

# MOBILE DEVELOPER



Il canvas

# Sommario

- La tela del Canvas.

# Canvas: introduzione

L'elemento `<canvas>` gode di un ottimo supporto nei principali browser (tranne IE 8).

Il Canvas è una sorta di tela di forma rettangolare sulla quale disegnare per mezzo di javascript.

# Canvas: height e width

Il TAG <canvas> prevede i soli due attributi:  
*width* per la larghezza della tela;  
*height* per l'altezza della tela.

```
<canvas id="tela" width="450" height="300" >  
</canvas>
```

# L'area bidimensionale del Canvas

Gli elementi base della tela bidimensionale sono:

*l'origine* corrisponde al punto in alto a sinistra.

*l'asse orizzontale* cioè l'asse delle x.

*l'asse verticale* cioè l'asse delle y.

L'interfaccia di programmazione è nota con il termine di HTML Canvas 2D Context.

# Supporto al Canvas

Prima di iniziare a disegnare sulla tela 2D del Canvas è necessario assicurarci che il browser supportino questo TAG:

mediante Modernizr  
oppure

```
function isSupportesCanvas(){  
    return !!document.createElement('canvas').getContext();  
}
```

# Canvas: primo semplice esempio

Si realizza una bandiera dell'Italia.

```
<script>  
  // si ottiene un riferimento al piano bidimensionale  
  var canvas = document.getElementById("bandiera-italiana");  
  var context = canvas.getContext("2d");  
  // bordo nero lungo il perimetro esterno  
  context.strokeStyle = "black";  
  context.strokeRect(0,0,450,300);  
  // rettangolo verde pari a un terzo della superficie complessiva  
  context.fillStyle = "rgb(0, 146, 70)";  
  context.fillRect(1,1,150,298);  
  // rettangolo rosso pari a un terzo della superficie complessiva  
  context.fillStyle = "#CE2B37";  
  context.fillRect(299,1,150,298);  
</script>
```



# Esportare la tela come immagine

Al fine di esportare il context 2D del Canvas è necessario:  
ottenere un riferimento al link.

impostare l'indirizzo a quello dell'immagine estratta dal  
canvas.

```
var scaricaImg = document.getElementById("bandiera-img");  
scaricaImg.href = canvas.toDataURL();
```

# Delimitatori dell'area di disegno

Dopo aver ottenuto il riferimento al contesto è necessario adoperare i seguenti delimitatori dell'oggetto Context:

`beginPath()` per l'inizio dell'area di disegno.

`closePath()` per la fine dell'area di disegno.

```
var tela = document.getElementById("tela");  
var context = mioCanvas.getContext("2d");  
context.beginPath();  
context.closePath();
```

# Canvas: spostarsi all'interno della tela

Gli spostamenti all'interno della tela avvengono mediante l'invocazione del seguente metodo:

`moveTo(x, y)` effettua il movimento dal punto corrente alla nuova coordinata `x, y`.

```
context.moveTo( X , Y );
```

# Canvas: disegnare linee

Il disegno di una linea retta all'interno della tela avviene mediante l'invocazione del seguente metodo:

`lineTo(x, y)` effettua il movimento dal punto corrente alla nuova coordinata `x, y`.

```
context.lineTo( X , Y );
```

# Canvas: angoli smussati

il metodo `arcTo(x1, y1, x2, y2, r)` che prende in input due punti di coordinate  $(x1, y1)$  e  $(x2, y2)$  ed il raggio di un arco permette di disegnare un angolo smussato fra i due punti.

```
context.arcTo(x1, y1, x2, y2, r);
```

# Canvas: stampa su tela

Il metodo `stroke()` permette di stampare sulla tela quanto disegnato sino al momento della sua invocazione.

```
context.stroke( );
```

# Canvas: disegnare forme

Con il Canvas di HTML5 è consentito disegnare le seguenti forme grazie alle delle funzioni predefinite:

Rettangoli

Cerchi

Archi

Bezier

# Canvas: forme

Il metodo `fillRect(x1, y1, x2, y2)` permette definire i vertici dell'area rettangolare e la property `fillStyle` consente di definire il colore di riempimento.

```
context.fillStyle = "blue";  
context.fillRect(0,0,260,360);
```



# Canvas: `rotate()`, `translate()` e `scale()`

Tre funzioni di manipolazione della matrice:

- rotate()* per la rotazione;
- translate()* per la traslazione;
- scale()* per la variazione della scala.

# Canvas: rotate()

Si può invertire il piano ruotandolo con il metodo rotate() che prende un solo parametro in input i radianti.

Si propone una rotazione del piano di 25° (conversione da gradi in radianti).

```
context.rotate((Math.PI/180)*25);
```

# Canvas: translate()

Il metodo `translate()` accetta due parametri: il primo indica la distanza di slittamento dall'asse verticale verso destra o sinistra, il secondo esprime la distanza dello slittamento dall'asse orizzontale verso il basso o l'alto.

# Canvas: scale()

Il metodo `scale()` dilata o restringe, rispettivamente con valori maggiori di uno o minori di uno, la distanza fra i punti della matrice, distorcendo di conseguenza il contenuto del piano.

# Canvas: gradiente

Il gradiente lineare è un metodo per effettuare sfumature da un colore all'altro. La sua realizzazione passa attraverso questi tre step:

1. Definire la direzione verso cui avviene la transizione tra due o più colori;
2. I colori coinvolti nella transizione;
3. L'avvicendamento dei colori del gradiente.

# Canvas: gradiente

Il seguente esempio mostra l'uso di un gradiente lineare per creare un background sfumato.

```
context.fillStyle="#ffff00"  
context.createLinearGradient(x1, y1, x2, y2);
```

# Canvas: testo

Per scrivere stringhe di testo nella tela il Canvas di HTML5 prevede questi metodi:

`strokeFill()` delimita i contorni del testo.

`fillText()` usato per il riempimento del corpo del testo.

# Canvas: librerie javascript

Le seguenti librerie javascript forniscono supporto nella creazione e interazione di elementi del TAG <canvas>:

Ai-canvas - da illustrator a <canvas>

Processing.js - <http://processingjs.org/>

jCotton - <http://jcotton.sourceforge.net/>

Paper.js - <http://paperjs.org/>

Createjs - <http://www.createjs.com/>

Adobe edge - <http://html.adobe.com/edge/animate/>



Domande & approfondimenti

# **MOBILE DEVELOPER**