HTML5 Programming Basic Skills (1)

Wen-Hsiang Lu (盧文祥)
Department of Computer Science and Information Engineering,
National Cheng Kung University
2015/12/25

Outline

- HTML5 New Functions
- Canvas

HTML5

- New Functions
  - Canvas
  - Video/Audio
  - Drag & Drop
  - RichText
  - Web Storage
  - Web Workers
  - Data Push
  - File API

Semantic Tag

- Example

```
<html>
<body>
   <div class="header">
      Page head …
   </div>
…
   <div class="content">
      Page content …
   </div>
…
  <div class="footer">
      Page end …
   </div>
</body>
</html>
```

```
<html>
<body>
   <header>
      Page head …
   </header>
…
   <section class="content">
      Page content …
   </section>
…
  <footer>
      Page end
  </footer>
</body>
</html>
```

HTML5 Development

- Working Draft (W3C): January, 22, 2008
- Last Call (WHATWG): October, 2009

HTML5 Language Syntax

- Perhaps follow XHTML

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>XHTML sample</title>
</head>
<body>
   <h1>Page head …</h1>
   <p> Page content …</p>
   <input type="text" value="Enter name" />
</body>
</html>
```

New Tags in HTML5

表 2.2 ‖ 與文件結構相關的新元素

section	通用的 section，構成文件的結構
article	section 的一種，表示從網頁中獨立的內容
aside	section 的一種，表示補充資訊的邊欄 section
nav	section 的一種，表示導覽目錄的邊欄 section
footer	section 的頁尾
header	section 的頁首
hgroup	多個標題的組合

表 2.3 ‖ 內嵌外部內容

figure	將包含說明的內容放入文件中時可用
video	動畫播放器
audio	聲音播放器
source	針對 video/audio，指定資源的連結以及類型
canvas	點陣圖畫布
embed	內嵌要透過外掛 (plug-in) 執行的內容

表 2.4 ‖ 表單

keygen	產生公開金鑰對
output	輸出用的表單元素
progress	表示進度
meter	計量

New Tags in HTML5

表 2.5 | 文字・其他

mark	強調網頁的目的或特別的脈絡
ruby/rt/rp	指定顯示注音標記
time	顯示日期時間
command	功能表中可執行的指令
details	使用者可以任意展開或關閉的詳細資訊
datalist	用來保存自動帶入候選值等等用途的資料清單

Cancelled Tags in HTML5

表 2.6 ‖ HTML5 廢止的元素 (包含非標準的元素)

元素名稱	說明	替代方案
applet	內嵌 Java Applet	請改使用 embed/object 元素
acronym	表示首字母縮寫	請改使用 abbr 元素
dir	目錄一覽	請改使用 ul 元素
frame	窗格	請使用 iframe 與 CSS, 結合伺服器端內容
frameset	窗格集合	
noframes	傳送給不支援窗格的瀏覽器的內容	
isindex	傳送到伺服端的 text 輸入欄位	請用 form 與文字欄位的組合
listing	直接輸出內容	請改用 pre 元素與 code 元素
xmp	直接輸出內容	
noembed	embed 不能使用時的代替元素	必須處理補救時, 可以使用 object 元素
plaintext	直接輸出內容	請用"text/plain"的 MIME 類型
rb	指定 ruby 元素目標的文字	文字請直接寫入 ruby 元素內

Cancelled Tags in HTML5

basefont	指定基準字體	請改用 CSS
big	放大字體	
blink	使字體閃爍	
center	內容置中對齊	
font	指定字型	
marquee	捲動文字	
s	刪除線	
spacer	插入空白	
strike	刪除線	
tt	以等寬字體顯示	
u	字體加下底線	
bgsound	音效	請改用 audio 元素

<canvas> Tag

- Use JavaScript to draw graphics on a web page.
- A rectangular area

```
<canvas id="darea1" width="300" height="200"></canvas>
var canvas= document.getElementById("darea1");
var cx= canvas.getContext("2d");


cx.strokeRect(0, 0, 90, 100);
cx.fillRect(50, 50, 70, 100);
cx.clearRect(70, 70, 30, 50);
cx.fillText("Canvas is powerful in HTML 5!", 120, 20);
cx.rect(0, 0, 300, 200);
```

```
cx.strokeRect (x, y, width, height);
cx.fillRect (x, y, width, height);
cx.clearRect (x, y, width, height);
```

- [Ex] html5-1-canvas.htm

drawImage() method

- Draw image
  - drawImage(image, sx, sy, sw, sh, dx, dy, dw, dh);
    - sx, sy, sw, sh: spliting area
    - dx, dy, dw, dh: displaying area
- [Ex] html5-2-canvas-drawimage.htm

Draw Complex Path

- Steps for drawing complex path
  - beginPath(): initiate the path drawing
  - moveTo(), lineTo(), arcTo(), arc(), … : use Canvas API to draw path
    - arcTo(x1, y1, x2, y2, radius)
    - arc(x, y, radius, startAngle, endAngle, anticloclwise)
  - stroke(), fill(), … : display path
  - closePath(): connect the lines together to form a shape
- [Ex] html5-3-canvas-stroke.htm

Display Path: stroke(), fill() & clip() methods

- Display complex path
  - stroke(): display the path with lines
  - fill(): fill in with some color or a pattern inside the assigned path
  - clip(): clip the assigned path

- [Ex] html5-4-canvas-clip.htm

Drawing with Color: Gradient

- Color or Pattern Style
  - fillStyle property: set color or pattern style when drawing a diagram
  - strokeStyle property: set color or pattern style when drawing a line
- CSS color: [Ex] fillStyle= “green”;
- Gradient (漸層)
  - Gradient color setting

```
g1= cx.createLinearGradient (x0, y0, x1, y1);
g1= cx.createRadialGradient (x0, y0, r0, x1, y1, r1);
g1.addColorStop(offset, color);
// offset= 0 => the start point of gradient
// offset= 1 => the end point of gradient
cx.fillStyle= g1;
```

- [Ex] html5-5-canvas-gradient.htm

Drawing with Pattern

- Pattern (圖樣)
  - Pattern setting

```
g1= cx.createPattern(image, repeat);
cx.fillStyle= g1;
```

    - repeat: four styles
      - Repeat, repeat-x, repeat-y, no-repeat
  - Transparency
    - globalAlpha property
      - 0: completely transparent
      - 1: completely not transparent
- [Ex] html5-6-canvas-pattern.htm

Shadow

- Shadow effect for lines or diagrams

```
cx.shadowOffsetX = 5;
cx.shadowOffsetY = 5;
cx.shadowColor = "gray";
cx.shadowBlur = 2;  //模糊度
```

- [Ex] html5-7-canvas-shadow.htm

Overlap: globalCompositeOperation

- Diagram overlapping
  - globalCompositeOperation property
- [Ex] html5-8-canvas-composite.htm

```
globalCompositeOperation=
1.  "source-atop";
2.  "source-in";
3.  "source-out";    //ok
4.  "source-over";   //ok

5.  "destination-atop";
6.  "destination-in";
7.  "destination-out";
8.  "destination-over";

9.  "lighter";
10. "copy";
11. "xor";    //ok
```

Draw Text: fillText() & strokeText()

- Draw text
  - fillText(text, x, y, maxWidth);
  - strokeText(text, x, y, maxWidth);
  - font
  - textBaseline: top, middle, bottom
  - textAlign: left, right, center
  - measureText(text): measure information of drawing text

- [Ex] html5-9-canvas-text.htm

<canvas> Transform (1)

translate (x, y);
scale (x, y);
rotate (angle);

- Transform

```
var canvas= document.getElementById("darea1");
var cx= canvas.getContext("2d");

cx.translate(50,20);
cx.fillText("translate=>"+text1,30,30);  //(2)

cx.scale(3,2);
cx.fillText("scale=>"+text1,50, 50);  //(3)

cx.rotate(45*Math.PI/180);
cx.fillText("rotate=>"+text1,150, 60);  //(4)
```

- [Ex] html5-12-canvas-translate.htm

<canvas> Transform (2)

- Transform

```
cx.globalAlpha= 0.5;
 cx.translate(100,100);
 cx.drawImage(img, 0, 0);    //(2)

cx.globalAlpha= 0.2;
 cx.rotate(30*Math.PI/180);
 cx.drawImage(img, 0, 0);    //(3)

 cx.transform( 1,0,0,-0.5, 20,img.height*1.5);
 cx.drawImage(img, 0, 0);    //(4)
```

transform (m_{11}, m_{12}, m_{21}, m_{22}, dx, dy);

$$\begin{pmatrix} m_{11}, m_{12} \\ m_{21}, m_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} dx \\ dy \end{pmatrix}$$

- [Ex] html5-13-canvas-transform.htm

save() & restore() methods

- Save/Restore states of drawing environment
  - cx.save();
  - cx.translate(50,20);
  - cx.fillText("translate=>"+text1,30,30); //(2)

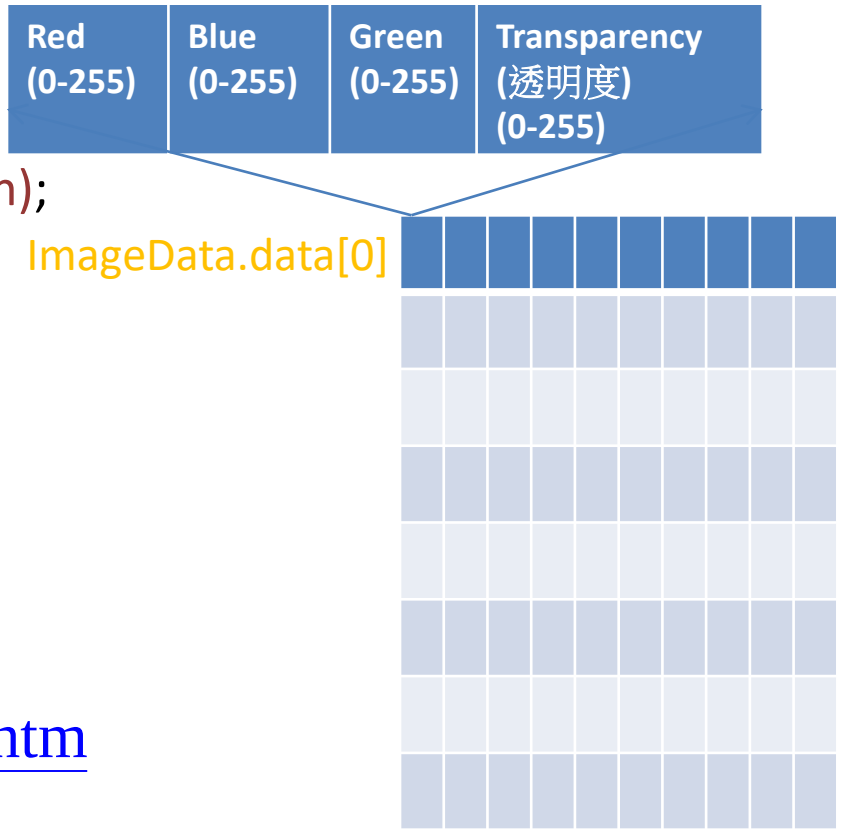
  - cx.restore();
  - cx.rotate(45*Math.PI/180);
  - cx.fillText("rotate=>"+text1,150, 0); //(4)

- [Ex] html5-14-canvas-save.htm

<canvas> Bitmap Drawing

- Bitmap array operation

```
var imageData= cx1.getImageData(x, y, w, h);
for (i=0; i<imageData.data.length; i++){
  if (i % 4 == 3){
    imageData.data[i]= 255*alpha;
  }
}
cx1.putImageData(imageData, x, y);
```



- [Ex] html5-15-canvas-imagedata.htm

<canvas>Animation Example

- Animation process
  - setInterval(), setTimeout()
  - clearRect()
  - save(), restore()

- [Ex] html5-16-canvas-clock.htm