

Und noch ein JavaScript Fader



[movingboxes.htm \(/code/movingboxes.htm?mode=download\)](#) HTML (5,13 kByte) 12.03.2016 17:14

```
<!DOCTYPE html >
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Fader</title>
    <style type="text/css">
      body {margin:0; padding:0;}
#wrapper {position:absolute; width:100%; height:160px; margin:0; padding:0; overflow:hidden;}
.fader {position:absolute; height:100%; margin:0; padding:0;}
.fader-item{position:relative; display:inline-block; padding:0; top:0; height:100%; width:240px;
overflow:hidden;}
.fader-item img{position:relative;}
    </style>
    <script type="text/javascript">
      /*  */
window.requestAnimFrame = (function(){
  return window.requestAnimationFrame ||
         window.webkitRequestAnimationFrame ||
         window.mozRequestAnimationFrame ||
         function(callback){
           window.setTimeout(callback, 1000 / 60);
         };
})();

function animate(elem, attr, dest, callback) {
  var from, to = dest, el = elem, fc = callback;
  if (el.style.getPropertyValue) {
    from = parseInt(el.style.getPropertyValue(attr));
  } else {
    from = parseInt(el.style.getAttribute(attr));
  };
  if (from == to) return;
  var start = Date.now();
  var tick = function() {
    var t = Math.min(1, ((Date.now() - start) / 1000));
    var eased = t&lt;.5 ? 4*t*t*t : (t-1)*(2*t-2)*(2*t-2)+1;
    var val = (eased * (to - from)) + from;
    if (el.style.setProperty) {
      el.style.setProperty(attr, val + "px");
    } else {
      el.style.setAttribute(attr, val + "px");
    };
    if (t&lt;1) {
      window.requestAnimFrame(tick);
    } else {
      if (fc) fc();
    }
  };
  tick();
}

function getElementsByClassName(searchClass, tagName, domNode) {
  if (domNode == null) domNode = document;
  if (tagName == null) tagName = '*';
  var el = new Array();
  var tags = domNode.getElementsByTagName(tagName);
  var tcl = " "+searchClass+" ";
  for(var i=0,j=0; i&lt;tags.length; i++) {
    var test = " " + tags[i].className + " ";
    if (test.indexOf(tcl) != -1) {
      el[j++] =tags[i];
    }
  }
}</pre></div>
```

```

    }
};
return el;
}

function getChildNodes(parentElement, tagName) {
    var childs = parentElement.childNodes;
    // get divs
    var children = [];
    for (var i=0; i<childs.length; i++) {
        if (((childs[i].nodeType == 1) ? childs[i].tagName.toLowerCase() : '') == tagName) {
            children.push(childs[i]);
        }
    };
    // delete whitespaces
    for (i=childs.length-1; i>0; i--) {
        if (childs[i].nodeType != 1) {
            parentElement.removeChild(childs[i]);
        }
    }
    return children;
}

function jsFader(){
    this.obj = null;
    this.items = [];
    this.currentFade = 0;
}
jsFader.prototype = {
    fit: function(el) {
        var it = el || this.items[this.currentFade];
        var img = getChildNodes(it, 'img')[0];
        if ((img.width / img.height) >= (it.offsetWidth / it.offsetHeight)) {
            img.style.width = 'auto';
            img.style.height = '100%';
            img.style.top = '0';
            img.style.left = -Math.floor((img.width - it.offsetWidth) / 2) + 'px';
        } else {
            img.style.width = '100%';
            img.style.height = 'auto';
            img.style.left = '0';
            img.style.top = -Math.floor((img.height - it.offsetHeight) / 2) + 'px';
        }
    },
    fade: function() {
        var self = this;
        var width = this.items[this.currentFade].offsetWidth;
        var last = this.items[this.currentFade];
        var fader = this.obj;
        this.currentFade = this.items[this.currentFade+1] ? this.currentFade+1 : 0;
        animate(fader, 'left', -width, function() {fader.appendChild(last); fader.style.left = 0;});
        setTimeout(function() {self.fade();}, 3000);
    },
    init: function(fader) {
        var self = this;
        this.obj = fader;
        fader.style.left = 0;
        this.items = getChildNodes(fader, "div");
        var width = this.items[0].offsetWidth;
        for (var i=0; i<this.items.length; i++) {
            width += this.items[i].offsetWidth;
            this.fit(this.items[i]);
        };
        fader.style.width = width + 'px';
        setTimeout(function() {self.fade();}, 3000);
        if (window.addEventListener) {
            window.addEventListener('resize', function(event) {self.fit();});
        } else if (window.attachEvent) {
            window.attachEvent('onresize', function(event) {self.fit();});
        } else {
            window['onresize'] = function(event) {self.fit();};
        }
    }
}
function init(){

```

```

var faderList = getElementsByClassName("fader", "div");
for (var i=0; i<faderList.length; i++) {
    var fader = new jsFader();
    fader.init(faderList[i]);
};
}

function ready(f){/complete|loaded/i.test(document.readyState)?f():setTimeout("ready("+f+")",9)}
ready(init);
/* ]]> */
</script>
</head>
<body>
<div id="wrapper">
<div class="fader">
<div class="fader-item"></div>
<div class="fader-item"></div>
<div class="fader-item"></div>
<div class="fader-item"></div>
<div class="fader-item"></div>
<div class="fader-item"></div>
</div>
</div>
</body>
</html>

```

AUTOR: UDO SCHMAL, VERÖFFENTLICHT: 15.08.2014, LETZTE ÄNDERUNG: 03.02.2016

© Copyright 2021 Udo Schmal (/)