/\*

CUSTOM FORM ELEMENTS

Created by Ryan Fait

www.ryanfait.com

The only things you may need to change in this file are the following

variables: checkboxHeight, radioHeight and selectWidth (lines 24, 25, 26)

The numbers you set for checkboxHeight and radioHeight should be one quarter

of the total height of the image want to use for checkboxes and radio

buttons. Both images should contain the four stages of both inputs stacked

on top of each other in this order: unchecked, unchecked-clicked, checked,

checked-clicked.

You may need to adjust your images a bit if there is a slight vertical

movement during the different stages of the button activation.

The value of selectWidth should be the width of your select list image.

Visit http://ryanfait.com/ for more information.

\*/

var checkboxHeight = \”25\”;

var radioHeight = \”25\”;

var selectWidth = \”190\”;

/\* No need to change anything after this \*/

document.write('<style type=\”text/css\”>input.styled { display: none; } select.styled { position: relative; width: ' + selectWidth + 'px; opacity: 0; filter: alpha(opacity=0); z-index: 5; } .disabled { opacity: 0.5; filter: alpha(opacity=50); }</style>');

var Custom = {

init: function() {

var inputs = document.getElementsByTagName(\”input\”), span = Array(), textnode, option, active;

for(a = 0; a < inputs.length; a++) {

if((inputs[a].type == \”checkbox\” || inputs[a].type == \”radio\”) && inputs[a].className == \”styled\”) {

span[a] = document.createElement(\”span\”);

span[a].className = inputs[a].type;

if(inputs[a].checked == true) {

if(inputs[a].type == \”checkbox\”) {

position = \”0 -\” + (checkboxHeight\*2) + \”px\”;

span[a].style.backgroundPosition = position;

} else {

position = \”0 -\” + (radioHeight\*2) + \”px\”;

span[a].style.backgroundPosition = position;

}

}

inputs[a].parentNode.insertBefore(span[a], inputs[a]);

inputs[a].onchange = Custom.clear;

if(!inputs[a].getAttribute(\”disabled\”)) {

span[a].onmousedown = Custom.pushed;

span[a].onmouseup = Custom.check;

} else {

span[a].className = span[a].className += \” disabled\”;

}

}

}

inputs = document.getElementsByTagName(\”select\”);

for(a = 0; a < inputs.length; a++) {

if(inputs[a].className == \”styled\”) {

option = inputs[a].getElementsByTagName(\”option\”);

active = option[0].childNodes[0].nodeValue;

textnode = document.createTextNode(active);

for(b = 0; b < option.length; b++) {

if(option[b].selected == true) {

textnode = document.createTextNode(option[b].childNodes[0].nodeValue);

}

}

span[a] = document.createElement(\”span\”);

span[a].className = \”select\”;

span[a].id = \”select\” + inputs[a].name;

span[a].appendChild(textnode);

inputs[a].parentNode.insertBefore(span[a], inputs[a]);

if(!inputs[a].getAttribute(\”disabled\”)) {

inputs[a].onchange = Custom.choose;

} else {

inputs[a].previousSibling.className = inputs[a].previousSibling.className += \” disabled\”;

}

}

}

document.onmouseup = Custom.clear;

},

pushed: function() {

element = this.nextSibling;

if(element.checked == true && element.type == \”checkbox\”) {

this.style.backgroundPosition = \”0 -\” + checkboxHeight\*3 + \”px\”;

} else if(element.checked == true && element.type == \”radio\”) {

this.style.backgroundPosition = \”0 -\” + radioHeight\*3 + \”px\”;

} else if(element.checked != true && element.type == \”checkbox\”) {

this.style.backgroundPosition = \”0 -\” + checkboxHeight + \”px\”;

} else {

this.style.backgroundPosition = \”0 -\” + radioHeight + \”px\”;

}

},

check: function() {

element = this.nextSibling;

if(element.checked == true && element.type == \”checkbox\”) {

this.style.backgroundPosition = \”0 0\”;

element.checked = false;

} else {

if(element.type == \”checkbox\”) {

this.style.backgroundPosition = \”0 -\” + checkboxHeight\*2 + \”px\”;

} else {

this.style.backgroundPosition = \”0 -\” + radioHeight\*2 + \”px\”;

group = this.nextSibling.name;

inputs = document.getElementsByTagName(\”input\”);

for(a = 0; a < inputs.length; a++) {

if(inputs[a].name == group && inputs[a] != this.nextSibling) {

inputs[a].previousSibling.style.backgroundPosition = \”0 0\”;

}

}

}

element.checked = true;

}

},

clear: function() {

inputs = document.getElementsByTagName(\”input\”);

for(var b = 0; b < inputs.length; b++) {

if(inputs[b].type == \”checkbox\” && inputs[b].checked == true && inputs[b].className == \”styled\”) {

inputs[b].previousSibling.style.backgroundPosition = \”0 -\” + checkboxHeight\*2 + \”px\”;

} else if(inputs[b].type == \”checkbox\” && inputs[b].className == \”styled\”) {

inputs[b].previousSibling.style.backgroundPosition = \”0 0\”;

} else if(inputs[b].type == \”radio\” && inputs[b].checked == true && inputs[b].className == \”styled\”) {

inputs[b].previousSibling.style.backgroundPosition = \”0 -\” + radioHeight\*2 + \”px\”;

} else if(inputs[b].type == \”radio\” && inputs[b].className == \”styled\”) {

inputs[b].previousSibling.style.backgroundPosition = \”0 0\”;

}

}

},

choose: function() {

option = this.getElementsByTagName(\”option\”);

for(d = 0; d < option.length; d++) {

if(option[d].selected == true) {

document.getElementById(\”select\” + this.name).childNodes[0].nodeValue = option[d].childNodes[0].nodeValue;

}

}

}

}

window.onload = Custom.init;