Manipulating Pixels

Dan Wahlin

@DanWahlin

http://weblogs.asp.net/dwahlin

pluralsight
hardcore developer training

Agenda

- Rendering Gradients
- Using Transforms
- Accessing Pixels
- Animation Fundamentals

Agenda

- Rendering Gradients
- Using Transforms
- Accessing Pixels
- Animation Fundamentals

Gradient Functions

addColorStop	drawImage	restore
arc	fill	rotate
arcTo	fillRect	save
beginPath	fillText	scale
bezierCurveTo	getImageData	setTransform
clearRect	isPointInPath	stroke
clip	lineTo	strokeRect
closePath	measureText	strokeText
createImageData	moveTo	toDataUrl
createLinearGradient	putImageData	transform
createPattern	quadraticCurveTo	translate
createRadialGradient	rect	

Gradient Functions

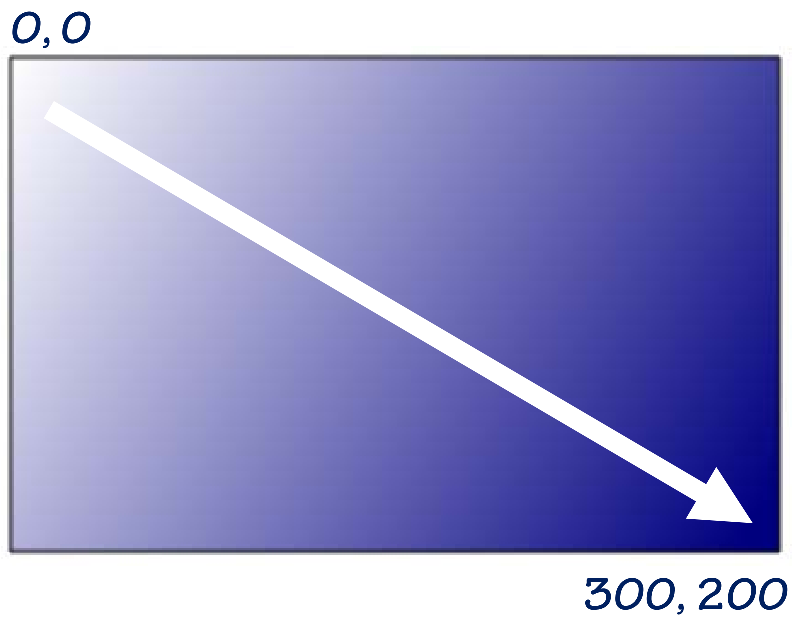
Functions:

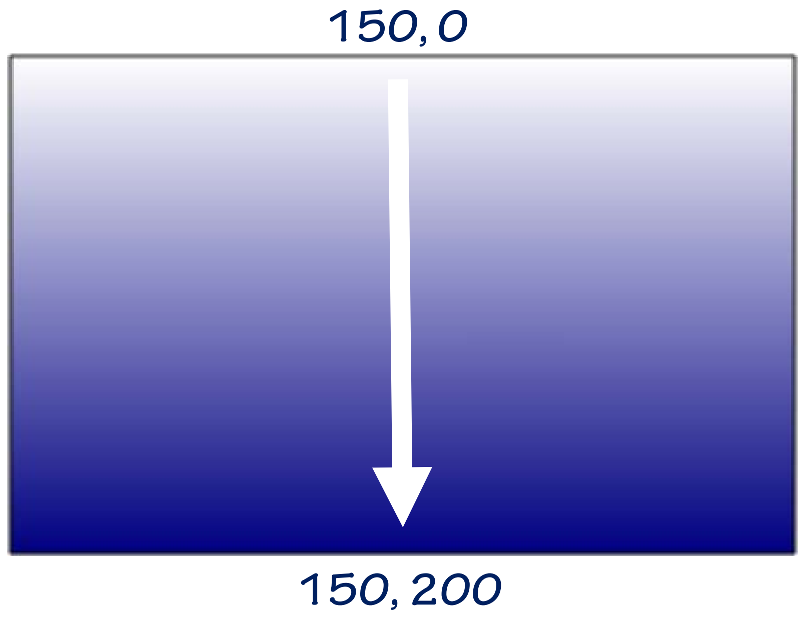
```
addColorStop(position, color);

createLinearGradient(x1, y1, x2, y2)

createRadialGradient(x1, y1, radius1,
                     x2, y2, radius2)
```

Properties:

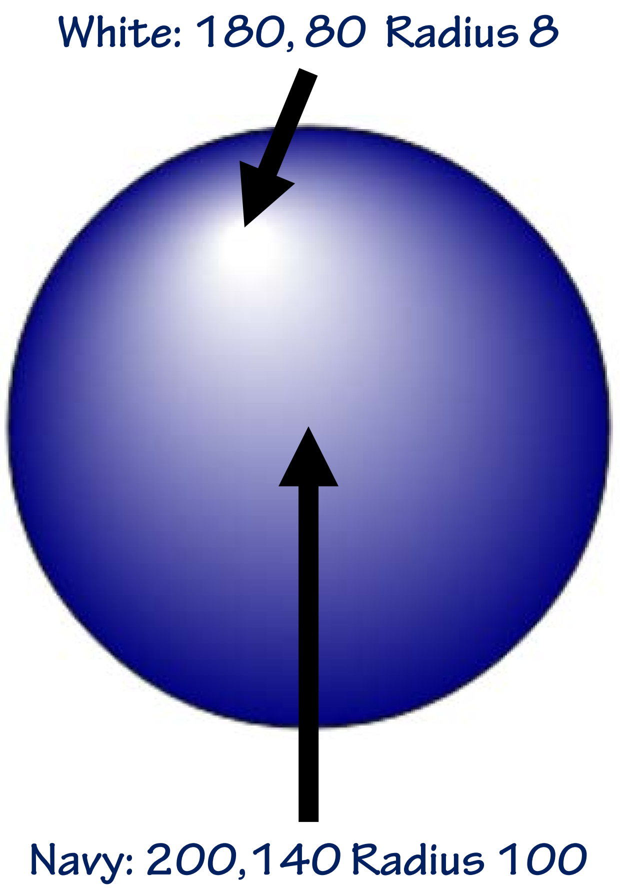
```
fillStyle, strokeStyle
```

Linear Gradient Coordinates





Radial Gradient Coordinates



Agenda

- Rendering Gradients
- Using Transforms
- Accessing Pixels
- Animation Fundamentals

Transform Functions

addColorStop	drawImage	**restore**
arc	fill	**rotate**
arcTo	fillRect	**save**
beginPath	fillText	**scale**
bezierCurveTo	getImageData	**setTransform**
clearRect	isPointInPath	stroke
clip	lineTo	strokeRect
closePath	measureText	strokeText
createImageData	moveTo	toDataUrl
createLinearGradient	putImageData	**transform**
createPattern	quadraticCurveTo	**translate**
createRadialGradient	rect	

Transform Functions

Functions:

```
restore()

rotate(angle)

save()

scale(x, y)

translate(x, y)
```

The Role of Matrices in Transformations

Matrix algebra used to calculate x and y coordinates when doing transformations:

$$\begin{bmatrix} \text{scale-x} & \text{skew-y} & \text{tx} \\ \text{skew-x} & \text{scale-y} & \text{ty} \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix Transform Functions

Functions:

```
setTransform(scale-x, skew-x,
             skew-y, scale-y,
             translate-x, translate-y)


transform(scale-x, skew-x,
          skew-y, scale-y,
          translate-x, translate-y)
```

Agenda

- Rendering Gradients
- Using Transforms
- Accessing Pixels
- Animation Fundamentals

Pixel Functions

addColorStop
arc
arcTo
beginPath
bezierCurveTo
clearRect
clip
closePath
createImageData
createLinearGradient
createPattern
createRadialGradient

drawImage
fill
fillRect
fillText
getImageData
isPointInPath
lineTo
measureText
moveTo
putImageData
quadraticCurveTo
rect

restore
rotate
save
scale
setTransform
stroke
strokeRect
strokeText
toDataUrl
transform
translate

Manipulating Pixels

- **The HTML5 canvas allows pixels to be manipulated or created using built-in functions**
  - Create backgrounds dynamically
  - Change hue, contrast, etc.
  - Convert to grayscale
  - Sharpen colors
  - Perform any pixel-related functionality

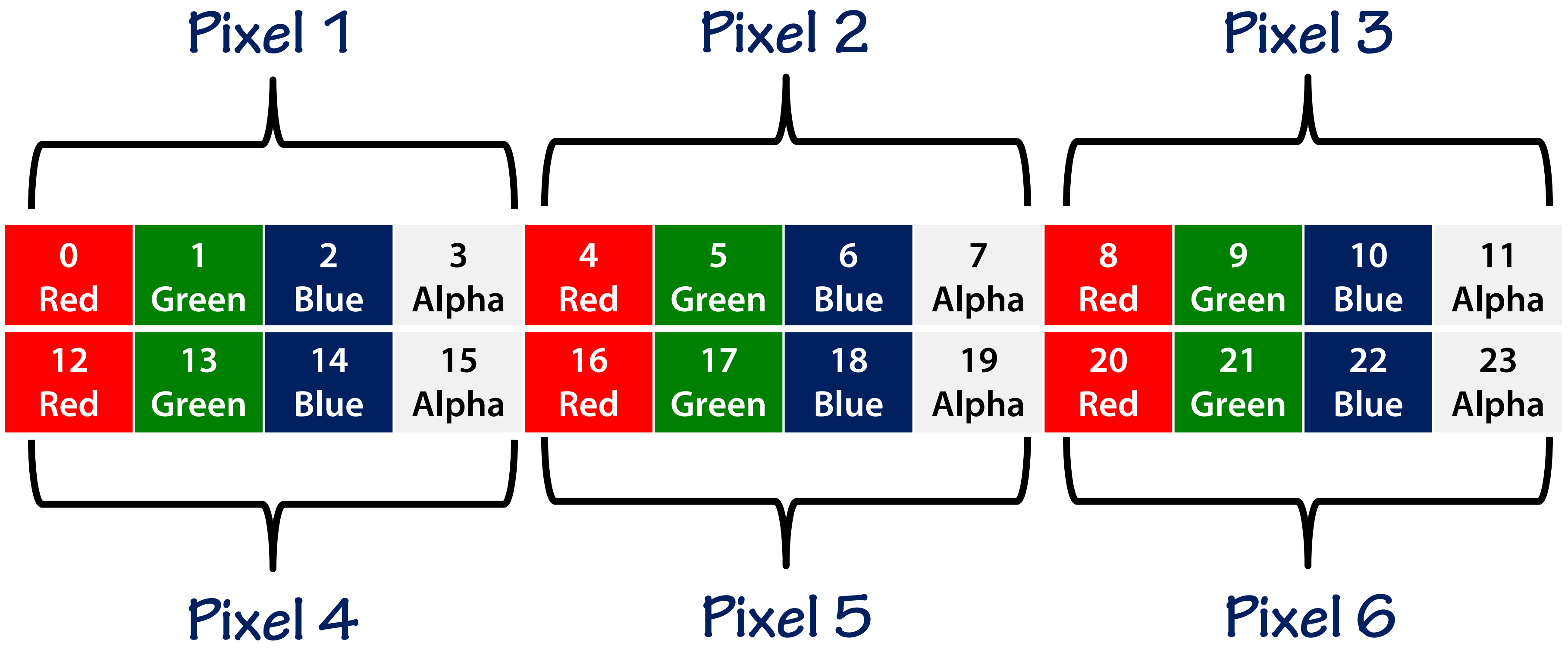
- **Any images loaded must be from the origin domain for pixel functions to work properly**

Pixel Functions

Functions:

```
createImageData(width, height)

createImageData(imgData);

getImageData(x, y, width, height)

putImageData(imgData, dx, dy, x, y,
             width, height)
```

Understanding Pixels



Iterating through Pixels

```
var pixelData =
  ctx.createImageData(200, 200);

for (var i = 0; i < pixelData.length; i+=4) {
    var r = pixelData[i];
    var g = pixelData[i+1];
    var b = pixelData[i+2];
    var a = pixelData[i+3];
    //manipulate pixel data
}

ctx.putImageData(pixelData, 0, 0);
```

Agenda

- Rendering Gradients
- Using Transforms
- Accessing Pixels
- Animation Fundamentals

Animation Fundamentals

- **HTML5 canvas doesn't natively support animations**

- **Animation techniques:**
  - Timer
  - Request animation frame

- **Animation steps:**
  1. Timer fires or frame requested
  2. Update positions
  3. Clear canvas
  4. Draw

Animation with a Timer

- Animation can be started using window.setInterval():

```
window.setInterval(function() {
    // Update
    // Clear
    // Draw
 }, milliseconds);
```

Animation with Frames

- Browsers that support requestAnimationFrame provide more efficient animations
- Browser determines optimal frames per second (FPS)

```
function animate(){
    // Update
    // Clear
    // Draw
    // Request new frame
    window.requestAnimationFrame(function() {
        animate();
    });
}
```

Animation with Frames Shim

- **Cross-browser requestAnimationFrame calls can be handled using a shim:**

```
window.requestAnimFrame = (function(callback) {
    return window.requestAnimationFrame ||
    window.webkitRequestAnimationFrame  ||
    window.mozRequestAnimationFrame     ||
    window.oRequestAnimationFrame       ||
    window.msRequestAnimationFrame      ||
    function(callback){
        window.setTimeout(callback, 1000 / 60);
    };
})();
```

Summary

- **HTML5 canvas supports the following pixel-related features:**
  - Rendering dynamic linear and radial gradients
  - Transformations (scale, skew, rotate, translate)
  - Direct access to pixels

- **Animations are not directly supported but can be implemented:**
  - Timers
  - Request animation frame