```
/**
 * reflection.js v1.9
 * http://cow.neondragon.net/stuff/reflection/
 * Freely distributable under MIT-style license.
 */

/* From prototype.js */
if (!document.myGetElementsByClassName) {
	document.myGetElementsByClassName = function(className) {
		var children = document.getElementsByTagName('*') || document.all;
		var elements = new Array();

		for (var i = 0; i < children.length; i++) {
			var child = children[i];
			var classNames = child.className.split(' ');
			for (var j = 0; j < classNames.length; j++) {
				if (classNames[j] == className) {
					elements.push(child);
					break;
				}
			}
		}
		return elements;
	}
}

var Reflection = {
	defaultHeight : 0.60,
	defaultOpacity: 0.80,

	add: function(image, options) {
		Reflection.remove(image);

		doptions = { "height" : Reflection.defaultHeight, "opacity" : Reflection.defaultOpacity }
		if (options) {
			for (var i in doptions) {
				if (!options[i]) {
					options[i] = doptions[i];
				}
			}
		} else {
			options = doptions;
		}

		try {
			var d = document.createElement('div');
			var p = image;

			var classes = p.className.split(' ');
```

```
                var newClasses = '';
                for (j=0;j<classes.length;j++) {
                        if (classes[j] != "reflect") {
                                if (newClasses) {
                                        newClasses += ' '
                                }

                                newClasses += classes[j];
                        }
                }

                var reflectionHeight = Math.floor(p.height*options['height']);
                var divHeight = Math.floor(p.height*(1+options['height']));

                var reflectionWidth = p.width;

                if (document.all && !window.opera) {
                        /* Fix hyperlinks */
        if(p.parentElement.tagName == 'A') {
                var d = document.createElement('a');
                d.href = p.parentElement.href;
        }

                                /* Copy original image's classes & styles to div */
                                d.className = newClasses;
                                p.className = 'reflected';

                                d.style.cssText = p.style.cssText;
                                p.style.cssText = 'vertical-align: middle';

                                var reflection = document.createElement('img');
                                reflection.src = p.src;
                                reflection.style.width = reflectionWidth+'px';
                                reflection.style.display = 'block';
                                reflection.style.height = p.height+"px";

                                reflection.style.marginBottom = "-"+(p.height-reflectionHeight)+'px';
                                reflection.style.filter = 'flipv
progid:DXImageTransform.Microsoft.Alpha(opacity='+(options['opacity']*100)+', style=1,
finishOpacity=0, startx=10, starty=0, finishx=0, finishy='+(options['height']*100)+')';

                                d.style.width = reflectionWidth+'px';
                                d.style.height = divHeight+'px';
                                p.parentNode.replaceChild(d, p);

                                d.appendChild(p);
                                d.appendChild(reflection);
                } else {
                        var canvas = document.createElement('canvas');
```

```
                        if (canvas.getContext) {
                                /* Copy original image's classes & styles to div */
                                d.className = newClasses;
                                p.className = 'reflected';

                                d.style.cssText = p.style.cssText;
                                p.style.cssText = 'vertical-align: middle; align: center;';

                                var context = canvas.getContext("2d");

                                canvas.style.height = reflectionHeight+'px';
                                canvas.style.width = reflectionWidth+'px';
                                canvas.height = reflectionHeight;
                                canvas.width = reflectionWidth;

                                d.style.width = reflectionWidth+'px';
                                d.style.height = divHeight+'px';
                                p.parentNode.replaceChild(d, p);

                                d.appendChild(p);
                                d.appendChild(canvas);

                                context.save();

                                context.translate(0,image.height-1);
                                context.scale(1,-1);

                                context.drawImage(image, 0, 0, reflectionWidth, image.height);

                                context.restore();

                                context.globalCompositeOperation = "destination-out";
                                var gradient = context.createLinearGradient(0, 0, 0,
reflectionHeight);

                                gradient.addColorStop(1, "rgba(255, 255, 255, 1.0)");
                                gradient.addColorStop(0, "rgba(255, 255, 255, "+(1-
options['opacity'])+")");

                                context.fillStyle = gradient;
                                if (navigator.appVersion.indexOf('WebKit') != -1) {
                                        context.fill();
                                } else {
                                        context.fillRect(0, 0, reflectionWidth, reflectionHeight*2);
                                }
                        }
                }
        } catch (e) {
        }
```

```
	},

	remove : function(image) {
		if (image.className == "reflected") {
			image.className = image.parentNode.className;
			image.parentNode.parentNode.replaceChild(image, image.parentNode);
		}
	}
}

function addReflections() {
	var rimages = document.myGetElementsByClassName('reflect');
	for (i=0;i<rimages.length;i++) {
		var rheight = null;
		var ropacity = null;

		var classes = rimages[i].className.split(' ');
		for (j=0;j<classes.length;j++) {
			if (classes[j].indexOf("rheight") == 0) {
				var rheight = classes[j].substring(7)/100;
			} else if (classes[j].indexOf("ropacity") == 0) {
				var ropacity = classes[j].substring(8)/100;
			}
		}

		Reflection.add(rimages[i], { height: rheight, opacity : ropacity});
	}
}

var previousOnload = window.onload;
window.onload = function () { if(previousOnload) previousOnload(); addReflections(); }
```