Javascript

jQuery.noConflict();

jQuery(document).ready(function(){

//menu

jQuery('.main\_menu li > ul').animate({'opacity': 0});

jQuery('.main\_menu li').hover(function () {

var currentItem = jQuery(this).find('ul:first');

currentItem.stop().css({display:'block'}).animate({'opacity': 1}, 300);

},function() {

var currentItem = jQuery(this).find('ul:first');

currentItem.stop().animate({'opacity': 0}, 300, function()

{

currentItem.css({display:'none'});

});

});

// improves menu for mobile devices

jQuery('.top\_menu ul:eq(0)').mobileMenu({

switchWidth: 1139, //width (in px to switch at)

topOptionText: jQuery('.top\_menu').data('selectname'), //first option text

indentString: '&nbsp;&nbsp;&nbsp;' //string for indenting nested items

});

// improves menu for mobile devices

jQuery('.main\_menu ul:eq(0)').mobileMenu({

switchWidth: 1139, //width (in px to switch at)

topOptionText: jQuery('.top\_menu').data('selectname'), //first option text

indentString: '&nbsp;&nbsp;&nbsp;' //string for indenting nested items

});

//menu

jQuery('.top\_menu li > ul').animate({'opacity': 0});

jQuery('.top\_menu li').hover(function () {

var currentItem = jQuery(this).find('ul:first');

currentItem.stop().css({display:'block'}).animate({'opacity': 1}, 300);

},function() {

var currentItem = jQuery(this).find('ul:first');

currentItem.stop().animate({'opacity': 0}, 300, function()

{

currentItem.css({display:'none'});

});

});

});

(function($){

//variable for storing the menu count when no ID is present

var menuCount = 0;

//plugin code

$.fn.mobileMenu = function(options){

//plugin's default options

var settings = {

switchWidth: 20,

topOptionText: 'Navigate to...',

indentString: '&nbsp;&nbsp;&nbsp;'

};

//function to check if selector matches a list

function isList($this){

return $this.is('ul, ol');

}

//function to decide if mobile or not

function isMobile(){

return ($(window).width() < settings.switchWidth);

}

//check if dropdown exists for the current element

function menuExists($this){

//if the list has an ID, use it to give the menu an ID

if($this.attr('id')){

return ($('#mobileMenu-'+$this.attr('id')).length > 0);

}

//otherwise, give the list and select elements a generated ID

else {

menuCount++;

$this.attr('id', 'mobile-menu-'+menuCount);

return ($('#mobileMenu-mobile-menu-'+menuCount).length > 0);

}

}

//change page on mobile menu selection

function goToPage($this){

if($this.val() !== null){document.location.href = $this.val()}

}

//show the mobile menu

function showMenu($this){

$this.css('display', 'none');

$('#mobileMenu-'+$this.attr('id')).show();

}

//hide the mobile menu

function hideMenu($this){

$this.css('display', '');

$('#mobileMenu-'+$this.attr('id')).hide();

}

//create the mobile menu

function createMenu($this){

if(isList($this)){

//generate select element as a string to append via jQuery

var selectString = '<select id="mobileMenu-'+$this.attr('id')+'" class="mobileMenu">';

//create first option (no value)

selectString += '<option value="">'+settings.topOptionText+'</option>';

//loop through list items

$this.find('li').each(function(){

//when sub-item, indent

var levelStr = '';

var len = $(this).parents('ul, ol').length;

for(i=1;i<len;i++){levelStr += settings.indentString;}

//get url and text for option

var link = $(this).find('a:first-child').attr('href');

var text = levelStr + $(this).clone().children('ul, ol').remove().end().text();

//add option

selectString += '<option value="'+link+'">'+text+'</option>';

});

selectString += '</select>';

//append select element to ul/ol's container

$this.parent().append(selectString);

//add change event handler for mobile menu

$('#mobileMenu-'+$this.attr('id')).change(function(){

goToPage($(this));

});

//hide current menu, show mobile menu

showMenu($this);

} else {

alert('mobileMenu will only work with UL or OL elements!');

}

}

//plugin functionality

function run($this){

//menu doesn't exist

if(isMobile() && !menuExists($this)){

createMenu($this);

}

//menu already exists

else if(isMobile() && menuExists($this)){

showMenu($this);

}

//not mobile browser

else if(!isMobile() && menuExists($this)){

hideMenu($this);

}

}

//run plugin on each matched ul/ol

//maintain chainability by returning "this"

return this.each(function() {

//override the default settings if user provides some

if(options){$.extend(settings, options);}

//cache "this"

var $this = $(this);

//bind event to browser resize

$(window).resize(function(){run($this);});

//run plugin

run($this);

});

};

})(jQuery);