JavaScript Canvas

Summary: in this tutorial, you'll learn about HTML Canvas and how to use JavaScript to draw on the canvas.

Introduction to the HTML5 Canvas element

HTML5 features the `<canvas>` element that allows you to draw 2D graphics using JavaScript.

The `<canvas>` element requires at least two attributes: `width` and `height` that specify the size of the canvas:

```
<canvas width="500" height="300" id="canvas"></canvas>
```

Like other elements, you can access the `width` and `height` properties of the `<canvas>` element via its DOM properties:

```
const canvas = document.querySelector('#canvas');
const width = canvas.width;// 500
const height = canvas.height;// 300
```

And you can also change the `width` and `height` of the `<canvas>` element using the DOM methods:

```
canvas.width = 600;
canvas.height = 400;
```

Fallback content

Unlike the `<img>` element, The `<canvas>` element requires the closing tag `</canvas>`. Any content between the opening and closing tags is fallback content that will display only if the browser doesn't support the `<canvas>` element. For example:

```
<canvas width="500" height="300" id="canvas">The browser doesn't support the canvas element</
```

Nowadays, most modern web browsers support the `<canvas>` element.

The rendering context

Initially, the canvas is blank. To draw something, you need to access the rendering context and use it to draw on the canvas.

The `<canvas>` element features the `getContext()` method that returns a render context object.

The `getContext()` takes one argument which is the type of context. For example, you use the `"2d"` to get a 2D rendering context object, which is an instance of the `CanvasRenderingContext2D` interface.

The 2D rendering context allows you to draw shapes, text, images, and other objects.

The following example shows how to select the canvas element using the `querySelector()` method and access the drawing context by calling its `getContext()` method:

```
let canvas = document.querySelector('#canvas');
let ctx = main.getContext('2d');
```

Check for canvas support

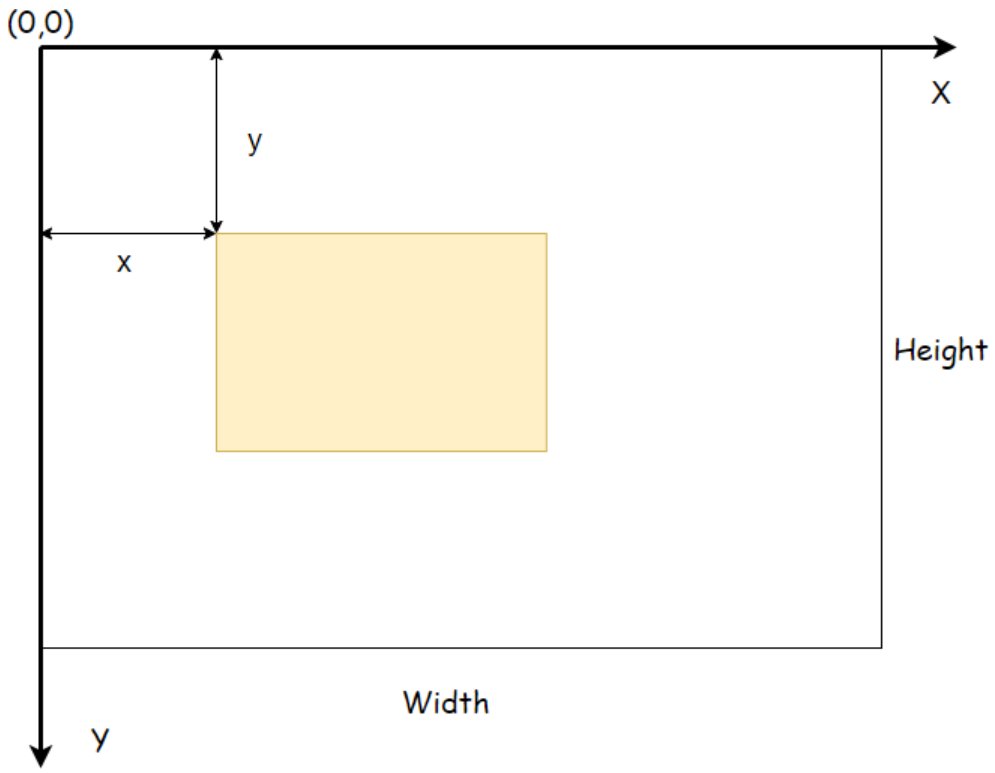
When using the `<canvas>` element, it's important to check if the browser supports the `getContext()` method. To do it, you use the following code:

```
let canvas = document.querySelector('#main');
if(canvas.getContext) {
   let ctx = main.getContext('2d');
}
```

The 2D context

The 2D drawing context features methods for drawing simple 2D shapes such as paths, rectangles, and arcs.

The coordinates in a 2D context begin at the upper-left of the `<canvas>` element, which is point (0,0) as shown in the following picture:



All coordinate values are calculated in relation to the `(0,0)` with `x` increasing to the right and `y` increasing to the bottom.

By default, the `width` and `height` determine the number of pixels is available in each direction.

Fills and strokes

Fill and stroke are two basic drawing operation on 2D drawing context.

- Fill fills in the shape with a specific style such as color, gradient, and image.
- Stroke adds colors to the edges of the shape.

The `fillStyle` and `strokeStyle` properties of the 2D drawing context will determine the fill and stroke styles.

You can set these properties to a string, a gradient object, or a pattern object. By default, they both set to a value of `'#000000'` .

The following illustrates how to set the fill and stroke styles for the 2D drawing context and draw a rectangle.

```
(() => {
    const canvas = document.querySelector('#main');
    if (!canvas.getContext) {
        return;
    }

    // get the context
    let ctx = canvas.getContext('2d');

    // set fill and stroke styles
    ctx.fillStyle = '#F0DB4F';
    ctx.strokeStyle = 'red';

    // draw a rectangle with fill and stroke
    ctx.fillRect(50, 50, 150, 100);
    ctx.strokeRect(50, 50, 150, 100);

})();
```

Click here to see the canvas demo.

Note that you'll learn how to draw a rectangle in a later tutorial.

Summary

- Use the HTML5 canvas element for drawing 2D graphics.
- Use the `getContext('2d')` to get the 2D drawing context for drawing 2D graphics on canvas.
- Use the `fillStyle` and `StrokeStyle` properties to set the styles for the 2D drawing context.