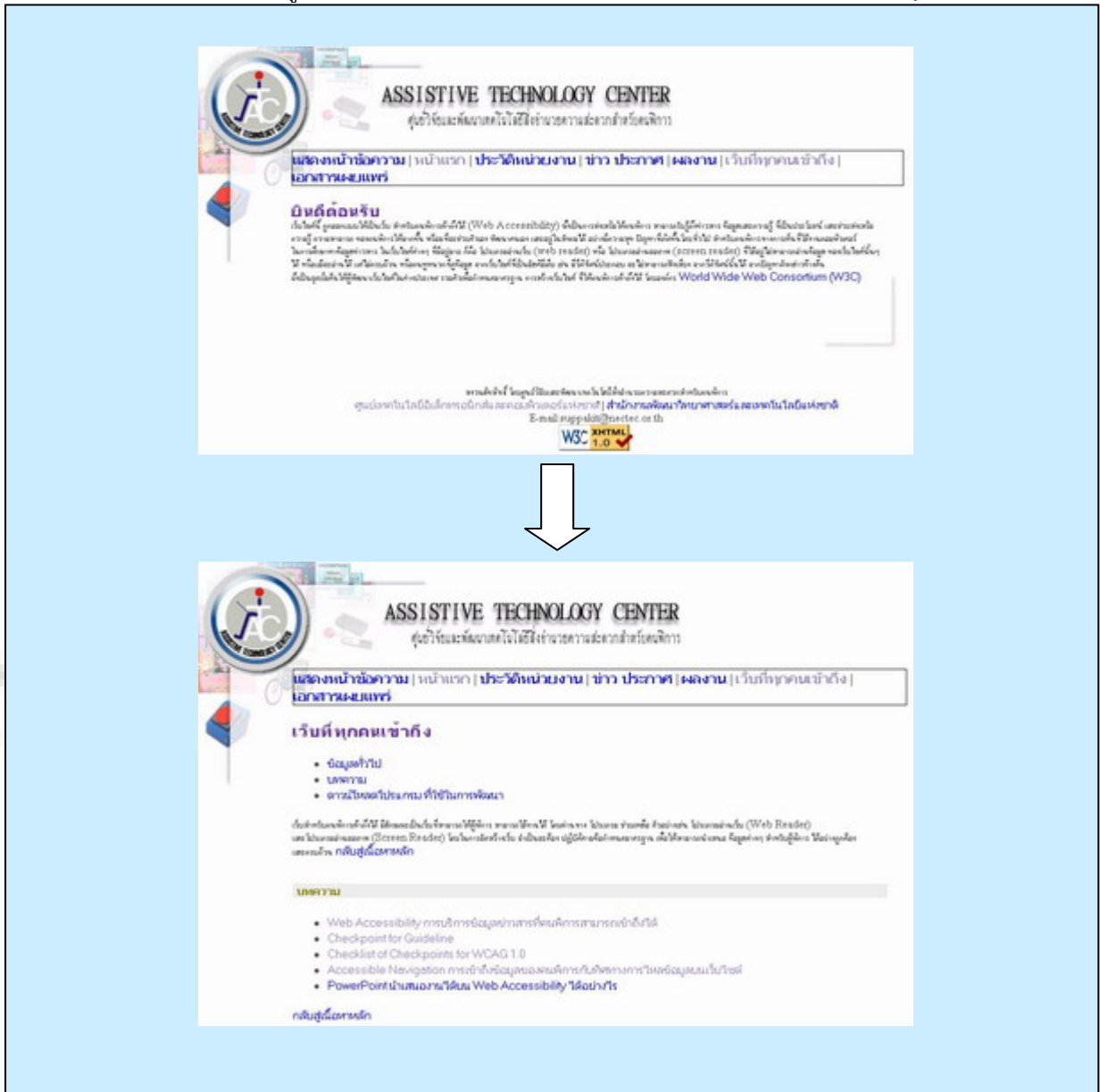
ข้อมูลบริษัท

เกี่ยวกับเรา

กิจกรรม

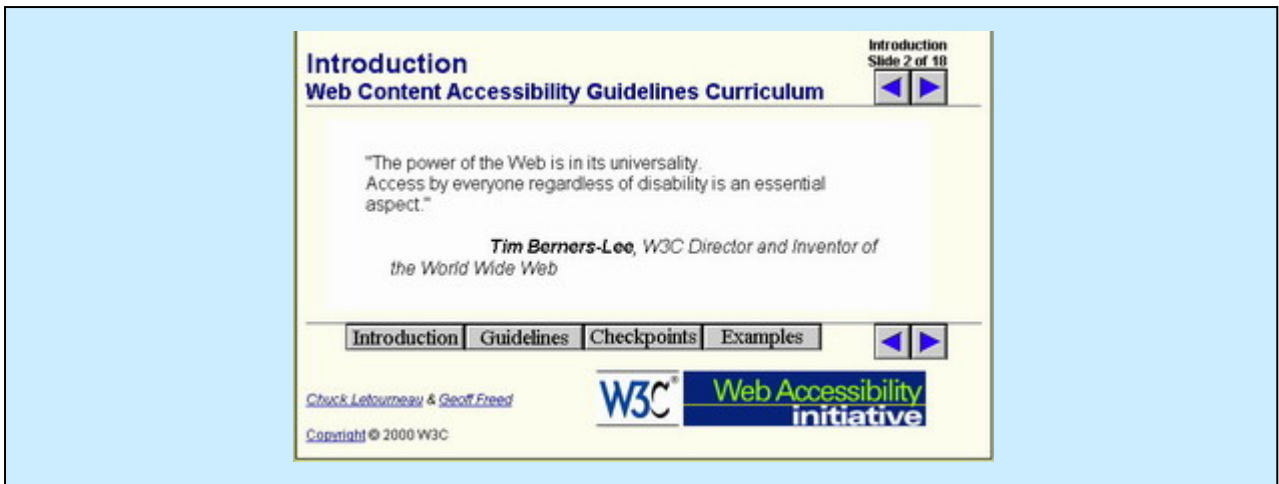
...

13.4 การใช้วิธีการเข้าถึงข้อมูลในแต่ละหน้าของเว็บไซต์ ให้มีลักษณะโครงสร้างของแต่ละหน้าให้เป็นแบบเดียวกัน เพื่อช่วยให้ผู้ใช้ง่ายต่อการจดจำตำแหน่งของการเชื่อมโยงที่ไปหน้าต่างๆได้ง่าย



13.5 การเตรียม Navigation Bar สำหรับที่จะให้ผู้ใช้สามารถที่เลือกหน้าเว็บเข้าไปได้โดยอาจจะเป็นปุ่ม Next กับ Back ซึ่งจะทำหน้าที่เป็นการเปลี่ยนหน้าเว็บที่อยู่ถัดไป แต่ถ้าต้องการให้ไปหน้าเว็บที่อื่นที่ไม่ได้อยู่ใกล้ๆกัน อาจจะใช้เป็นปุ่ม link เพื่อเชื่อมโยงไปยัง Content ซึ่งจะระบุหัวข้อๆไว้ให้เลือกเข้าไปอีกทีหนึ่งได้

ตัวอย่าง :-ในการทำ Slide แสดงการนำเสนอข้อมูลแบบ Presentation ซึ่งจะต้องนำเสนอไปที่ละหน้า และควรต้องมีการเตรียมปุ่มสำหรับการเชื่อมโยงไปยังหน้าถัดไปหรือ ปุ่มกลับก่อนหน้า

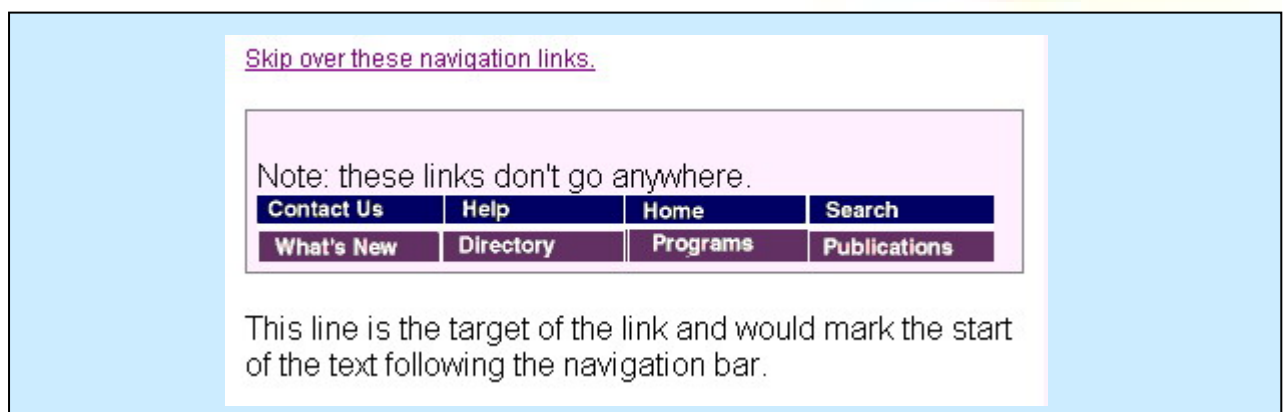


Code HTML

```
<a href="int1-0.htm" accesskey="p">
  
</a>
<a href="int3-0.htm" accesskey="n">
  
</a>
```

13.6 การสร้างวิธีการอ่านข้ามส่วนที่เป็นเมนู (Skip Navigation) สำหรับผู้ที่ต้องใช้โปรแกรมอ่านหน้าจอจำเป็นต้องมีส่วนนี้ช่วยให้ผู้ใช้สามารถข้ามการอ่านในส่วนที่เป็นเมนูได้ ซึ่งโดยปกติแล้วหน้าของเว็บไซต์ที่ออกแบบในแต่ละหน้าจะมีโครงสร้างเดียวกัน เป็นส่วนมาก ทำให้ผู้ใช้สามารถจำได้ง่าย แต่ผู้ใช้เมื่อสามารถจำได้แล้วว่า เมนูในการเลือกเป็นอะไรแล้ว จะทำให้เกิดความรู้สึกเบื่อในการฟังคำซ้ำๆ ดังนั้นจึงต้องมีวิธีการสร้างการอ่านข้ามโดยใช้วิธีการ link หรือเชื่อมโยง ข้อความภายในหน้า เว็บ ดังตัวอย่าง

ตัวอย่าง :- จากตัวอย่างจะเห็นได้ว่ามีส่วนที่เรียกว่า Skip Navigation เพื่อที่จะข้ามการอ่านในส่วนเมนูไปยังส่วนเนื้อหา



Code HTML

```
...
<A HREF="#content">Skip over these navigation links.</A>
<P> Note: these links don't go anywhere.<BR>
<A HREF="#here" NAME="here"><IMG SRC="img/tb02_01.gif"
ALT="Contact Us " BORDER=0></A>
```

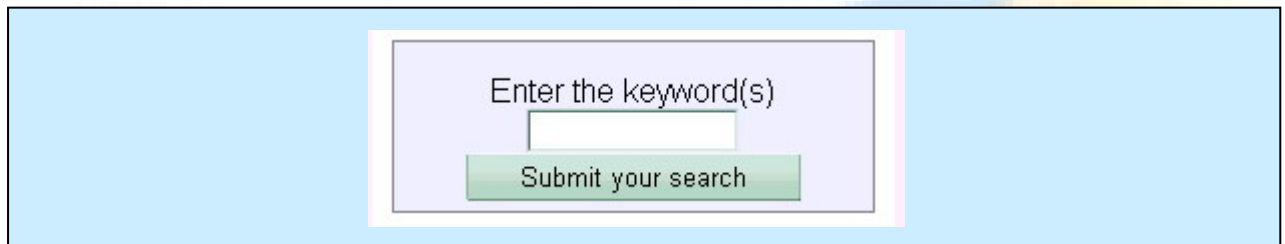
```

<A HREF="#here"><IMG SRC="img/tb02_02.gif" ALT="Help "
BORDER=0></A>
<A HREF="#here"><IMG SRC="img/tb02_04.gif" ALT="Home "
BORDER=0></A>
<A HREF="#here"><IMG SRC="img/tb02_03.gif" ALT="Search "
BORDER=0></A><BR>
<A HREF="#here"><IMG SRC="img/tb03_01.gif" ALT="What's New "
BORDER=0></A>
<A HREF="#here"><IMG SRC="img/tb03_02.gif" ALT="Directory "
BORDER=0></A>
<A HREF="#here"><IMG SRC="img/tb03_03.gif" ALT="Programs and
Services " BORDER=0></A>
<A HREF="#here"><IMG SRC="img/tb03_04.gif" ALT="Publications and
Forms " BORDER=0></A></P>
<P><A NAME="content">This line is the target of the link and would mark the
start of the text following the navigation bar.</A></P>
...

```

13.7 การเตรียมเครื่องมือสำหรับการค้นหา (Searching) ของเว็บไซต์ ต้องให้สามารถหาค่าได้ในรูปแบบตามเงื่อนไขได้ ทั้งแบบการใช้ Boolean หรือ การใช้แบบ Collection ของการแบ่งกลุ่มการเลือก ซึ่งอาจจะเป็นการใช้ Radio button Check Box หรือ จะเป็นการเลือกแบบ Selection List และก็ต้องระลึกไว้เสมอว่าในการออกแบบฟอร์มนั้นก็ต้องให้ Accessible ด้วย

ตัวอย่าง



13.8 การสร้างการเชื่อมโยงของข้อความในส่วนของหัวข้อ หรือ ย่อหน้า หรือ รายการแสดง(list) ไปยังหน้าใหม่ ต้องให้มีจุดสังเกตได้ง่าย กล่าวคือให้มีการใช้ข้อความที่ต้องการเชื่อมโยงนั้น ในข้อความส่วนแรกหรือส่วนหน้า ที่เรียกว่า Front- loads เพื่อให้ผู้ที่อ่านสามารถเข้าใจได้ง่ายในการอ่าน ให้เป็นลำดับต่อๆ กันไป

ตัวอย่าง

- โทรศัพท์มือถือที่น่าสนใจ
- [Nokia](#) - เป็นมือถือที่ยังครองความเป็นที่ 1 อยู่ในขณะนี้
 - [Sony-Ericsson](#) - เป็นมือถือที่เป็นเจ้าแรกๆ ในการพัฒนาเทคโนโลยีบนมือถือ
 - [Samsung](#) - เป็นมือถือที่เป็นเทคโนโลยีจากเกาหลีที่กำลังมาแรงในขณะนี้

Code HTML

```

<ul>โทรศัพท์มือถือที่น่าสนใจ
<li><a href="http://www.nokia.com">Nokia</a> - เป็นมือถือที่ยังครองความเป็นที่ 1 อยู่ในขณะนี้</li>

```

- Sony-Ericsson - เป็นมือถือที่เป็นเจ้าแรกๆ ในการพัฒนาเทคโนโลยีบนมือถือ
- Samsung - เป็นมือถือที่เป็นเทคโนโลยีจากเกาหลีที่กำลังมาแรงในขณะนี้

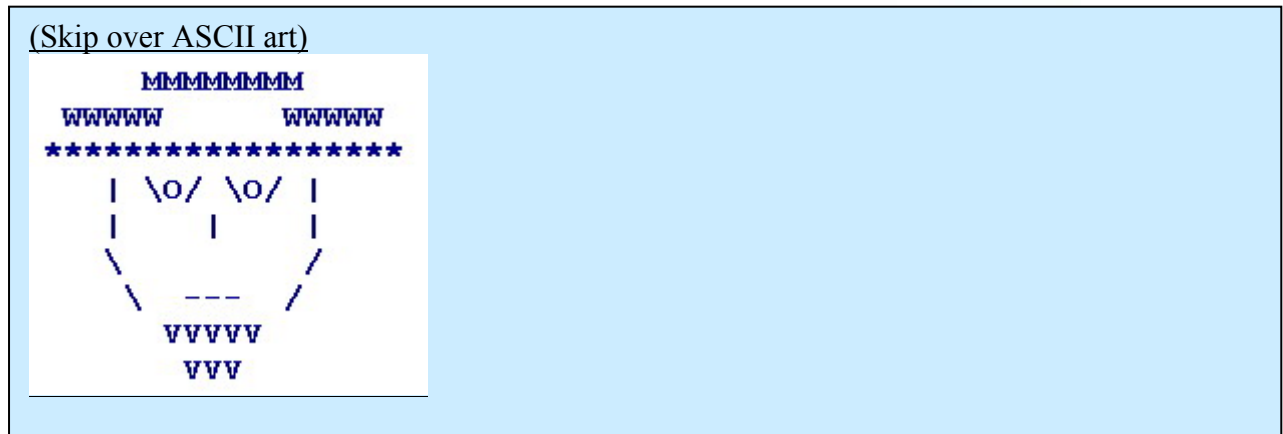
13.9 การจัดเตรียมข้อมูลสำหรับการแจ้งให้ทราบเกี่ยวกับการใช้เอกสารที่อยู่ในกลุ่มเดียวกัน แต่ต้องแสดงในหลายหน้า เช่น เอกสารที่เป็นบทเรียน ซึ่งมีหลายหน้าควรมีการใช้ "LINK rel=next" และ "LINK rel=prev" ในการบอกให้ผู้ใช้ทราบ

Code HTML

```
<HEAD>
  <link rel="prev" href="lesson1.htm">
  <link rel="contents" href="overview.htm">
  <link rel="next" href="lesson2.htm">
</HEAD>
```

13.10 การใช้ตัวอักษรมาทำเป็นภาพ(ASCII Art) ให้ทำการเพิ่มส่วนการสร้าง link ข้ามการอ่าน เพื่อให้ผู้ใช้สามารถเลือกที่จะข้ามการอ่านในช่วงที่เป็น ASCII Art ไป เพื่อ

ตัวอย่าง



Code HTML

```
<a href="#post-art">skip over ASCII art </a> of a bearded person wearing a hat.
<!--ข้าม ASCII art ไปที่ตำแหน่ง -->
<a name="post-art">First text of following main body...</a>
```

14. สร้างข้อมูลเอกสารต่างๆ ให้ถูกต้อง และเข้าใจง่าย

14.1 การใช้คำศัพท์ที่เป็นคางๆ เข้าใจได้โดยคนทั่วไป ไม่ใช่ศัพท์ที่ยากเกินไป

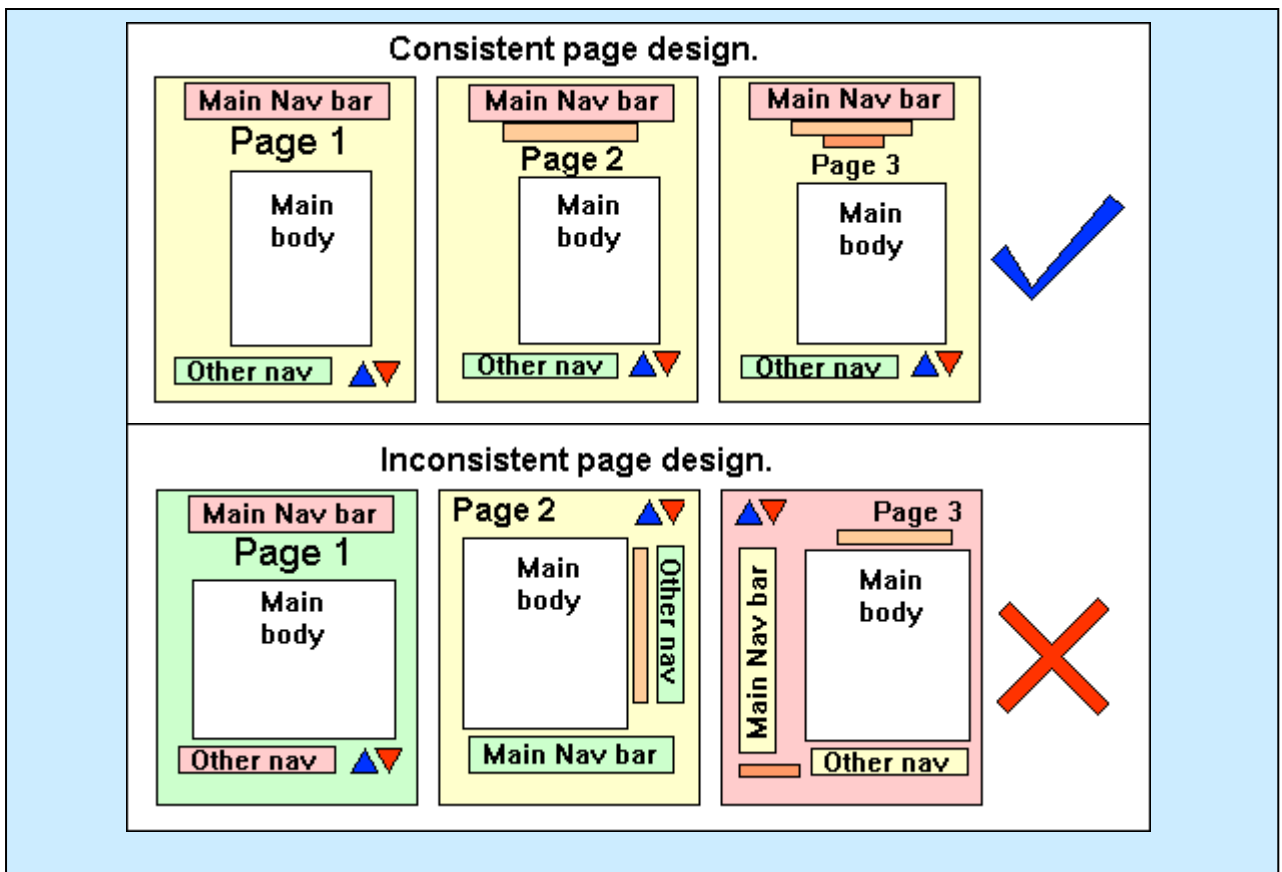
14.2 การสร้างความร่วมกับข้อมูลกราฟิก หรือข้อมูลเสียง จะทำให้เข้าใจความหมายได้ง่ายขึ้น ซึ่งสามารถที่จะช่วยในการเข้าถึงข้อมูลของ คนได้หลายประเภท ตัวอย่างเช่น คนที่มีความบกพร่องทางการมองเห็น ก็สามารถรับข้อมูลได้จากข้อมูลเสียง และถ้าเป็น คนที่บกพร่องทางการได้ยิน ก็สามารถที่จะเห็นภาพหรืออ่านข้อความได้

ตัวอย่าง



14.3 การวางโครงสร้างของเว็บเพจ ในแต่ละหน้ามีการออกแบบเป็นโครงสร้างเดียวกันจะทำให้ผู้ใช้สะดวกในการเข้าถึงข้อมูลได้ง่ายและรวดเร็ว คือสะดวกในการจำได้ง่ายโครงสร้าง สามารถรู้ส่วนประกอบในโครงสร้างได้ว่าต่อไปจะเป็นข้อมูลของส่วนใด

ตัวอย่าง



เอกสารอ้างอิง

- [1] **Web Content Accessibility Guidelines** WCAG1.0 (<http://www.w3.org/TR/WCAG10/>)
- [2] **Bobby Online Service** (ที่มา <http://bobby.watchfire.com/bobby/html/en/index.jsp>)
- [3] **The Cynthia Says** (ที่มา <http://www.cynthiasays.com/>)
- [4] **A-Prompt** (ที่มา <http://aprompt.snow.utoronto.ca/>)