Images from anywhere

A brief overview of Mootools

The problem

- We need shoes to stat
- Shoes are all over the web
- People don’t know what an image url is

The solution

- JavaScript (with the help of MooTools)
- A tiny bit of Ruby (with the help of Merb)
- http://shoestat.com/units/new
- http://github.com/vanpelt/instimage

The namespace

```
1 var Instimage = {
2   pending: [],
3   canceled: false,
4   search: "",
5   loading: new Element('img', {src:'/loading.gif', id:'loading'}),
6   preload: function() { ... },
7   //etc...
8 }
```

Preload the images

```
1 preload: function(img){
2    var image = new Image()
3    image.onload = this.goodImg
4    image.onerror = this.goodImg
5    this.cancel.delay(10000, this)
6    image.src = img
7 }
```

A little scope

```
goodImg: function() {
  if(this.width > 200 && this.height > 150 && !Instimage.canceled)
    Instimage.resize(this).addClass('virgin').inject('images')
  }
  Instimage.pending.pop()
  Instimage.inform()
}
```

The storage API

```
1 resize: function(img){
2   img = $(img).store('meta', {width: img.width, height: img.height})
3   if(img.width > img.height) {
4     var width = 200
5     var height = (200 / img.width) * img.height
6   } else {
7     var height = 200
8     var width = (200 / img.height) * img.width
9   }
10  return img.setStyles({width: width, height: height})
11 }
```

Some more scope

```
1 inform: function() {
2   $$('#images img.virgin').each(function(i){
3     i.removeClass('virgin')
4     this.img2JSON(i)
5     i.addEvent('click', function(){
6       Instimage.cancel()
7       $('src').set('value', this.src)
8       Instimage.explode(this)
9       new Fx.Scroll(window).toElement('form')
10    })
11  }, Instimage)
12  if(this.pending.length == 0) {
13    //Logic to inform the user how we did
14  }
15 }
```

More storage and XSJSON

```
1 img2JSON: function(img){
2   img.addEvent('mouseenter', function(){
3     var meta = this.retrieve('meta')
4     if(!(meta && meta.exif)) {
5       new Element('script', {type:'text/javascript', src:'http://
img2json.appspot.com/go/?
callback=Instimage.meta&url='+encodeURIComponent(this.src)}).inject('body')
6       $('stats').grab(Instimage.loading, 'top')
7     } else
8       Instimage.meta(meta)
9   })
10 }
```

FX, chaining & pseudo

```
explode: function(img) {
  if(img.hasClass('exploded')) {
    new Fx.Tween(img, {
      property: 'opacity',
      onComplete: function(){
        Instimage.resize(img).removeClass('exploded').setStyle('position'
exploded').fade('show')
      }
    }).start(0)
  } else {
    var pos = img.getPosition('images')
    var size = img.retrieve('meta')
    img.addClass('exploded')
    img.set('morph', {duration: 'long', transition: 'bounce:out'});
    img.setStyles.delay(500, img, {top:pos.y, left:pos.x})
    img.morph.delay(500, img, { top:40, left:0, position: 'absolute',
idth:size.width, height:size.height})
  }
  $$('#images img:not(.exploded)').fade()
}
```

Events and logic

```
1 if(Instimage.search != $('search').value) {
2   Instimage.search = $('search').value
3   src = 'http://images.google.com/images?q='+encodeURIComponent(Instimage.search)
4 }
5 if(src.test(/\.(jpg|jpeg|gif|png)$/)) {
6   Instimage.preload(src)
7 } else {
8   new Request.JSON({
9     onSuccess: function(json) {
10      Instimage.pending = json
11      if(json.length == 0)
12        Instimage.inform()
13      json.each(Instimage.preload, Instimage)
14    },
15    onFailure: function() {
16      $('loading').dispose()
17    },
18    data: { src: src }
19  }).get('/instimage/scrape')
20 }
```

A little bit of ruby

```
 1 def scrape
 2   provides :json
 3   root = params[:src][/^(https?:\/\/.+?\/).+/, 1]
 4   if root =~ /images\.google\.com/
 5     images = open(params[:src]).read.scan(/dyn\.Img\(".*?",".*?",".*?","(.
+?)"/).flatten
 6   else
 7     begin
 8       parsed = Hpricot(open(params[:src]))
 9       images = (parsed/"img").map do |i|
10         rel(i['src'])
11       end.flatten.uniq
12       images += (parsed/"a").map do |a|
13         src = a['href']
14         next unless src =~ /\.(jpg|jpeg|gif|png)$/
15         rel(src)
16       end.flatten.uniq
17     rescue
18       images = []
19     end
20   end
21   display images
22 end
```

Conclusion

- A little bit of code goes a long way
- Images + Javascript == Oreos + Milk
- Shoestat is the bomb.com (thanks to Mootools)
- Play with it (http://github.com/vanpelt/instimage)