

HTML CANVAS

CANVAS

- Il tag HTML5 `<canvas>` è letteralmente la tela su cui dipingere!
- Inseriamola nella pagina sempre con **width** e **height**.

Salvare come
canvas.html

```
<html>
<head>
<title>Esempio canvas</title>
<style type="text/css">
  body { background-color:#ededed;}
  #myCanvas { background:#fff; border:1px solid #000; }
</style>
<body>

  <canvas id="myCanvas" width="400" height="300"></canvas>

</body>
</html>
```

CANVAS

- Sulle CANVAS si dipinge tramite JAVASCRIPT!

```
<body>
  <canvas id="myCanvas" width="400" height="300"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');
    var centerX = 100;
    var centerY = 100;
    var radius = 20;

    context.beginPath();
    context.arc(centerX, centerY, radius, 0, 2 * Math.PI, false);
    context.fillStyle = 'green';
    context.fill();
  </script>
</body>
```

Ricorda!

- Per qualsiasi funzione in JAVASCRIPT puoi consultare la documentazione sul sito ufficiale:

w3schools.com

Ricorda!



w3schools.com

Ad esempio:

http://www.w3schools.com/tags/canvas_arc.asp

Contiene tutte le informazioni sul metodo **arc** che in javascript si applica al contesto!



Ciclo FOR

Il ciclo **for** permette di impostare una variabile contatore che ci permette di ripetere più volte lo stesso blocco di istruzioni.

```
for (<inizializzazione_contatore>;  
<condizione>; <incremento_contatore> )  
{  
  //istruzioni  
}
```

```
<script>
```

```
var canvas = document.getElementById('myCanvas');
```

```
var context = canvas.getContext('2d');
```

```
var centerX;
```

```
var centerY;
```

```
var radius;
```

```
for ( i=0; i<10; i++)
```

```
{
```

```
radius = Math.random()*10;
```

```
centerX = Math.random()*100;
```

```
centerY = Math.random()*100;
```

```
context.beginPath();
```

```
context.arc(centerX, centerY, radius, 0, 2 * Math.PI, false);
```

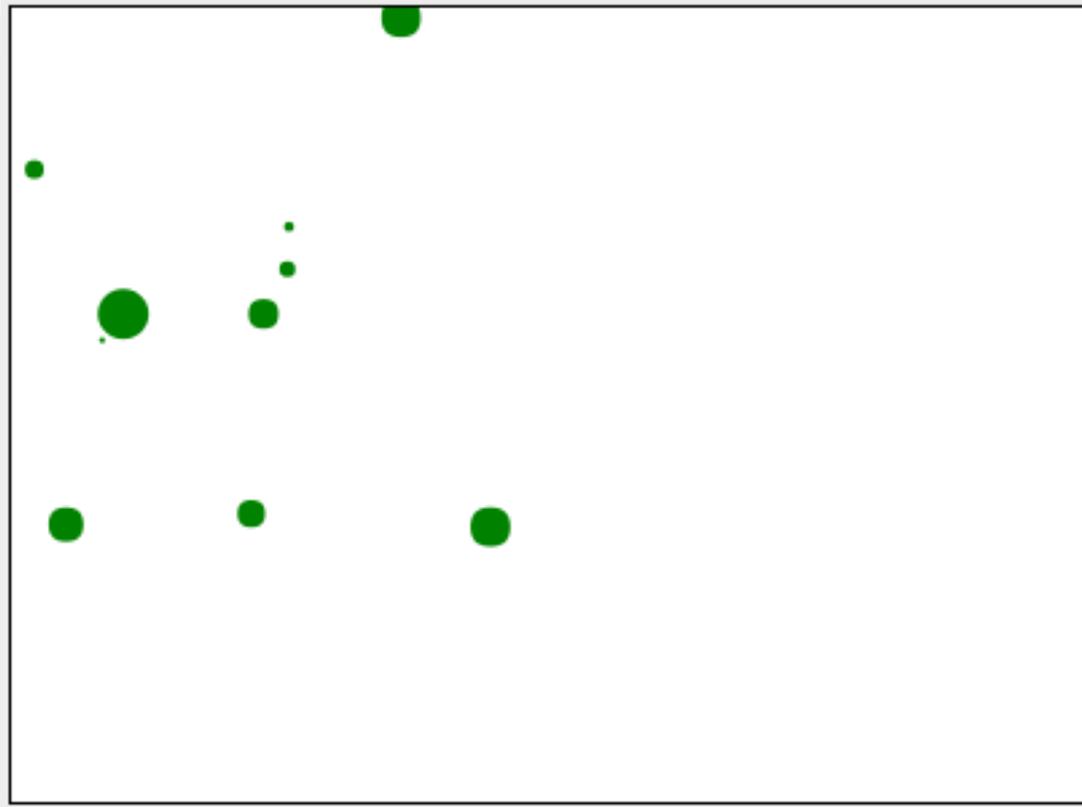
```
context.fillStyle = 'green';
```

```
context.fill();
```

```
}
```

```
</script>
```

Risultato



Math.random

JavaScript random() Method

← [JavaScript Math Object](#)

Example

Return a random number between 0 (inclusive) and 1 (exclusive):

```
Math.random();
```

The result could be:

```
0.085363824153319
```

[Try it yourself »](#)

Le palline si muovono!

- Per far muovere le palline, il nostro codice Javascript diventerà una funzione e una istruzione

setInterval(funzione, millisecondi);

- Dirà al browser di chiamare la nostra funzione ogni 100 millisecondi.

Le palline si muovono!

- Le variabili diventano Array, ovvero una lista di palline di cui si memorizzano le coordinate del centro ed il raggio.

```
var centerX = new Array(10);
```

```
<script>
```

```
var canvas = document.getElementById('myCanvas');
```

```
var context = canvas.getContext('2d');
```

```
var centerX = new Array(10);
```

```
var centerY = new Array(10);
```

```
var radius = new Array(10);
```

```
var dx = 2;
```

```
var dy = 2;
```

```
for ( i=0; i<10; i++) //inizializziamo le palline
```

```
{ radius[i] = Math.random()*10;
```

```
centerX[i] = Math.random()*200;
```

```
centerY[i] = Math.random()*200; }
```

```
function draw()
```

```
{ //funzione che disegna
```

```
for ( i=0; i<10; i++)
```

```
{ context.beginPath();
```

```
context.arc( centerX[i], centerY[i], radius[i], 0, 2 * Math.PI, false);
```

```
context.fillStyle = 'green';
```

```
context.fill();
```

```
if( centerX[i]<0 || centerX[i]>400)
```

```
dx=-dx;
```

```
if( centerY[i]<0 || centerY[i]>300)
```

```
dy=-dy;
```

```
centerX[i]+=dx;
```

```
centerY[i]+=dy;
```

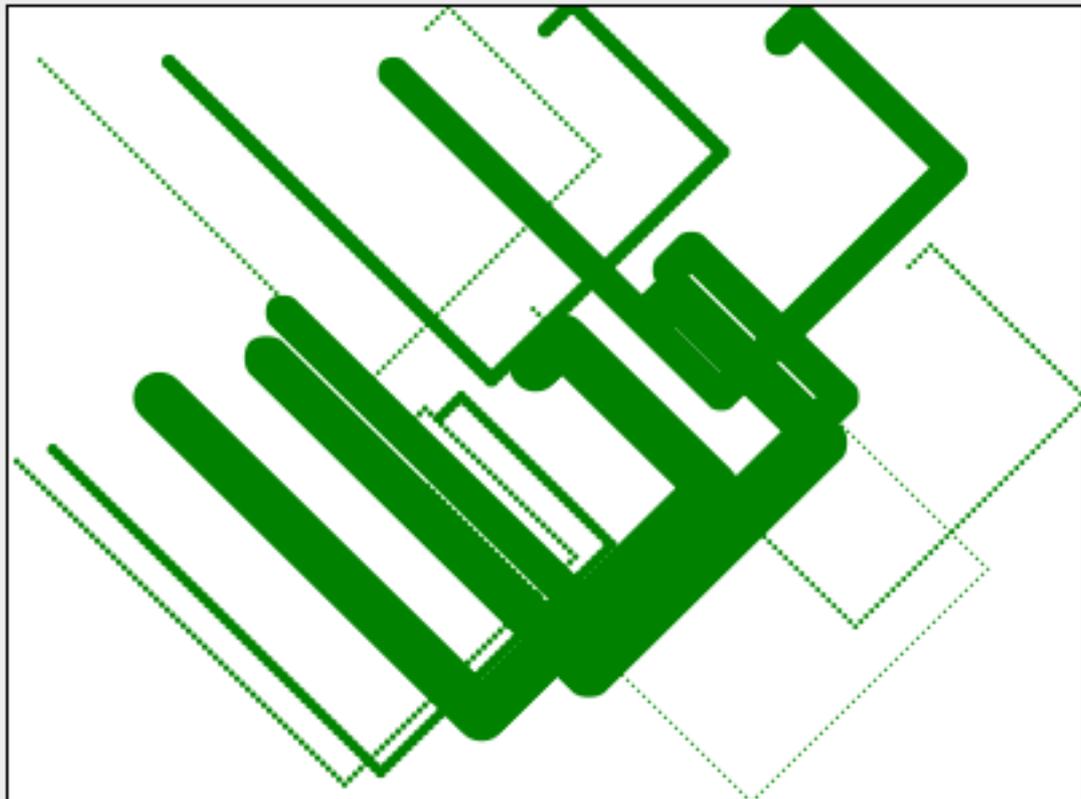
```
}
```

```
}
```

```
setInterval(draw,100);
```

```
</script>
```

Risultato strano...



Pulire la canvas ogni volta!

```
context.clearRect(0,0,400,300);
```

Inserire questa istruzione all'inizio della funzione draw()



Cambiare colore

- La seguente funzione Javascript permette di avere dei colori random.

```
function getRandomColor() {  
  var letters = '0123456789ABCDEF'.split("");  
  color = '#';  
  for (var i = 0; i < 6; i++ ) {  
    color += letters[Math.floor(Math.random() * 16)];  
  }  
  return color;  
}
```

Ora tocca a voi!

- Come inseriamo la chiamata a questa funzione per ottenere palline di tutti i colori?
- Come possiamo fare in modo che le variabili incremento dx e dy siano diversi per ogni pallina in modo da dare proprio un movimento irregolare ed indipendente sulla canvas?

Ora tocca a voi

- Sapendo che il metodo per disegnare un rettangolo è:

context.fillRect(0,0,50,20);

Modificare il nostro sorgente per avere dei rettangolini al posto delle palline