

JavaScript animations and jQuery

cs380

Presentations

- Why People Hate Javascript
 - https://docs.google.com/presentation/d/1jEOYE-NnWfkN_eq4-pLpteUvqU2MrXOSbEQeGAEmw4E/edit#slide=id.p
- Javascript and Objects
 - <https://docs.google.com/presentation/d/1-LEYMGonXYuNVE9O8yKA6fOPRnHsssAmutqd3GATxb8/edit#slide=id.p>

Javascript and JQuery

Selection

```
<!DOCTYPE html>
<html>
<body>
<p>An unordered list:</p>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
  var x = document.getElementsByTagName("LI");
  for (i = 0; i < x.length; i++) {
    x[i].style.backgroundColor = "red";
  }
}
</script>
</body>
</html>
```

file:d1.html

```
<!DOCTYPE html>
<html>
<body>
<script src="../JQ/jquery-1.9.1.min.js"></script>
<p>An unordered list:</p>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
<button onclick='$("li").css("background-color", "red")'>
Try it</button>
</body>
</html>
```

file: d2.html, d2.1.html, d3.html

Could this be improved?

More JQuery

```
<!DOCTYPE html>
<html>
<body>
<script src="../../JQ/jquery-1.9.1.min.js"></script>

<p>An unordered list</p>
<ul>
  <li id="coffee" onclick="$('li').css('background-color', 'transparent');$('#coffee').css('background-color', 'brown');">Coffee</li>
  <li id="tea" onclick="$('li').css('background-color', 'transparent');$('#tea').css('background-color', 'gold');">Tea</li>
  <li id="milk" onclick="$('li').css('background-color', 'transparent');$('#milk').css('background-color', 'blue');">Milk</li>
</ul>
</body></html>
```

d4.html

From Lab4

```
<html>
  <head>
    <script>
      document.getElementById("myspan").innerHTML="new content";
    </script>
  </head>
  <body>
    <span id="myspan" onclick="blob()">This is my span</span>
  </body>
</html>
```

- Javascript does not do the expected replace
 - Why?
 - How to fix?
 - Best practice?

file: issue.html

Javascript Properties, also from Lab4

```
<html>
  <body>
    <div id="mydiv">Original content</div>
    <button onclick="doReplace()">Replace</button>

    <script>
      function doReplace() {
        document.getElementById("mydiv").innerHTML="New Content";
        console.log(document.getElementById("mydiv"));
        console.log(document.getElementById("mydiv").innerHTML);
      }
    </script>
  </body>
</html>
```

The button does not have the expected effect of changing the text in the browser

Why? What does it do?

file: issue2.html, issue2js.html

JQuery Again

- All property setting in base JS is done with functions by JQuery
 - Addresses exactly the issue from last slide, namely
 - Setting/examining an unknown property is OK
 - Using an unknown function is NOT

```
<html>
  <head>
    <script src="../../JQ/jquery-1.9.1.min.js"></script>
  </head>
  <body>
    <div id="mydiv">Original content</div>
    <button onclick="doReplace()">Replace</button>

    <script>
      function doReplace() {
        $("#mydiv").html("New Content");
        console.log($("#mydiv"));
        console.log($("#mydiv").html());
        console.log($("#mydiv").HTML());
      }
    </script></body></html>
```

JQuery ready function

```
<html>
  <head>
  </head>
  <body onload="replacer()">
    <span id="myspan" onclick="blob()"></span>
    <script>
      function replacer() {
        document.getElementById("myspan").innerHTML="my span";
      }
    </script>
  </body>
</html>
```

Converts a standard JS object
to a JQuery object

```
<html>
  <head>
    <script src=".../JQ/jquery-1.9.1.min.js"></script>
  </head>
  <body>
    <span id="myspan" onclick="blob()"></span>
    <script>
      $( document ).ready( function() {
        $("#myspan").html("my span");
      });
    </script>
  </body>
</html>
```


JQuery warnings

- avoid mixing with regular javascript objects
 - `var x = document.getElementById("id")`
`x.css("background-color":"red");`
 - Convert regular javascript object to JQuery using `$()` (but why?)
 - `var x = document.getElementById("id")`
`$(x).css("background-color":"red");`
- You can, but should avoid, searching entire DOM for objects. Better to get known enclosing item then search there
 - `$("ul").find("li").css("background-color", "red")`
 - Also, use variables to hold objects rather than re-search.
 - (Not important when docs are small)
- It may swallows links
 - `<a data-ajax="false" href="...">My link`
 - tries to give a better experience on page transitions my experience is that I often go no where
 - This is mostly jquery mobile

The Event Loop in JavaScript

- Javascript is single threaded
- Execution is line-by-line
 - So norm is that everything is synchronous and blocking

```
<html>
  <head>
    <script src="../../JQ/jquery-1.9.1.min.js"></script>
  </head>
  <body>
    <script>
      console.log("Message 1");
      setTimeout(function() {
        console.log("Message 2");
      }, 100);
      console.log("Message 3");
    </script>
  </body>
</html>
```

What is order of messages in log?

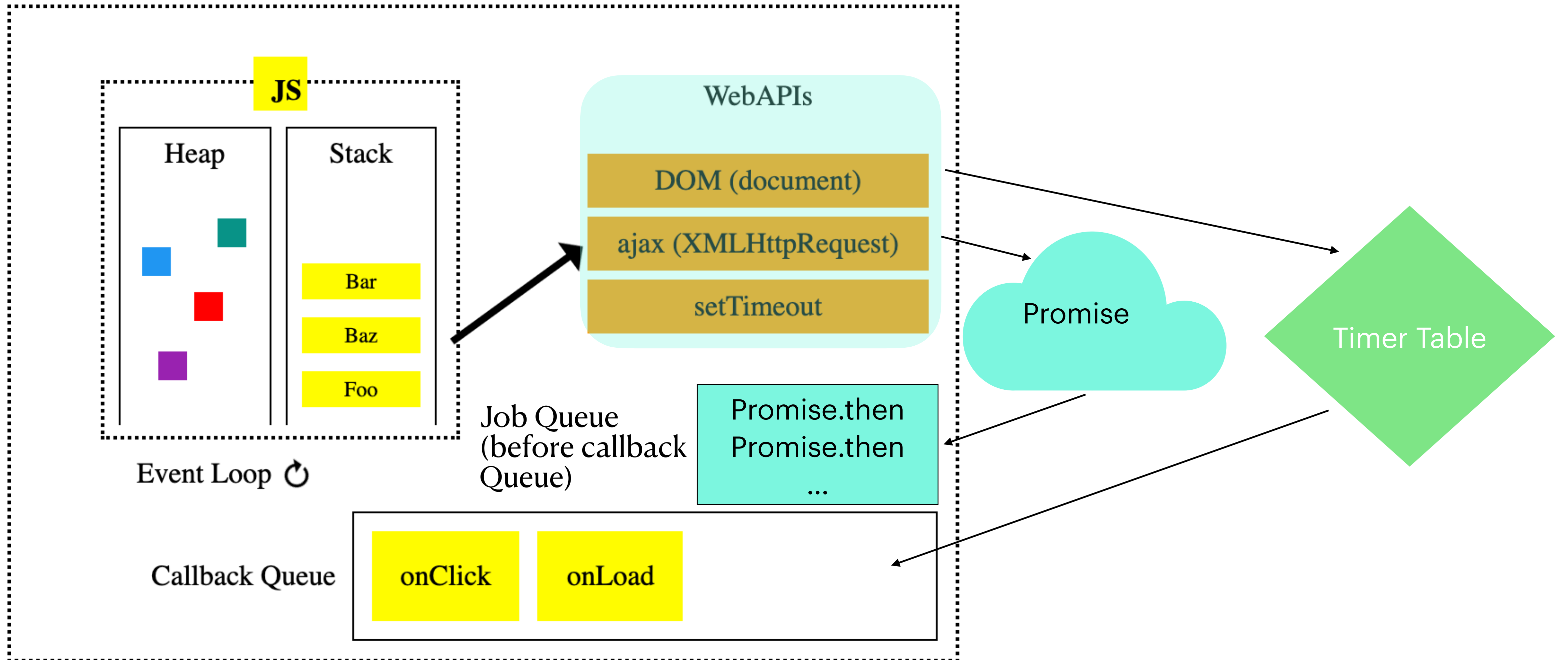
What about if timeout is 0?

Why?

file: eventloop1.html

Outside reading: <https://medium.com/@Rahulx1/understanding-event-loop-call-stack-event-job-queue-in-javascript-63dcd2c71ecd>

Event Loop



Async tasks (Job Queue and callback queue) only done when stack is empty!

What happens on screen, and When?

```
<html>
  <head>
    <script src="../../JQ/jquery-1.9.1.min.js"></script>
  </head>
  <body>
    <div id="countout"></div>
    <button onclick="push()" id="mybutton">Push Me</button>
    <script>
      var pushCount=0;
      $(document).ready(function() {
        $("#countout").html("Count " + pushCount);
      });
      function push() {
        gtsleep(2000);
        pushCount++;
        $("#countout").html("Count " + pushCount);
      }
      function gtsleep(milliseconds) {
        var start = new Date().getTime();
        for (var i = 0; i < 1e9; i++) {
          if (i%100000==0) {
            console.log(i);
            $("#countout").html("sleeping " + i);
          }
          if ((new Date().getTime() - start) > milliseconds){
            break;
          }
        }
      }
    </script></body></html>
```

What if I add a timeout?

NEVER DO THIS

file: eventloop2[a].html

Callbacks and Asynchrony

- Callbacks are functions
 - typically to be executed asynch in future

```
setTimeout(function() {  
    console.log("TimeoutFinished 2");  
}, 1000);
```

- May be synchronous (still a callback, just one that goes directly onto call stack)

```
const gods = ['Apollo', 'Artemis', 'Ares', 'Zeus'];  
gods.forEach(function (eachName, index){  
    console.log(index + '. ' + eachName);  
});
```

Promises

- An implementation of async function with callbacks

```
fetch('products.json').then(function(response) {  
  return response.json();  
}).then(function(json) {  
  products = json;  
  initialize();  
}).catch(function(err) {  
  console.log('Fetch problem: ' + err.message);  
});
```

- May have multiple “then” functions
 - each then has response function
 - optional “reject” function
- Thens are chained, second then only after first. Can assume results of previous items in chain
- Optional “Catch”

Promises

```
<html>
<head><script src="../../JQ/jquery-1.9.1.min.js"></script></head>
<body>
  <div id="countout"></div>
  <button onclick="push()" id="mybutton">Push Me</button>
  <script>
    var pushCount=0;
    $(document).ready(function() {
      $("#countout").html("Count " + pushCount);
    });
    function push() {
      gtsleep2(1000).then(function(val) {
        pushCount++;
        $("#countout").html(val + " " + pushCount);
      },
        function(reason) {
          $("#countout").html("Rejected " + reason + " " + pushCount);
        });
    }
    function gtsleep2(ms)
    {
      return(new Promise(function(resolve, reject) {
        setTimeout(function() { resolve("success"); }, ms);
      }));
    }
  </script></body></html>
```

Executed on "resolve"

Executed on "reject"

Build and return a new
promise object

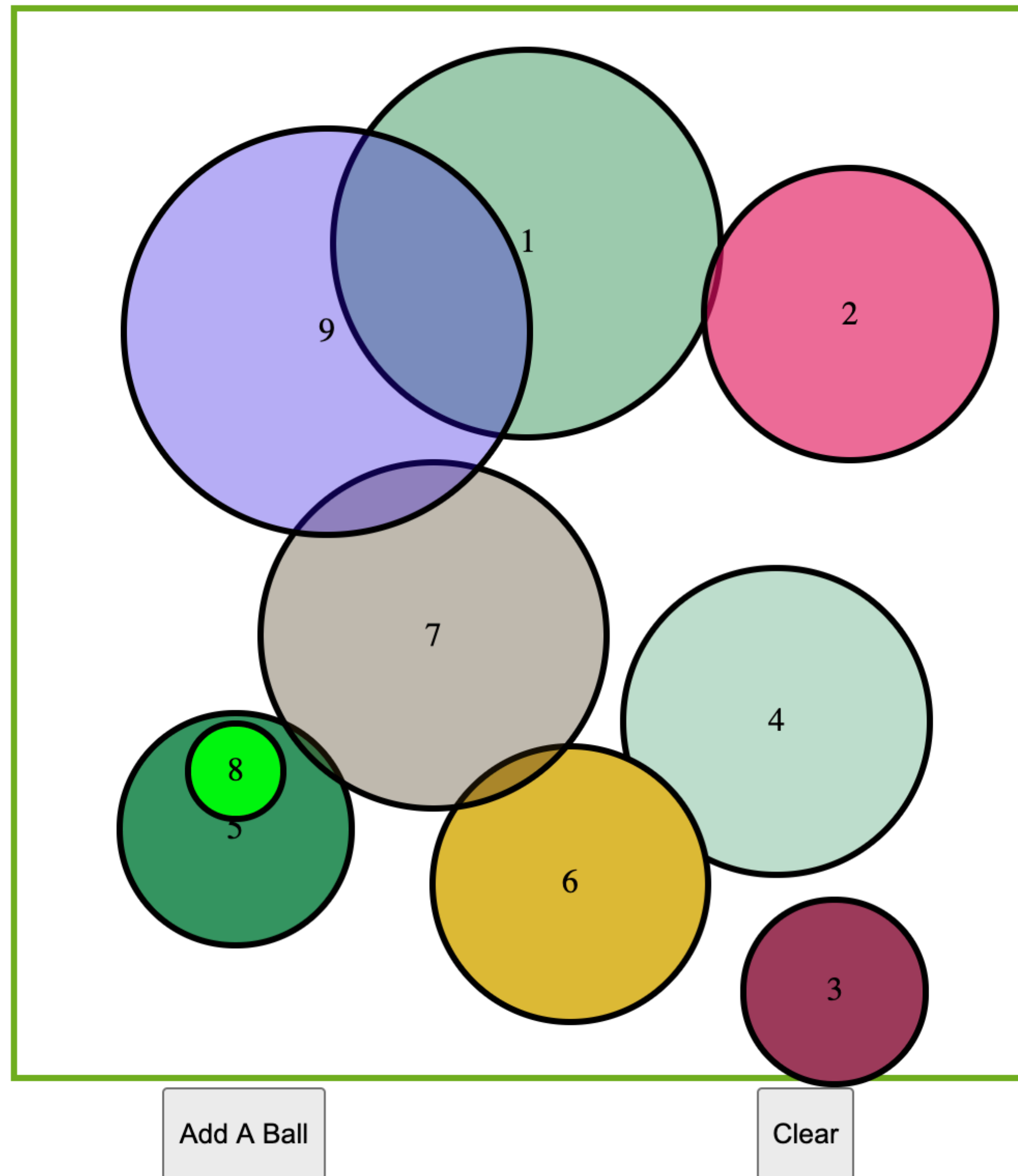
reject or resolve
here

file:eventloop3.html

setTimeout and setInterval

- Do Once in future:
 - `handle = setTimeout(callback, time)`
 - puts callback func into timerTable
 - `clearTimeout(handle)`
 - removes from timerTable
- Do repeatedly in future:
 - `handle = setInterval(callback, time)`
 - `clearInterval(handle);`
- Write `setInterval` using `setTimeout()` — including the ability to stop!

Screen Drawing



b2.html

“Drawing” with divs

idea: make a div and put it where you want it

```
<html>
  <head>
    <script src="../JQ/jquery-1.9.1.min.js"></script>
    <style>
      .maind {
        width: 90%;
        height: 90%;
        margin-left: 50px;
        margin: auto;
        border: 3px solid #73AD21;
      }
      .balld {
        position: absolute;
        text-align: center;
        border: 3px solid black;
      }
      .bttn {
        height: 45px;
      }
    </style>
  </head>
```

DwD, part 2

The Layout

```
<body>
  <div id="mdiv" class="maind"></div>
  <table width="100%">
    <tr><td width="50%"><center><button class="btn" onclick="makeBall()">Add A Ball</button></center></td>
      <td><center><button class="btn" onclick="clearBalls()">Clear</button></center></td>
    </tr>
  </table>
```

DwD, part 3

Javascript (using JQuery)

```
<script>
//The number of balls created.
var counter = 0;

/**
 * Create a random color. Actually this returns a string
 * which can be evaluated into a random color
 */
function randomColor() {
    return (
        "rgba(" +
        Math.round(Math.random() * 250) + "," +
        Math.round(Math.random() * 250) + "," +
        Math.round(Math.random() * 250) + "," +
        Math.ceil(Math.random() * 10) / 3 + ")"
    );
}

/**
 * Clear all of the balls
 */
function clearBalls() {
    $(".ball").remove();
}

```

```
function makeBall() {
    counter=counter+1;
    canvas = $("#mdiv"); // get the place where the ball will
    tx = canvas.width();
    ty = canvas.height();
    radius = 14 + Math.random() * (0.4*(tx<ty ? tx : ty)); //
    x = Math.random() * (tx-radius); // ball location
    y = Math.random() * (ty-radius);
    // make the ball
    jelem = $('<div>'+counter+"</div>"); //document.createElement
    jelem.addClass("ball");
    jelem.css( {
        'line-height':radius+'px',
        'margin-left': x+'px',
        'margin-top':y+'px',
        "height": radius+'px',
        'width':radius+'px',
        'border-radius':radius+'px',
        'background-color': randomColor() });
    jelem.hover(function(){ //mouseover and mouseout
        $(this).css('border', '3px solid yellow');
    }, function(){
        $(this).css('border', '3px solid black');
    });
    //put the ball into the target div
    canvas.append(jelem);
}

```

Drawing with Canvas

- canvas is an html element that you can literally draw on.
- Just doing circles so everything here could be done with divs
- Diagonal lines, etc not so much
- canvas and jquery do not talk well so using base javascript for canvas

```
<head>
  <style>
    .maind {
      border: 3px solid #73AD21;
    }
  </style>
</head>
<body>
  <script src="../../JQ/jquery-1.9.1.min.js"></script>
  <canvas id="canvas" class="maind"></canvas>
  <button onclick="startBall()">New Ball</button>
```

DwC, Javascript pt 1

```
var ballcount=0;
var balls = [];
var animating=0;
$(document).ready(
    function() {
        console.log("A");
        console.log("b");
        ww = Math.floor(0.9*window.innerWidth);
        wh= Math.floor(0.9*window.innerHeight);
        canvas = document.getElementById("canvas");
        canvas.width=ww;;
        canvas.height=wh;
    }
);
function randomColor() {
    return (
        "rgba(" +
        Math.round(Math.random() * 250) + "," +
        Math.round(Math.random() * 250) + "," +
        Math.round(Math.random() * 250) + "," +
        Math.ceil(Math.random() * 10) / 3 + ")"
    );
}
```

DwC, javascript part 2

```
function makeBall() {
    ballcount = ballcount + 1;
    canvas = document.getElementById("canvas");
    tx = canvas.clientWidth;
    ty = canvas.clientHeight;
    ball = new Object();
    ball.radius = Math.random() *(tx*0.1) + 14;
    ball.x = Math.random() * (tx - 2*ball.radius) + ball.radius;
    ball.y = Math.random() * (ty - 2*ball.radius) + ball.radius;
    ball.color=randomColor();
    ball.speed = Math.random()*ball.radius*0.33+1;
    ball.counter = ballcount;
    return ball;
    //drawBall(ball);
}
function startBall() {
    balls.push(makeBall());
    if (animating==0) {
        console.log("Interval start");
        animating=1;
        window.requestAnimationFrame(drawBalls);
    }
}
```

DwC, Javascript part 3

```
function drawBall(ctx, ball) {
  ctx.beginPath();
  ctx.arc(ball.x, ball.y, ball.radius, 0, 2 * Math.PI);
  console.log(ball.counter + " " + ball.x + " " + ball.y + " " + ball.radius);
  ctx.fillStyle = ball.color;
  ctx.fill();
  ctx.stroke();
  ball.x=ball.x+ball.speed;
}

function drawBalls() {
  canvas = document.getElementById("canvas");
  var tx = canvas.clientWidth;
  var ty = canvas.clientHeight;
  var ctx = canvas.getContext("2d");
  ctx.clearRect(0, 0, tx, ty);
  for (i=balls.length-1; i>=0; i--) {
    drawBall(ctx, balls[i]);
    if ((balls[i].x-balls[i].radius) > tx) {
      balls.splice(i,1);
    }
  }
  if (balls.length>0)
    window.requestAnimationFrame(drawBalls);
  else
    animating=0;
}
```


Clicks in a Canvas

```
$(document).ready(  
  function() {  
    ww = Math.floor(0.9*window.innerWidth);  
    wh = Math.floor(0.9*window.innerHeight);  
    canvas = document.getElementById("canvas");  
    canvas.width=ww;;  
    canvas.height=wh;  
  
    canvas.addEventListener("mousedown", function(e)  
    {  
      getMousePosition(canvas, e);  
    });  
  }  
);  
function getMousePosition(canvas, event) {  
  let rect = canvas.getBoundingClientRect();  
  let x = event.clientX - rect.left;  
  let y = event.clientY - rect.top;  
  console.log("Coordinate x: " + x, "Coordinate y: " + y);  
}
```

Scope in Javascript, when do variables exist?

DwD, adding animation

Adjustments to CSS and html layout

```
.btn {
  height:45px;
  width:50%
}
.telem {
  width:33%;
  text-align: center;
}
```

```
<body>
  <div id="mdiv" class="maind"></div>
  <table width="100%">
    <tr><td class="telem"><button class="btn"
onclick="addBall()">Add A Ball</button></td>
      <td class="telem"><button class="btn"
onclick="moveOneStep()">Move</button></td>
      <td class="telem"><button class="btn"
onclick="clearBalls()">Clear</button></td>
    </tr>
  </table>
```

file:b2anim.html, b2animu.html

DwD, adding animation

Javascript changes

```
// unchanged above here
cnvas.append(jelem);
// new .. create and fill an object for each ball
ball = new Object();
ball.element = jelem;
ball.addx = Math.random()*20*(Math.random()>0.5?1:-1);
ball.addy = Math.random()*20*(Math.random()>0.5?1:-1);
ball.radius = radius;
ball.xloc=x;
ball.yloc=y;
return ball;
}
```

```
var balls = [];
function addBall() {
    balls.push(makeBall());
}
```

```
function moveOneStep() {
    for (ball of balls) {
        let cnvas = $("#mdiv"); // get the place where th
        let tx = cnvas.width();
        let ty = cnvas.height();
        ball.xloc += ball.addx;
        ball.yloc += ball.addy;
        if (ball.xloc >= (tx-ball.radius)) {
            ball.xloc=tx-ball.radius;
            ball.addx=-ball.addx;
        }
        if (0 > ball.xloc) {
            ball.xloc=0;
            ball.addx = -ball.addx;
        }
        if (ball.yloc >= (ty-ball.radius)) {
            ball.yloc = ty-ball.radius;
            ball.addy = -ball.addy;
        }
        if (0 > ball.yloc) {
            ball.yloc = 0;
            ball.addy = -ball.addy;
        }
        ball.element.css({'margin-left': ball.xloc+'px',
            'margin-top':ball.yloc+'px',});
    }
}
```