emit feltrinelli
ente morale istruzione tecnica

HTML5 and Next RIA Apps

Le API canvas HTML5

Le API canvas HTML5

Il concetto di canvas è stato definito da Apple nel motore di WebKit di Mac OSX e utilizzato per creare i widget della dashboard.

Prima dell'introduzione delle canvas era possibile disegnare nei browser solo attraverso plugin come Adobe Flash o mediate gli SVG (Scalable Vector Graphics).

Gli elementi canvas creano un'area rettangolare nella pagina, che ha dimensioni di default pari a 300x150 px e per inserire un canvas nella nostra pagina, basta in seguente tag:

```
<canvas></canvas>
```

Dopo aver aggiunto l'elemento canvas nella pagina si, può utilizzare Javascript per manipolarlo o disegnarci dentro. Le API canvas supportano le operazioni di disegno bidimensionale.

Il canvas è, in estrema sintesi, una grande matrice di pixel, ognuno dei quali modificabile singolarmente nelle sue quattro componenti RGBA, rosso, verde, blu e alpha, la trasparenza.
Quindi sul canvas, vi è la possibilità di replicare i comportamenti fino ad oggi esclusive di Flash.

Le API canvas HTML5

Come la maggior parte degli elementi HTML, anche al canvas possono essere associate regole CSS per aggiungere bordi, margini etc. In tal senso vedremo un'esercitazione su come creare un triangolo di “alert” .

Supporto dei browser
Escludendo Internet Explorer, ad oggi tutti i browser supportano i canvas; tuttavia alcune sezioni delle specifiche per l'implementazione dei canvas non sono ugualmente supportate, come ad esempio le API per la gestione del testo nel disegno.

Browser	Supporto
Chrome	Dalla versione 1.0
Firefox	Dalla versione 1.5
Internet Explorer	Non supporato
Opera	Dalla versione 9
Safari	Dalla versione 1.3

Per sfruttare i canvas anche su Internet Explorer si può utilizzare il progetto open-source “explorercanvas” scaricabile da qui:

http://code.google.com/p/explorercanvas/wiki/Instructions

Le API canvas HTML5

Supporto dei browser

Per verificare il supporto al canvas possiamo utilizzare Modernizr oppure possiamo considerare il seguente codice:

```
<!DOCTYPE html>
<html>
<head>
<title>Test supporto Canvas</title>
<script type="text/javascript">
try
{
    document.createElement('canvas').getContext('2d');
    document.getElementById('msgSupport').innerHTML = 'Canvas supportato';
}  catch(e) {
    document.getElementById('msgSupport').innerHTML = 'Canvas NON supportato';
}

</script>
</head>
<body>
    <div id="msgSupport"></div>
</body>
</html>
```

Le API canvas HTML5

Primo esempio disegno diagonale su canvas

Adesso andremo a creare un piccolo esempio di creazione e utilizzo del canvas per disegnare una linea rettangolare:

```
<!DOCTYPE html>
<html>
<head>
<title>Test supporto Canvas</title>
<script type="text/javascript">
/**
 * Verifico esistenza Canvas
 */
function checkCanvasFeature()
{
    try
    {
        document.createElement('canvas').getContext('2d');
        document.getElementById('msgSupport').innerHTML = 'Canvas supportato';
        return true;
    }  catch(e) {
        document.getElementById('msgSupport').innerHTML = 'Canvas NON supportato';
        return false;
    }

}
```

Le API canvas HTML5

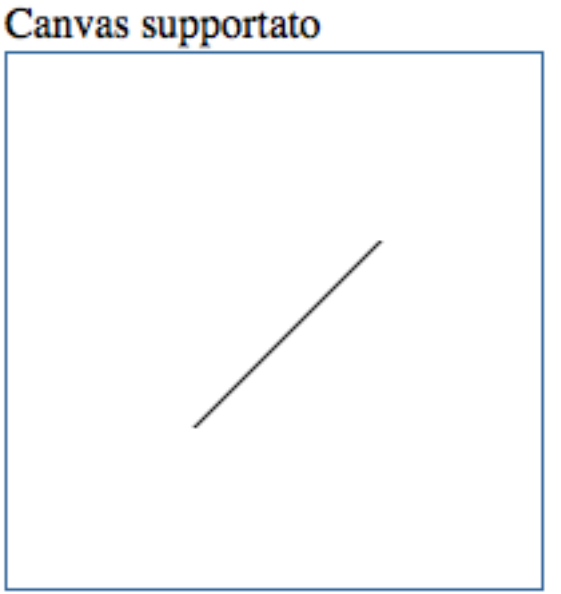
Primo esempio disegno diagonale su canvas

```
/**
 * Intercetto e disegno elementi sul canvas
 */
function creaLineaDiagonale()
{
    if(!checkCanvasFeature()) return;
    // === Recupera l'elemento canvas e il suo contesto
    var canvas = document.getElementById('testCanvas');
    var context = canvas.getContext('2d');

    // === Crea una linea con coordinate assolute
    context.beginPath();
    context.moveTo(70, 140);
    context.lineTo(140,70);

    // === Disegno materialmente la linea sul canvas
    context.stroke();
}

// === eseguo all'evento load
window.addEventListener('load', creaLineaDiagonale, true);
</script>
</head>
```

Le API canvas HTML5

Primo esempio disegno diagonale su canvas

```
    <body>
        <div id="msgSupport"></div>
        <canvas id="testCanvas"
                width="200" height="200"
                style="border: 1px solid #336699;"></canvas>
    </body>
</html>
```



Le API canvas HTML5

Secondo esempio disegno su canvas

Adesso andremo a creare un disegno leggermente complesso: un triangolo di alert:

```
/**
 * Intercetto e disegno elementi sul canvas
 */
function disegnaAlert()
{
    if(!checkCanvasFeature()) return;

    var context = document.getElementById('testCanvas').getContext("2d");

    // === Dimensioni del triangolo
    var width = 125;
    var height = 100;
    var padding = 20;

    // === Crea un percorso che disegna il triangolo
    context.beginPath();
    context.moveTo(padding + width/2, padding);
    context.lineTo(padding + width, height + padding);
    context.lineTo(padding, height + padding);
    context.closePath();
```

Le API canvas HTML5

Secondo esempio disegno su canvas

Adesso andremo a creare un disegno leggermente complesso: un triangolo di alert:

```
//  === Crea il fill gradient del triangolo
var gradient = context.createLinearGradient(0,0,0,height);
gradient.addColorStop(0, primaryColor);
gradient.addColorStop(1, secondaryColor);

// === Aggiunge ombra e colore dell'ombra
context.shadowBlur = 10;
context.shadowColor = "black";

// === Imposto lo stile della linea
context.lineWidth = lineWidth * 2;
context.lineJoin = "round";
context.strokeStyle = gradient;
context.stroke();

// Imposta l'ombra trasparente
context.shadowColor = "transparent";

// Fill the path
context.fillStyle = gradient;
context.fill();
```

Le API canvas HTML5

Secondo esempio disegno su canvas

Adesso andremo a creare un disegno leggermente complesso: un triangolo di alert:

```
// === Aggiunge l'effetto reflecion
gradient=context.createLinearGradient(0,padding,0,padding+height);
gradient.addColorStop(0, "transparent");
gradient.addColorStop(0.5, "transparent");
gradient.addColorStop(0.5, tertiaryColor);
gradient.addColorStop(1, secondaryColor);

context.fillStyle = gradient;
context.fill();

// === Imposto lo stile della linea esterna (Nera, arrotondata e in grassetto)
context.lineWidth = lineWidth;
context.lineJoin = "round";
context.strokeStyle = "#333";
context.stroke();

// === Imposto posizione, font e colore del punto esclamativo
context.textAlign = "center";
context.textBaseline = "middle";
context.font = "bold 60px 'Times New Roman', Times, serif";
context.fillStyle = "#333";
```

Le API canvas HTML5

Secondo esempio disegno su canvas

Adesso andremo a creare un disegno leggermente complesso: un triangolo di alert:

```
        // === Provo a scrivere il testo dentro
        try{
            context.fillText("!", padding + width/2, padding + height/1.5);
        }catch(ex){}

    }

    // === eseguo all'evento load
    window.addEventListener('load', disegnaAlert, true);
</script>
</head>
<body>
        <div id="msgSupport"></div>
        <canvas id="testCanvas"
                width="200" height="200"
                style="border: 1px solid #336699;"></canvas>
</body>
```

Le API canvas HTML5

Elenco link e risorse in rete sui canvas
Ecco un elenco di tutorial che mostrano la potenza dei canvas:

http://www.html5canvastutorials.com/labs/html5-canvas-elastic-stars-with-kineticjs/

http://www.html5canvastutorials.com/labs/html5-canvas-interactive-flower/

http://www.html5canvastutorials.com/labs/html5-canvas-ultimate-flash-killer/

http://www.html5canvastutorials.com/labs/html5-canvas-google-bouncing-balls/

http://www.html5canvastutorials.com/labs/html5-canvas-arc-crazy-snake/