Let’s Build A Runloop

Introduction

http://bit.ly/runloop-bcn

★☆Magic★☆

```
focusOut: function(event) {
  // wait for activeElement to get set (I think?)
  Ember.run.next(this, function(){
    // guard against case where this.$() is undefined.
    // otherwise get random failures
    if (this.$()) {
      if (!this.$().has(document.activeElement).length) {
        this.close();
      }
    }
  });
}
```

★☆Magic★☆

```
templateNameDidChange: (->
  # WTFZOMG
  Ember.run.next => Ember.run.next => @rerender()
).observes("templateName")
```

★☆Magic★☆

```
contentDidChange: ->
  # 2 ticks until DOM update
  Em.run.next(this, (->
    Em.run.next(this, (->
      @$().trigger("liszt:updated")))
    )
  )
```

Why is the runloop a difficult concept to master?

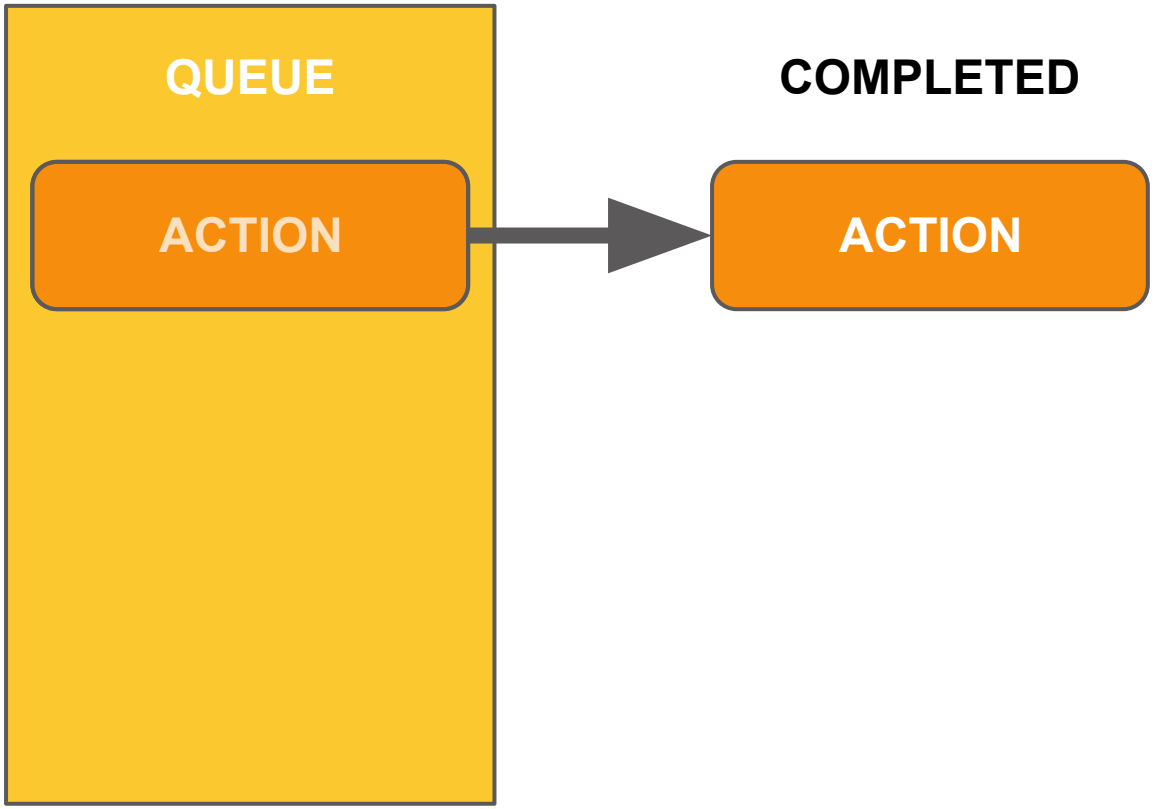
Why is the runloop a difficult concept to master? (reason #1)

“runloop”

Why is the runloop a difficult concept to master? (reason #1)

deferred action queue

```
Ember.run(function(){
  Ember.run.schedule('actions',
someFunction);
});
```



Why is the runloop a difficult concept to master? (reason #2)

We don’t know what problems the runloop is solving.

Why is the runloop a difficult concept to master? (reason #2)

```
focusOut: function(event) {
  // wait for activeElement to get set (I think?)
  Ember.run.next(this, function(){
    // guard against case where this.$() is undefined.
    // otherwise get random failures
    if (this.$()) {
      if (!this.$().has(document.activeElement).length) {
        this.close();
      }
    }
  });
}
```

Why is the runloop a difficult concept to master? (reason #2)

Mostly, Ember documentation focuses on usage.

Why is the runloop a difficult concept to master? (reason #2)

Motivation → Abstraction → Usage

Why is the runloop a difficult concept to master? (reason #2)

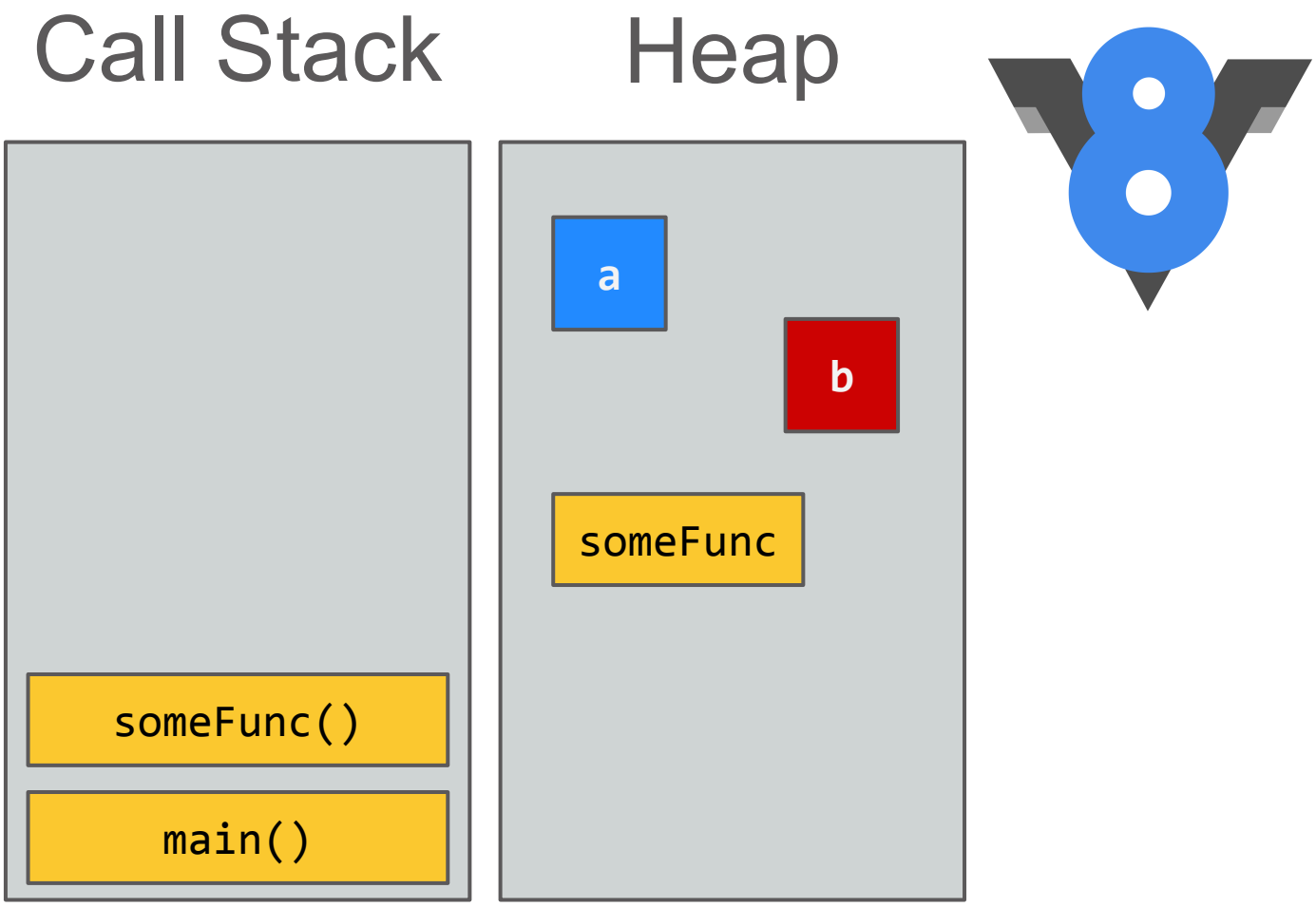
Understanding

Motivation → Abstraction → Usage

The JavaScript Event Loop

Let’s build a runloop, pt 1

The JavaScript Event Loop



The JavaScript Event Loop

[Animation of the call stack]

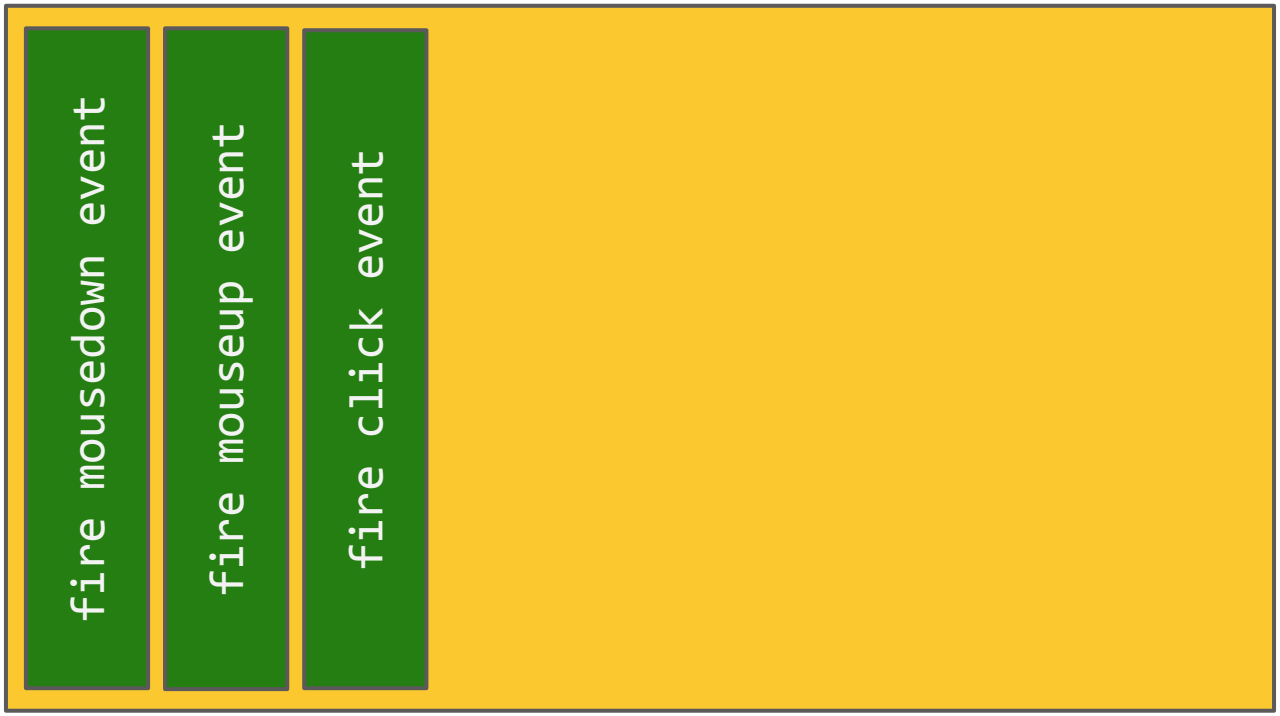
The JavaScript Event Loop

```
setTimeout(function runThis(){
 alert('Whammo Blammo!');
},0);
```

Schedules to *somewhere*...

The JavaScript Event Loop

Event Queue

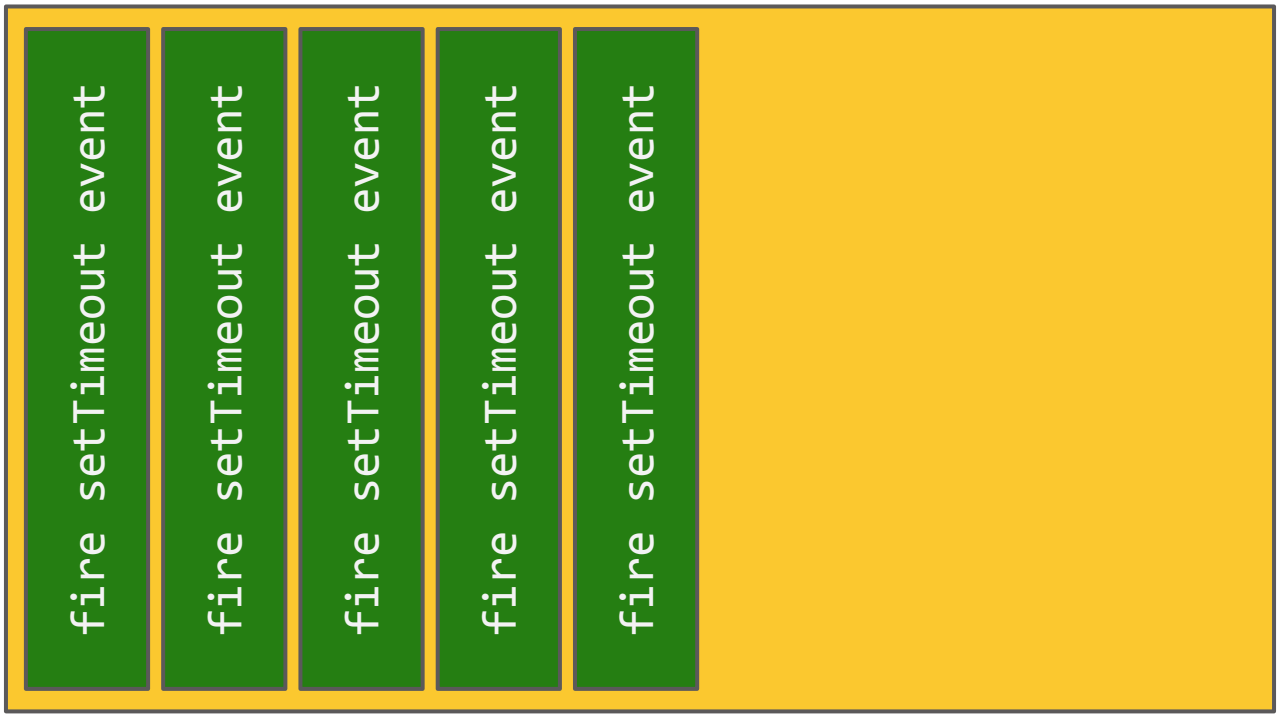


The JavaScript Event Loop

```
setTimeout(function runThis(){
 alert('Whammo Blammo!');
},0);
```

The JavaScript Event Loop

```
var times = 5; while(times--) {
 setTimeout(function(){ alert('whammo') }, 0);
}
```



The JavaScript Event Loop

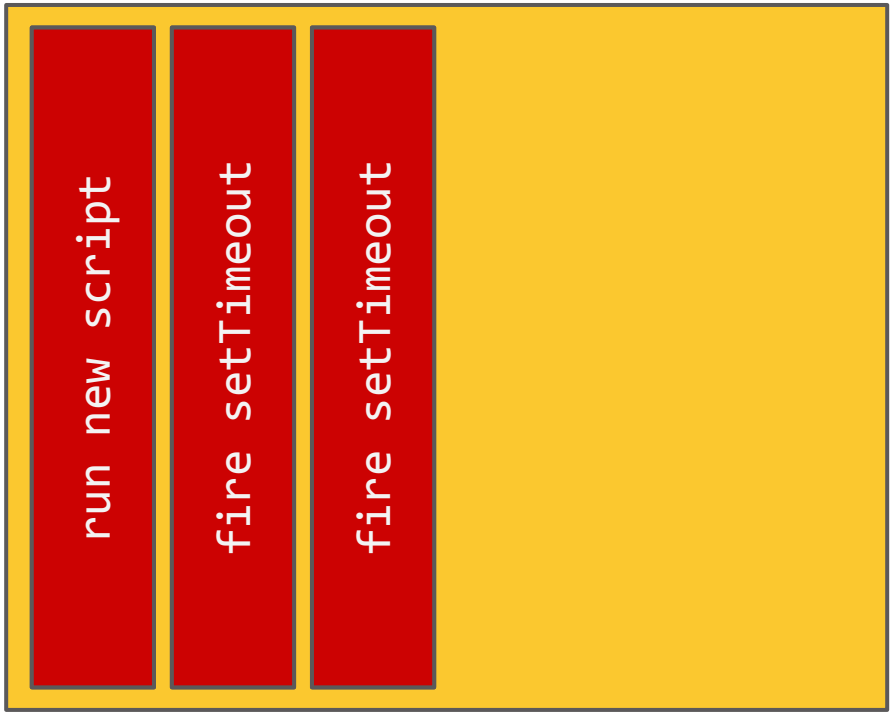
```
alert('whammo');
```

The JavaScript Event Loop

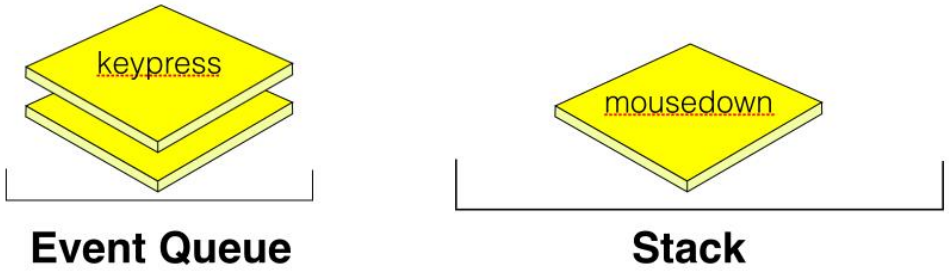
```
setTimeout(function(){
  alert('whammo');
  setTimeout(function(){
    alert('kablamo');
  }, 0);
}, 0);
```

The JavaScript Event Loop

```
setTimeout(function(){

alert('whammo'););

setTimeout(function(){){
alert('kablamo');
  alert('kablamo'););

}, 0););

}, 0);
```



The JavaScript Event Loop



Animation of the event loop

The JavaScript Event Loop

Intro Exercise

Use Chrome devtools to see the event queue and stack

Perilous Time(ing)s

Let’s build a runloop, pt 2

Perilous Time(ing)s

How does a browser render a document?

Perilous Time(ing)s

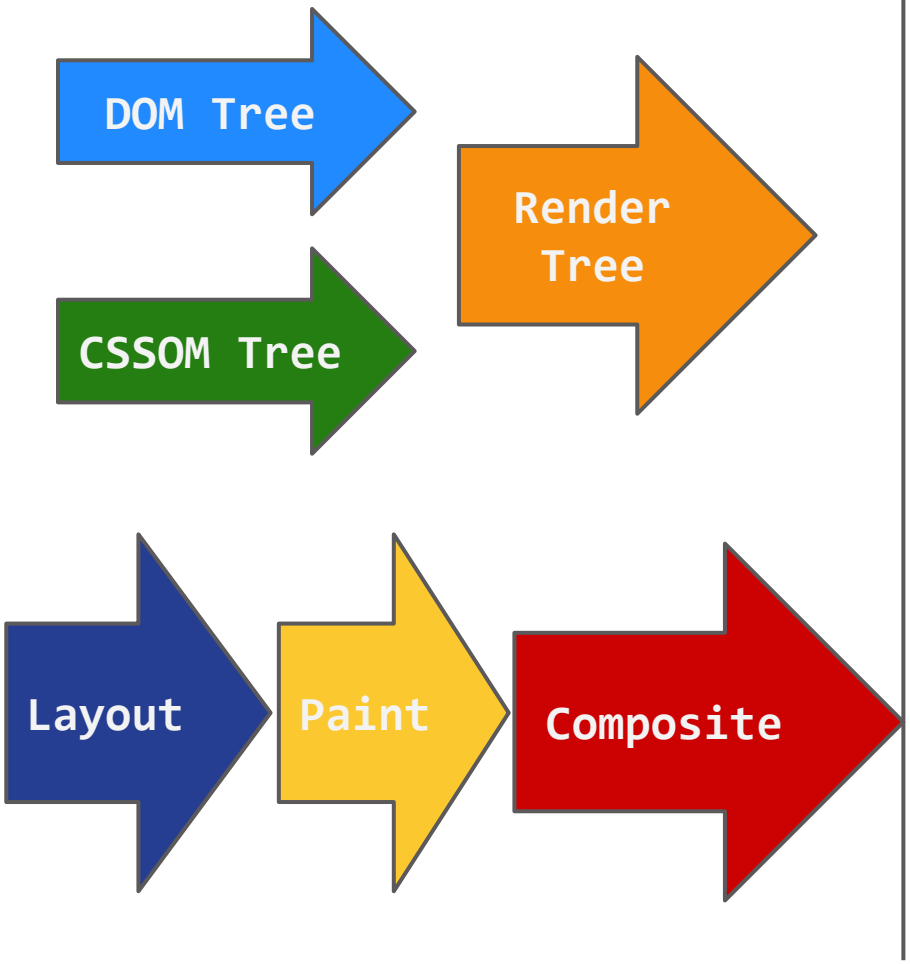
```
<html>
 <head>
 <style>
   img {
     border-radius: 3px;
     color: black;
   }
 </style>
 </head>
 <body>
   <img src="/hamster.jpg" width="482" height="362">
 </body>
</html>
```

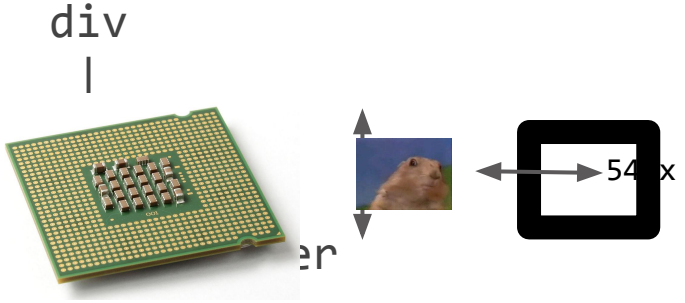
Perilous Time(ing)s

Perilous Time(ing)s

Render pipeline

Perilous Time(ing)s

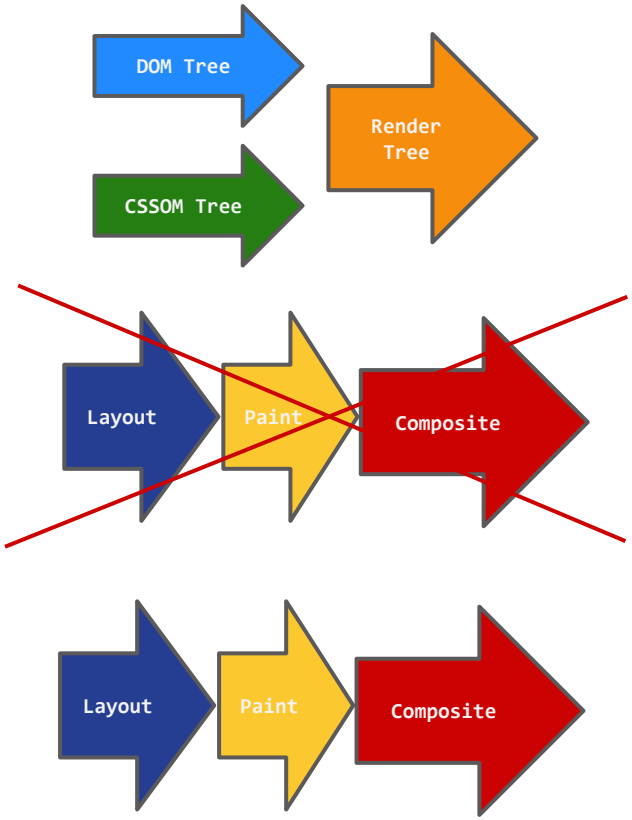




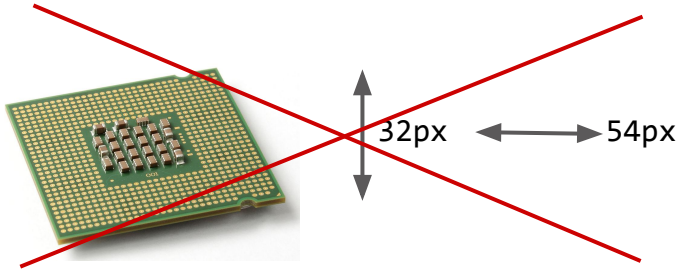
Perilous Time(ing)s

Layout Invalidation

Perilous Time(ing)s



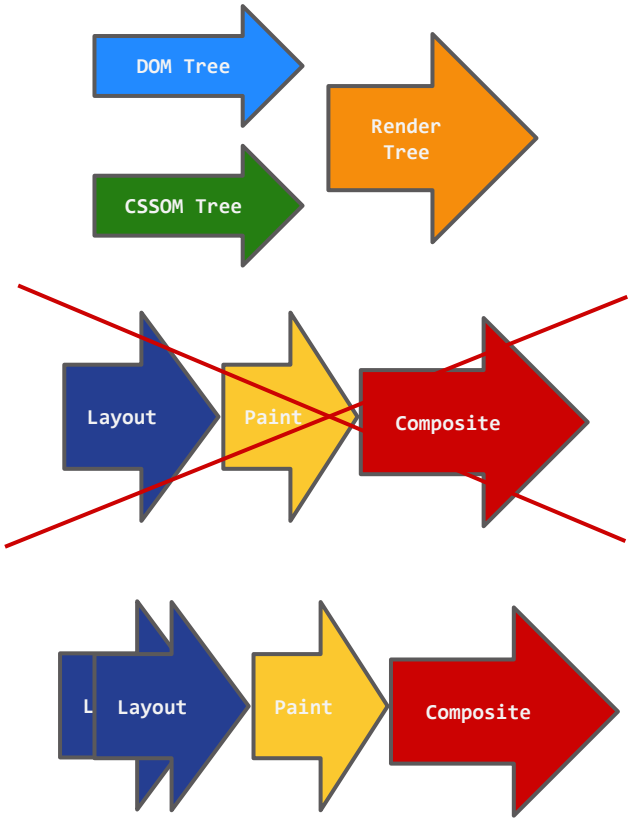

```
HTML
 |
 |-- DIV width: 10px
 |     |
 |     |-- P width: 15px
 |     |
 |     |-- P width: 25px
 |
 |-- SPAN
```



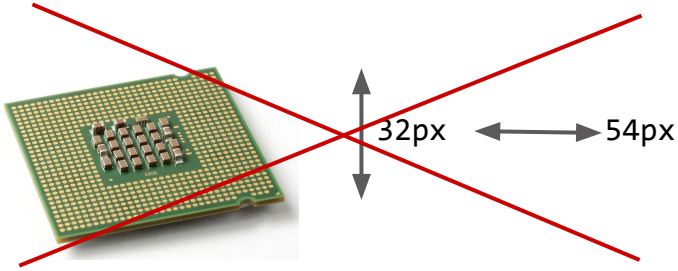
Perilous Time(ing)s

Forced Layout

Perilous Time(ing)s




```
img.className = widerClass;
var height = img.offsetHeight;
img                          ss;
```



Perilous Time(ing)s

Exercise A

Examine Forced Layout

A First Runloop

Let’s build a runloop, pt 3

A First Runloop

Exercise B

Build a runloop that defers rendering done during the loop over images

Pipelining Tasks

Let’s build a runloop, pt 4

A First Runloop

Exercise C

Add an actions queue that flushes before the render queue

Backburner.js

Let’s build a runloop, pt 5

Backburner.js

- github.com/ebryn/backburner.js
- Stand-alone runloop library

Backburner.js

Configurable queues

&

Schedule a task once

Backburner.js

```
var backburner = new Backburner(['render']),
    person = {name: "Erik"};

function updateName() {
  $('#name').text(person.name);
}

function setName(name) {
  person.name = name;
  backburner.deferOnce('render', updateName);
}

backburner.run(function() {
  setName("Kris");
  setName("Tom");
  setName("Yehuda");
});
```

Backburner.js

source

```
pushUniqueWithoutGuid: function(target, method, args, stack) {
  var queue = this._queue;

  for (var i = 0, l = queue.length; i < l; i += 4) {
    var currentTarget = queue[i];
    var currentMethod = queue[i+1];

    if (currentTarget === target && currentMethod === method) {
      queue[i+2] = args;  // replace args
      queue[i+3] = stack; // replace stack
      return;
    }
  }

  queue.push(target, method, args, stack);
},
```

Backburner.js

Auto-runs

```
function createAutorun(backburner) {
 backburner.begin();
 backburner._autorun = global.setTimeout(function() {
   backburner._autorun = null;
   backburner.end();
 });
}
```

Backburner.js

Auto-runs (demo on Cory's laptop)

Backburner.js

Moving between queues

The Ember.js Runloop

Let’s build a runloop, pt 6

The Ember.js Runloop

Multiple queues

sync, actions, routerTransition, render, afterRender, destroy

The Ember.js Runloop

Multiple queues

sync, actions, routerTransition, render, afterRender, destroy

The Ember.js Runloop

```
var MyView = Ember.View.extend({
  renderMasonry: function(){
    var element = this.$();
    Ember.run.scheduleOnce('afterRender', function(){
      $.masonry(element);
    });
  }.on('didInsertElement')
});
```

Backburner.js

Join a running runloop

Ember wraps events

In Ember's Event Dispatcher, click events are put into the run loop

* How does Ember wrap BB?
* Queues in Ember, what are they for?
* How to use runloop in tests
* Conserving array change with Ember runloop, Q/A group style (leave queues on slides for reference)
* Conserving DOM changes with Ember runloop, Q/A group style (leave queues on slides for reference)