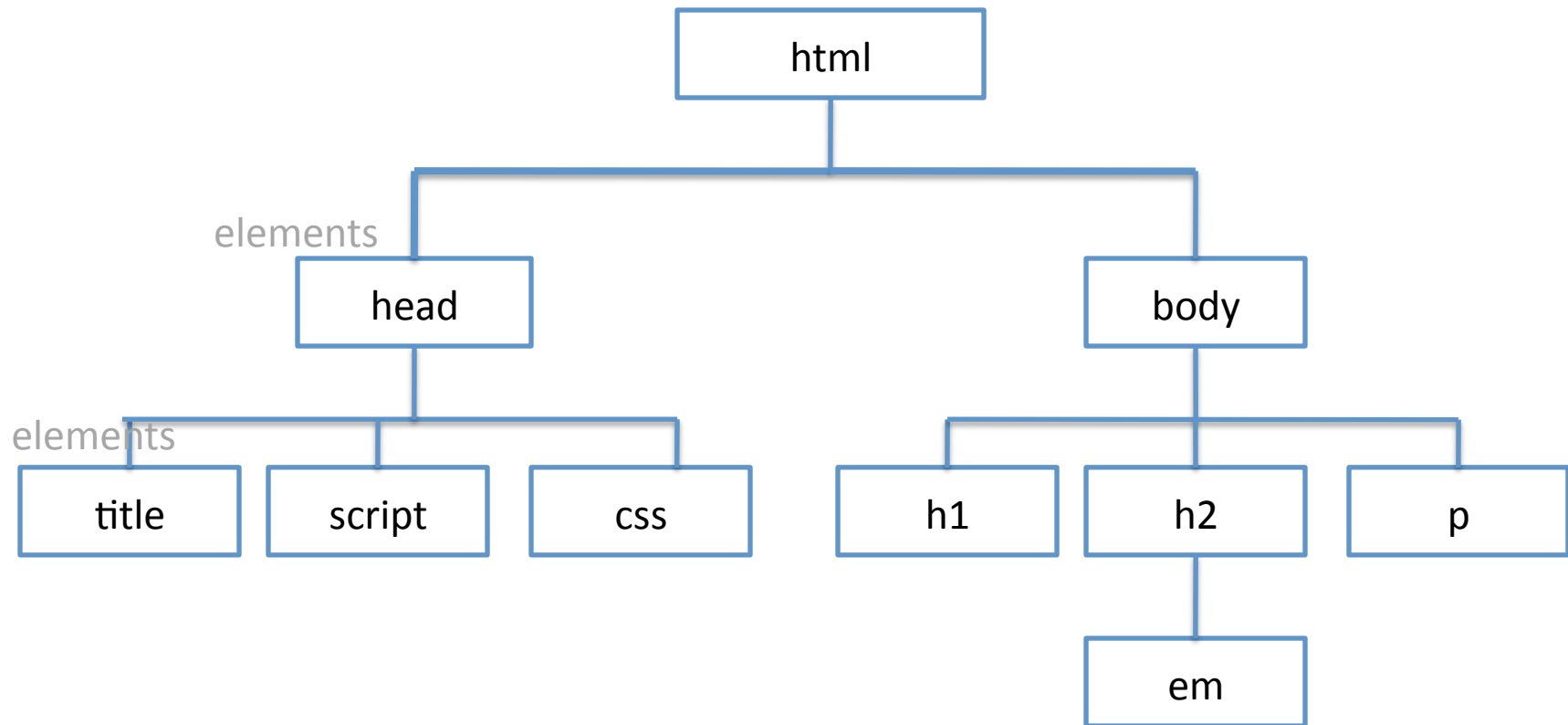


Using Javascript with Canvas

DOM (Document Object Model)



Canvas and Javascript

```
<script>
var canvas
=document.getElementById
  ("mycanvas");

var context =
canvas.getContext("2d");
context.fillRect
(10, 10, 100, 100);
</script>
```

```
<body>

<canvas width="600"
  height="200"
  id="myCanvas">
</canvas>

</body>
```

Canvas and Javascript

Here's our 100 x 100 rectangle,
positioned at 10, 10 in the canvas.



And our canvas, which is 600 wide, 200 high
and has a 1 pixel black border around it.

Canvas and Javascript

```
<!doctype html>
<html lang="en">
<head>
<title>Look What I Drew</title>
<meta charset="utf-8" /> <style>
canvas { border: 1px solid black; } </style>

<script> window.onload = function() {
var canvas = document.getElementById("myCanvas");
var context = canvas.getContext("2d"); context.fillRect(10, 10, 100, 100);
}; </script>

</head>

<body>
<canvas width="600" height="200" id="myCanvas">
</canvas>
</body>
</html>
```

HTML Coding Step 0

0. Create a canvas element on the body

```
<body>
```

```
<canvas id="canvas" width="500px"  
  height="500px">
```

```
</body>
```

Javascript Coding Step 1

1. Have the canvas draw when the window has loaded (DOM is ready).

```
window.onload= function(){  
... draw  
}
```

Javascript Coding Step 2

2. Access the Canvas Element

```
var canvas =  
document.getElementById("canvas");
```

Javascript Coding Step 3

3. Obtain the 2d context of the canvas element

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

Javascript Coding Step 4

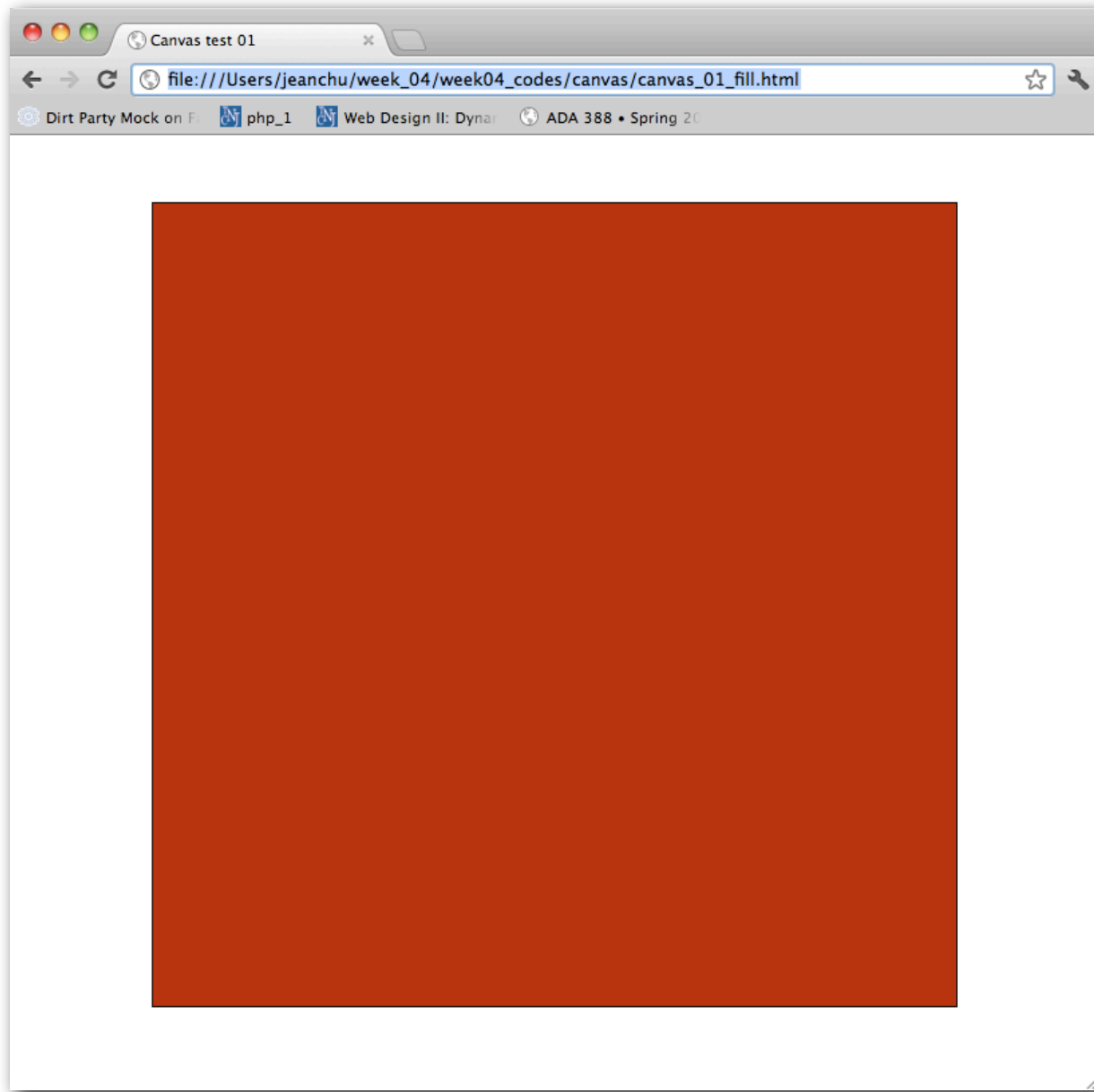
4. Have my ctx object render

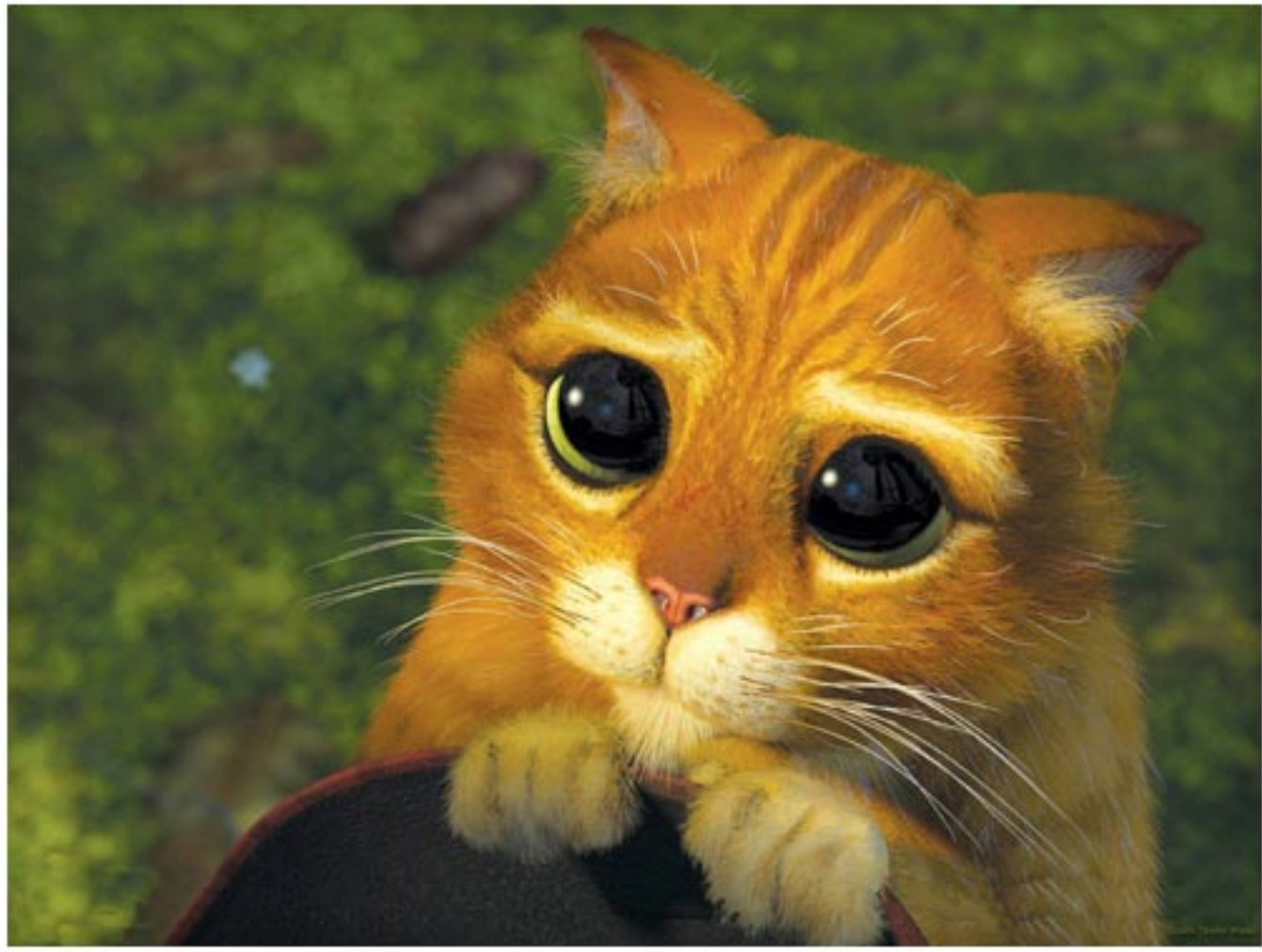
```
var w = canvas.width;
```

```
var h = canvas.height;
```

```
ctx.fillStyle = "rgb(200,0,0)";
```

```
ctx.fillRect (0, 0, w,h);
```





I WANT ANIMATION... PLS...

Animation Tip

- Use the setInterval javascript event
 - setInterval is a window object for timing event. It executes a function or an expression over and over again in a specified number of milliseconds

```
setInterval ( dosomething , 10 );
```



expression



millisecond

Step 1

- Create an initial stage and global variables

```
window.onload=function(){  
  
    var ctx = canvas.getContext("2d");  
    var w = canvas.width;  
    var h = canvas.height;  
    var timer=0;  
  
    ctx.fillStyle = "rgb(255,255,255)";  
    ctx.fillRect (0, 0, w,h);  
  
};
```

Step 2

- Call a draw function by **setInterval** to do something every other milliseconds
 - From the draw function, Clear the stage and redraw

```
window.onload=function(){  
    var ctx = canvas.getContext("2d");  
    var w = canvas.width;  
    var h = canvas.height;  
    var timer=0;  
  
    ctx.fillStyle = "rgb(255,255,255)";  
    ctx.fillRect (0, 0, w,h);  
  
    Goes here  
};
```

```
setInterval(draw,10);
```

```
function draw() {  
    var red, green, blue;  
    timer+=1;  
    red=(timer%255);  
    green=((timer+100)%255);  
    blue=((255-timer)%255);  
  
    ctx.clearRect(0, 0, w, h);  
  
    ctx.fillStyle = 'rgb('+red+', '+green+', '+blue+')';  
    ctx.fillRect (0, 0, w,h);  
}
```

Modernizer : an HTML5 detection library

DETECTION



Modernizer

Modernizer is a JavaScript library that detects support for many HTML5 and CSS3 features.



<http://www.modernizr.com/>

Modernizr

```
<script  
  src="modernizr.min.js">  
</script>
```

```
if (Modernizr.canvas) {  
  // Your browser supports canvas  
  // element! Let's draw some shapes!  
}  
  
else {  
  // no native canvas support available ☹️  
  // Maybe I should show an alternate  
  // image or a message about it.  
}
```

References

- Canvas tutorial by Mozilla
 - https://developer.mozilla.org/en/Canvas_tutorial
- Canvas tutorial by Opera
 - <http://dev.opera.com/articles/view/html-5-canvas-the-basics/>
- Canvas reference from W3 School
 - http://www.w3schools.com/html5/html5_ref_canvas.asp

Books

- HTML5 Canvas, Oreilly Media

http://teaching.jeanhochu.com/tcnj/web2/notes/textbooks/html5_css3/HTML5_Canvas.pdf

- Dive into HTML 5 “Let’s call it a draw(ing Surface)

<http://diveintohtml5.info/canvas.html>