

JavaScript

Introducción y sintáxis básica Canvas, página web de Rectángulos

Jhon Gesell Villanueva Portella¹

¹Toulouse Lautrec.
Diseño.

Diseño y Desarrollo para Medios Digitales.

15 de julio de 2020

Contenido

1 Introducción

2 Página web de Rectángulos

Contenido

1 Introducción

2 Página web de Rectángulos

Introducción 01/02

El lienzo le permite dibujar objetos gráficos (por ejemplo, líneas, rectángulos, círculos) y animarlos, el lienzo está integrado en todos los navegadores modernos, por lo que no hay que instalar un software adicional para que los usuarios finales puedan tratar.

Web Programming with html5, css, and javascript de John Dean (2019), pág. 570

Canvas es un elemento HTML incorporado en HTML5 que permite la generación de gráficos dinámicamente por medio del scripting. Entre otras cosas, permite la renderización interpretada dinámica de gráficos 2D y mapas de bits, así como animaciones con estos gráficos. Wikipedia

Wikipedia (2020)

Introducción 02/02

El lienzo le permite dibujar objetos gráficos (por ejemplo, líneas, rectángulos, círculos) y animarlos, el lienzo está integrado en todos los navegadores modernos, por lo que no hay que instalar un software adicional para que los usuarios finales puedan tratar.

Web Programming with html5, css, and javascript de John Dean (2019), pág. 570

Sintaxis básica canvas 01/03

Para usar el lienzo, necesita (1) un elemento del lienzo y (2) llamadas al método JavaScript que dibujen objetos gráficos dentro del área de dibujo del elemento del lienzo. Aquí hay un elemento de lienzo de ejemplo, que crea un área de dibujo rectangular en blanco de 480 píxeles por 250 píxeles en una página web:

```
<body onload="rectangleExamples();">
<canvas id="canvas" width="480" height="250">
  Sorry - This page uses <code>canvas</code> and
  your browser doesn't support it.
</canvas>
</body>
```

Web Programming with html5, css, and javascript de John Dean (2019), pág. 571

Sintaxis básica canvas 02/03

Después de que se carga la página web, el evento onload se activa y llama a la función `rectangleExamples`, que dibuja formas rectangulares dentro del área de dibujo del lienzo. Examinaremos la función `rectangleExamples` en la siguiente sección. Como habrás adivinado, el atributo `id` del elemento del lienzo permite que la función acceda al objeto del lienzo. Tenga en cuenta el texto que aparece entre las etiquetas del elemento del lienzo. Ese es el contenido alternativo. Se muestra cuando el navegador del usuario no admite el elemento de lienzo. Si necesita admitir navegadores antiguos, debe incluir dicho contenido alternativo, pero dado que todos los navegadores modernos admiten lienzo,

Web Programming with html5, css, and javascript de John Dean (2019), pág. 571

Sintaxis básica canvas 03/03

Hay dos tipos de comandos de dibujo de lienzo: los que dibujan imágenes bidimensionales y los que dibujan imágenes tridimensionales. Vamos a mantener las cosas simples y seguir con las imágenes bidimensionales. Para crear imágenes bidimensionales, recupera el contexto bidimensional del lienzo de esta manera:

```
var ctx = canvas.getContext("2d");
```

Normalmente, debe mantenerse alejado de las abreviaturas oscuras para nombres de variables, pero ctx es una abreviatura estándar para el contexto de un lienzo.

Web Programming with html5, css, and javascript de John Dean (2019), pág. 572

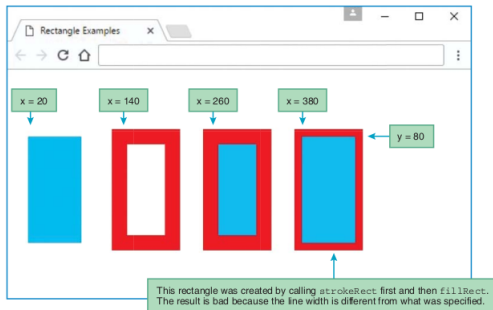
Contenido

1 Introducción

2 Página web de Rectángulos

Página web Rectángulos 01/04

Muestra cuatro rectángulos con el fin de ilustrar varios conceptos de dibujo. Al examinar la página web, describiremos cómo dibujar un rectángulo y especificaremos su posición, altura, ancho, color y ancho del borde.



Web Programming with html5, css, and javascript de John Dean (2019), pág. 573

Página web Rectángulos 02/04

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="author" content="John Dean">
<title>Rectangle Examples</title>
<script>
function rectangleExamples() {
    var ctx; // the canvas object's context
    ctx = document.getElementById("canvas").getContext("2d");
    ctx.fillStyle = "deeppskyblue";
    ctx.fillRect(20, 80, 70, 140);

    ctx.lineWidth = 20;
    ctx.strokeStyle = "red";
    ctx.strokeRect(140, 80, 70, 140);

    ctx.fillRect(260, 80, 70, 140);
    ctx.strokeRect(260, 80, 70, 140);

    ctx.strokeRect(380, 80, 70, 140);
    ctx.fillRect(380, 80, 70, 140);
} // end rectangleExamples
</script>
</head>

<body onload="rectangleExamples();">
<canvas id="canvas" width="480" height="250">
    Sorry - This page uses <code>canvas</code>, and
    your browser doesn't support it.
</canvas>
</body>
</html>
```

Normally, it's inappropriate
to call `strokeRect` first
and then `fillRect`.

Página web Rectángulos 03/04

Regrese a la ventana del navegador de la página web Rectángulos y observe que los rectángulos primero, tercero y cuarto tienen interiores llenos de azul. Para dibujar un rectángulo relleno, utiliza el objeto de contexto para llamar al método `fillRect`. Aquí está la sintaxis:

```
context.fillRect(x, y, width, height);
```

Web Programming with html5, css, and javascript de John Dean (2019), pág. 574

Página web Rectángulos 04/04

Los argumentos x e y son enteros que especifican la posición de coordenadas x , y de la esquina superior izquierda del rectángulo que se va a dibujar. Si x es igual a 0 e y es igual a 0, entonces la esquina superior izquierda del rectángulo coincidirá con la esquina superior izquierda del área del lienzo. Los argumentos de ancho y alto son enteros que especifican el ancho y el alto del rectángulo. Para los valores de posición x e y , y los valores de dimensión de ancho y alto, JavaScript asume unidades de píxeles y no especifica unidades explícitamente.

Web Programming with html5, css, and javascript de John Dean (2019), pág. 574

Fill Color 01/05

Aquí está el código que dibuja el primer rectángulo en la página web Rectángulos:

```
ctx.fillStyle = "deepskyblue";  
ctx.fillRect(20, 80, 70, 140);
```

Web Programming with html5, css, and javascript de John Dean (2019), pág. 574

Fill Color 02/05

La variable `ctx` contiene el objeto de contexto para el área de dibujo del lienzo de la página web. Observe cómo asignamos "deepskyblue."^a la propiedad `fillStyle` del objeto de contexto. Cuando asigna un valor a la propiedad `fillStyle` del objeto de contexto, significa que las siguientes llamadas al método de dibujo de forma utilizarán el color especificado al rellenar la forma dibujada. Entonces, en el código anterior, cuando el objeto de contexto llama al método `fillRect`, el motor de JavaScript dibuja un rectángulo con un color de relleno de azul cielo profundo. Vea la ventana del navegador y observe el azul para los colores de relleno de los rectángulos. Mientras lo hace, tenga en cuenta la posición y el tamaño del primer rectángulo y verifique que se desprenden de los argumentos 20, 80, 70 y 140 de la llamada al método `fillRect`. Recuerde que los dos primeros argumentos especifican la posición del rectángulo en relación con las áreas de dibujo del lienzo. Con valores de 20 y 80, la esquina superior izquierda del primer rectángulo se coloca 20 píxeles a la derecha del borde izquierdo del área del lienzo y 80 píxeles hacia abajo desde el borde superior del área del lienzo. Los siguientes dos argumentos especifican las dimensiones del rectángulo. Con valores de 70 y 140, el ancho del primer rectángulo es de 70 píxeles y su altura es de 140 píxeles.

Web Programming with html5, css, and javascript de John Dean (2019), pág. 574

Fill Color 03/05

El color de dibujo predeterminado es el negro. Asignar un valor a la propiedad `fillStyle` cambia el color de relleno, y el motor de JavaScript usa ese nuevo color para todas las formas dibujadas en el futuro hasta que el navegador ejecute una nueva asignación `fillStyle`. La propiedad `fillStyle` afecta no solo a formas estándar como rectángulos y círculos, sino también a caracteres de texto.

Web Programming with html5, css, and javascript de John Dean (2019), pág. 574

Fill Color 04/05

Al asignar colores para formas de lienzo, puede usar cualquiera de los formatos de valor de color presentados anteriormente en el capítulo CSS. Aquí están los formatos válidos:

- nombre de color.
- Valor RGB: especifica cantidades de rojo, verde y azul.
- Valor RGBA: especifica RGB, más la cantidad de opacidad.
- Valor HSL: especifica cantidades de matiz, saturación y luminosidad.
- HSLA: especifica HSL, más la cantidad de opacidad.

Web Programming with html5, css, and javascript de John Dean (2019), pág. 574

Fill Color 05/05

Para los formatos RGB, los valores rojo, verde y azul deben ser (1) tres enteros entre 0 y 255 o (2) tres porcentajes entre 0 % y 100 %. Para los formatos HSL, los valores de matiz, saturación y luminosidad deben ser un número entero entre 0 y 360 (para grados en la rueda de color de matiz), un porcentaje entre 0 % y 100 % y otro porcentaje entre 0 % y 100 %. El rango válido para la opacidad es un número entre 0.0 (completamente transparente) y 1.0 (completamente opaco).

Web Programming with html5, css, and javascript de John Dean (2019), pág. 574

Bordes de rectángulo 01

Mire nuevamente la ventana del navegador de la página web Rectángulos, y esta vez observe los bordes rojos alrededor del segundo, tercer y cuarto rectángulos. Para dibujar un borde rectangular, use el método `strokeRect` del objeto de contexto. Aquí está su sintaxis:

```
context.strokeRect(x, y, width, height);
```

Web Programming with html5, css, and javascript de John Dean (2019), pág. 575

Bordes de rectángulo 02

Los argumentos x , y , ancho y alto funcionan igual que con el método `fillRect`. Aquí está la llamada al método `strokeRect` para dibujar el segundo rectángulo de la página web `Rectangle`:

```
ctx.strokeRect(140, 80, 70, 140);
```

Web Programming with html5, css, and javascript de John Dean (2019), pág. 575

Bordes de rectángulo 03

Si vuelve a la ventana del navegador, puede ver que la esquina superior izquierda del segundo rectángulo está justo arriba y a la izquierda del punto $x = 140$, $y = 80$ especificado por la llamada al método `strokeRect`.

¿Por qué no se coloca la esquina del rectángulo en el punto $x = 140$, $y = 80$ con precisión? Si el ancho del borde era de 1 píxel, entonces la esquina superior izquierda del rectángulo estaría en $x = 140$, $y = 80$. Pero con un borde grueso, la mitad va fuera del borde del rectángulo especificado y la otra mitad va en el interior.

Aquí está el código que aparece justo encima de la llamada al método `strokeRect` del segundo rectángulo:

```
ctx.lineWidth = 20;  
ctx.strokeStyle = "red";
```

Web Programming with html5, css, and javascript de John Dean (2019), pág. 575

Contenido

3 Referencias

Referencias



Web Programming with html5, css, and javascript

Dean, John.

Jones & Bartlett Learning (2019)

Referencias



Learn to Code HTML and CSS: Develop and Style Websites

Howe, Shay.

New Riders, 2014