



Добавяне на интерактивност с HTML 5, jQuery и CSS 3

проф. г-р Асен Рахнев, докторант Мая Стоева

СЪДЪРЖАНИЕ

- I. Пояснения към домашна работа № 2
- II. Да направим банер само с CSS3 и HTML5
- III. Използване на HTML5 Canvas API
 - използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)
 - трансформиране на съществуващи картинки с <canvas>
 - използване на Scalable Vector Graphics (SVG) в уеб страниците
 - силните и слабите места на <canvas> и SVG
 - Cross-browser поддръжка

Пояснения към домашна работа № 2






Да се създаде банер, написан само с HTML5 и CSS3, за който:

- 1) Може да използвате като графични елементи, банера, който направихме в началото.
- 2) Задължително да е с размер, избран от стандартните.
- 3) Може да използвате банера, който ще направим след малко като помощен код, но трябва да има поне една различна анимация.
- 4) Да вградите банера в първата домашна: като страница, към която води линк от дом. Работа номер 1, например.

Да направим банер само с CSS3 и HTML5

- 1) Създаване на HTML-частта
- 2) Създаване на CSS стиловете и статичния вид на банера
- 3) Създаване на анимациите за отделните елементи.

Използване на HTML5 Canvas API

Browser support quick check: <canvas>		Canvas 2D context	Canvas text
		4.0	4.0
		2.0	3.5
		9.0	9.0
		9.0	10.5
		3.1	4.0

Използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)

HTML5 Canvas API—а е много интересна и занимателна страна на технологията, която позволява динамично да генерираме и рендираме графики, диаграми, картинки, и анимации. Ще се запознаем и с основите на rendering API, за създаване на графики, които после можем да скалираме и адаптираме към средата на браузъра.

Използване на `<canvas>` елемента за рисуване на форми (*shapes*), текст (*text*), и картинки (*images*)

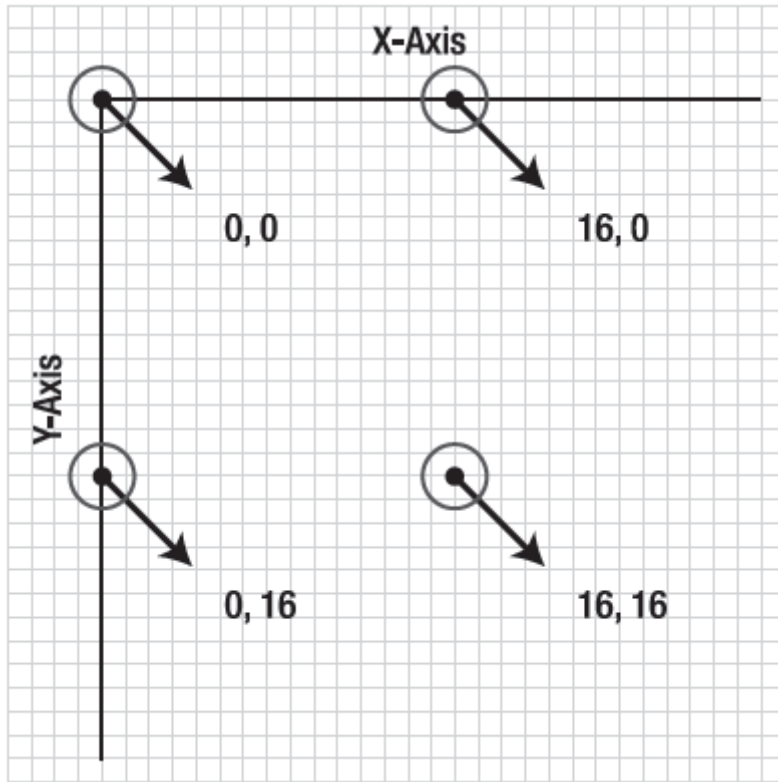
- концепцията за Canvas, създадена в началото от Apple за нуждите на Mac OS X WebKit от създаване на *dashboard widgets*
- Canvas всъщност е *bitmap canvas*, и финалният резултат не може да се скалира, както Scalable Vector Graphic (SVG) картинките могат. Обектите, нарисувани върху canvas не са част от DOM или който и да е друг namespace — нещо, което се смята за слабост. SVG картинките, от друга страна се мащабират “безкрайно” под различните резолюции и позволяват прихващания като клик на мишката и то точно къде е станало това.

Използване на `<canvas>` елемента за рисуване на форми (`shapes`), текст (`text`), и картинки (`images`)

-Но HTML5 Canvas API има 2 предимства, за които си струва да го използваме: изпълнява се добре тъй като не е нужно да съхранява обекти за всеки примитив, който се изрисува; лесно е да се имплементира, на базата на много популярни двумерни API-та за рисуване, познати от другите езици за програмиране.”

- Когато използвате **canvas** елемента във вашата уеб страница, се създава правоъгълна област с размери по подразбиране 300 px широка и 150 px висока, но тя може да се промени и да се зададат и други атрибути.

Използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)



`<canvas></canvas>`

`/*` за браузъри, които не го поддържат задавайте пояснения:

`<canvas>`

Обновете своя браузър, да се насладите на canvas ефектите!

`</canvas>`

`*/`

Използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)

Проверка на поддръжката:

```
try {  
    document.createElement("canvas").getContext("2d");  
    document.getElementById("support").innerHTML =  
        "HTML5 Canvas is supported in your browser.";  
} catch (e) {  
    document.getElementById("support").innerHTML = "HTML5  
    Canvas is not supported in your browser.";  
}
```

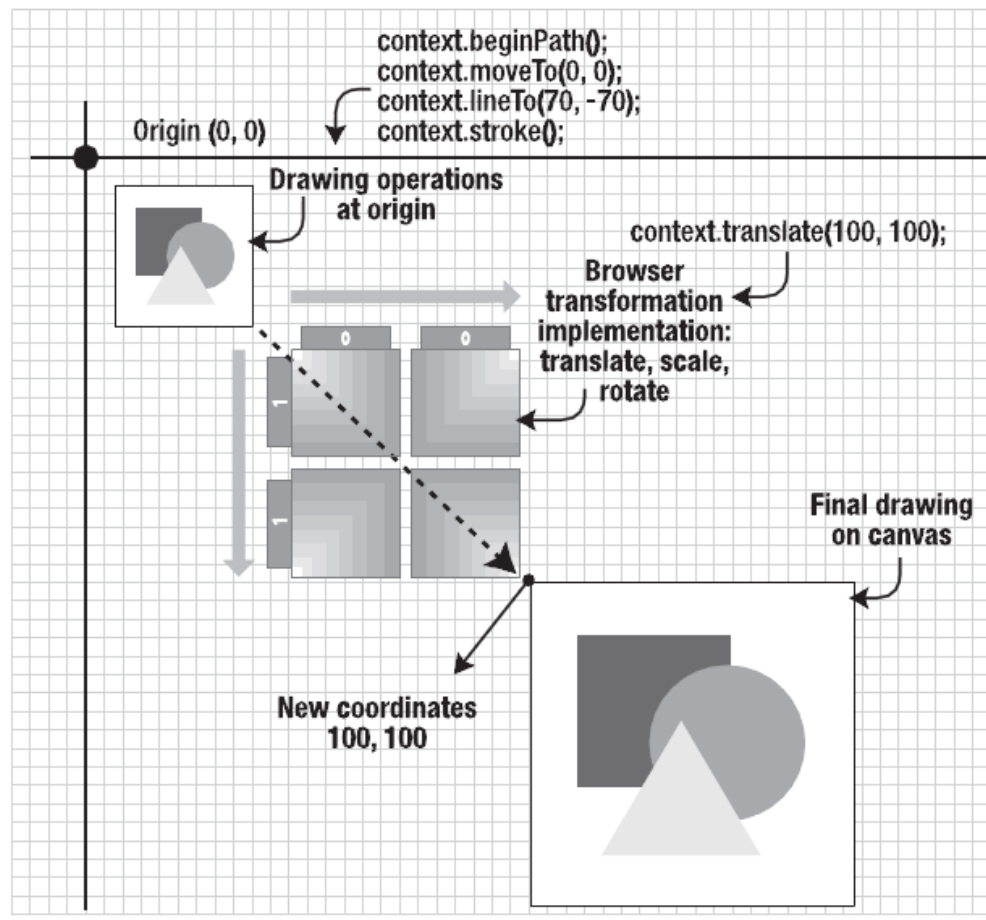
Използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)

Проверка на поддръжката:

```
try {  
  document.createElement("canvas").getContext("2d");  
  document.getElementById("support").innerHTML =  
  "HTML5 Canvas is supported in your browser.";  
} catch (e) {  
  document.getElementById("support").innerHTML = "HTML5  
Canvas is not supported in your browser."  
}
```

Използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)

- Използване на *javascript* функция за рисуване на диагонал
- Използване на трансформации за същата линия

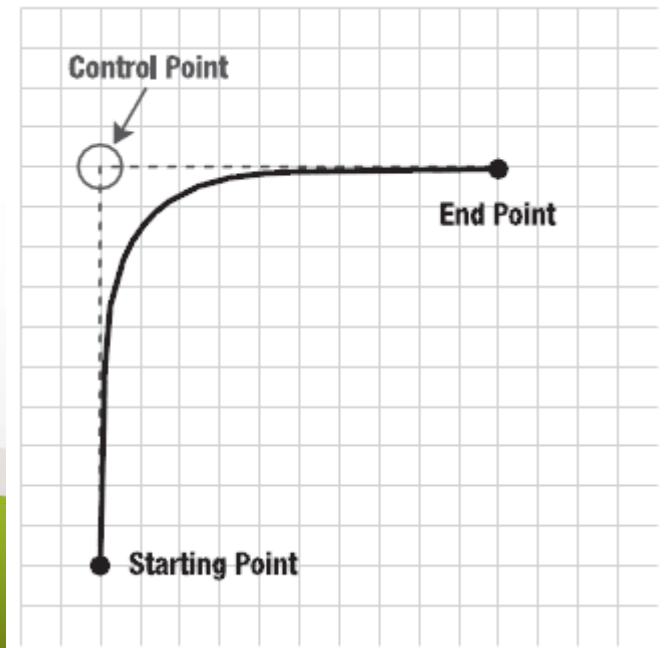


Използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)

- Работа с пътища (Paths)

lineJoin: bevel, miter или round

lineCap: butt butt, square, или round



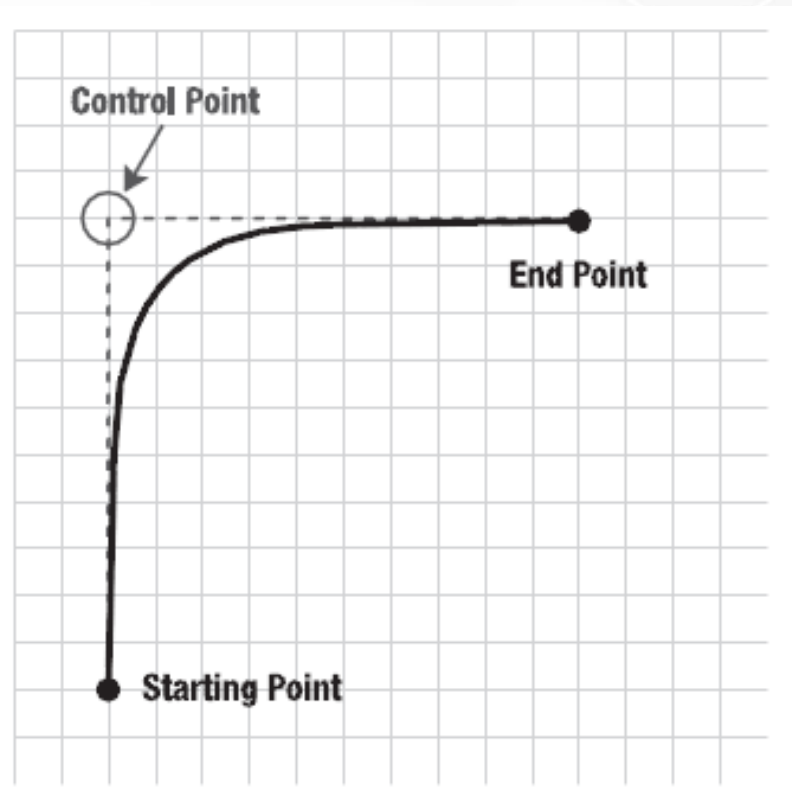
Quadratic крива: начало, контролни точки

Използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)

- Работа с криви

Quadratic крива:
начало, край, и
контролни точки

Други възможни криви
са: `bezierCurveTo`,
`arcTo`, и `arc`.



Използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)

- Вмъкване на картинка в Canvas
- Използване на градиенти
- Използване на Background Patterns
- Мащабиране
- Използване на Canvas Transforms – операции и на завъртане **context.rotate(angle)**

```
context.save();  
// ъгъл на завъртане в радиани  
context.rotate(1.57);  
context.drawImage(myImage, 0, 0, 100, 100);  
context.restore();
```

Използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)

- Използване на Canvas Text

fillText (text, x, y, maxwidth)

strokeText (text, x, y, maxwidth)

Property	Values	Note
font	CSS font string	Example: italic Arial, sans-serif
textAlign	start, end, left, right, center	Defaults to start
textBaseline	top, hanging, middle, alphabetic, ideographic, bottom	Defaults to alphabetic

Използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)

- Добавяне на сянка shadow към Canvas Text

Property	Values	Note
shadowColor	Any CSS color	Can include an alpha component
shadowOffsetX	Pixel count	Positive values move shadow to the right, negative left
shadowOffsetY	Pixel count	Positive values move shadow down, negative up
shadowBlur	Gaussian blur	Higher values cause blurrier shadow edges

Използване на <canvas> елемента за рисуване на форми (shapes), текст (text), и картинки (images)

- Работа с Pixel Data или достъп до пикселите на нашето пано или Canvas 😊

context.getImageData(sx, sy, sw, sh)

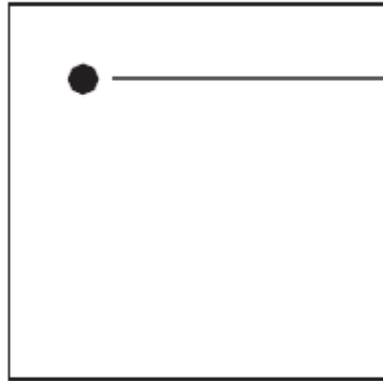
// връща текущото състояние като колекция от integer

Red component: $((width * y) + x) * 4$

Green component: $((width * y) + x) * 4 + 1$

Blue component: $((width * y) + x) * 4 + 2$

Alpha component: $((width * y) + x) * 4 + 3$



10x10 pixel canvas
with 1 black pixel

R G B A
00 00 00 FF

RGBA Value for
Black Pixel (2,2)

Image Operations:
-createImageData
-getImageData
-putImageData

Internal data
structure for
image pixels on
the canvas

```
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFF  
FFFFFFFF000255FFFFFF
```