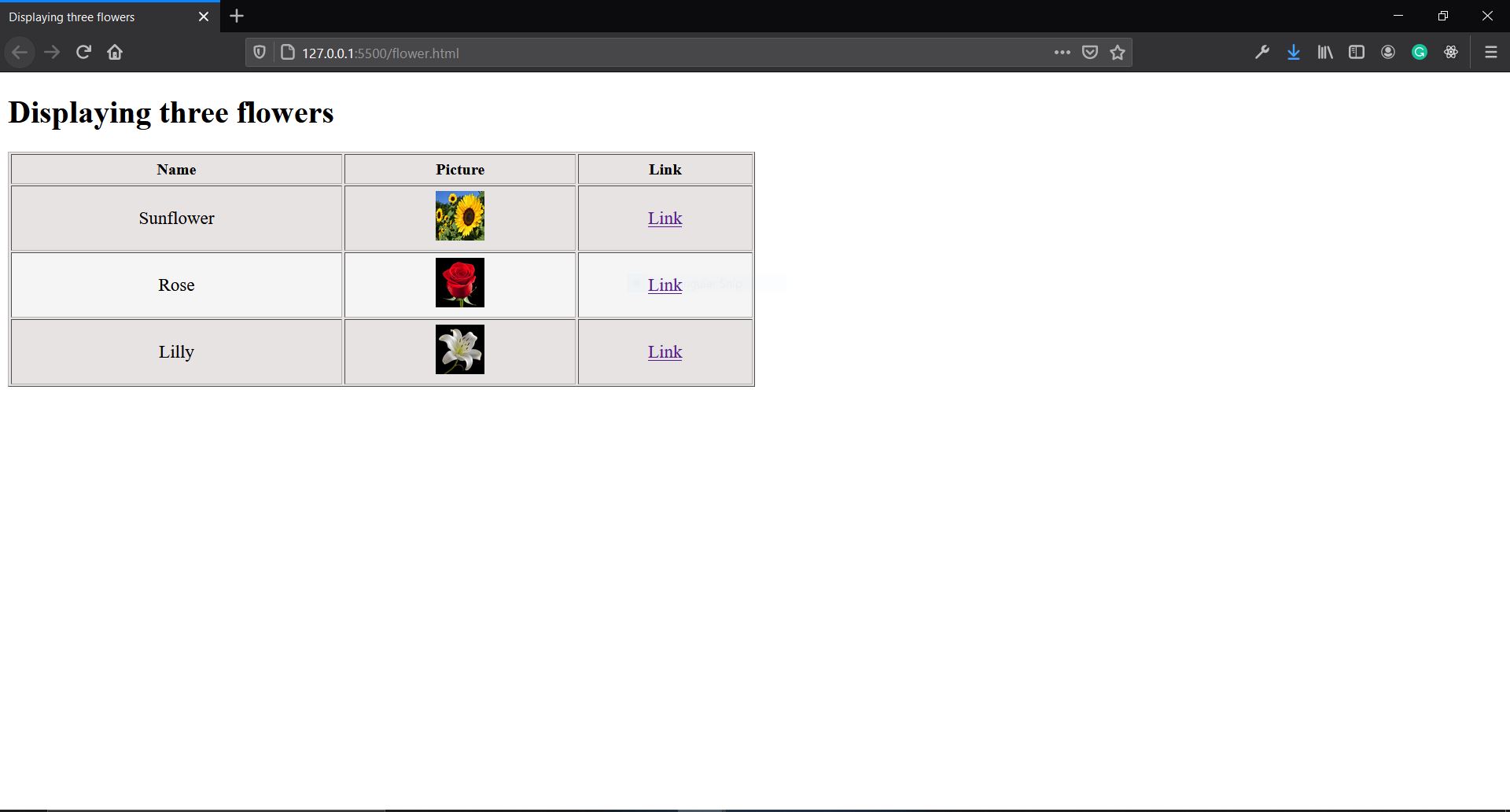
1.



2.

jquery\_one.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>jquery multi scroll plugin</title>

<link rel="stylesheet" type="text/css" href="jquery.multiscroll.css" />

</head>

<body>

<div id="multiscroll">

<div class="ms-left">

<div class="ms-section"><img src="../sunflower.jpg" alt="sunflower" width="300" height="300"></div>

</div>

<div class="ms-right">

<div class="ms-section"><img src="../rose.jpg" alt="rose" width="300" height="300"></div>

</div>

</div>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>

<script type="text/javascript" src="jquery.easings.min.js"></script>

<script type="text/javascript" src="jquery.multiscroll.js"></script>

<script type="text/javascript">

$(document).ready(function() {

$('#multiscroll').multiscroll({

// sectionsColor:[,] // to add different colors

});

});

</script>

</body>

</html>

Css

/\*!

\* multiscroll.js 0.2.2

\* https://github.com/alvarotrigo/multiscroll.js

\* @license MIT licensed

\*

\* Copyright (C) 2016 alvarotrigo.com - A project by Alvaro Trigo

\*/

html, body {

margin: 0;

padding: 0;

overflow:hidden;

-webkit-tap-highlight-color: rgba(0,0,0,0);

}

.ms-section {

position: relative;

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

box-sizing: border-box;

}

.ms-section.ms-table{

display: table;

width: 100%;

}

.ms-tableCell {

display: table-cell;

vertical-align: middle;

width: 100%;

height: 100%;

}

.ms-easing {

-webkit-transition: all 0.7s ease-out;

-moz-transition: all 0.7s ease-out;

-o-transition: all 0.7s ease-out;

transition: all 0.7s ease-out;

}

#multiscroll-nav {

position: fixed;

z-index: 100;

margin-top: -32px;

top: 50%;

opacity: 1;

}

#multiscroll-nav.right {

right: 17px;

}

#multiscroll-nav.left {

left: 17px;

}

#multiscroll-nav ul{

margin: 0;

padding: 0;

}

#multiscroll-nav li{

display: block;

width: 14px;

height: 13px;

margin: 7px;

position:relative;

}

#multiscroll-nav li a{

display: block;

position: relative;

z-index: 1;

width: 100%;

height: 100%;

cursor: pointer;

text-decoration: none;

}

#multiscroll-nav li .active span{

background: #333;

}

#multiscroll-nav span{

top: 2px;

left: 2px;

width: 8px;

height: 8px;

border: 1px solid #000;

background: rgba(0, 0, 0, 0);

-webkit-border-radius: 50%;

-moz-border-radius: 50%;

border-radius: 50%;

position: absolute;

z-index: 1;

}

.multiscroll-tooltip {

position: absolute;

color: #fff;

font-size: 14px;

font-family: arial, helvetica, sans-serif;

top: -2px;

white-space: nowrap;

max-width: 220px;

}

.multiscroll-tooltip.right {

right: 20px;

}

.multiscroll-tooltip.left {

left: 20px;

}

.ms-responsive .ms-hide{

display: none;

}

.ms-scrollable {

overflow: hidden;

position: relative;

}

.ms-scroller{

overflow: hidden;

}

.iScrollIndicator{

border: 0 !important;

}

Js

/\*!

\* multiscroll.js 0.2.2

\* https://github.com/alvarotrigo/multiscroll.js

\* @license MIT licensed

\*

\* Copyright (C) 2016 alvarotrigo.com - A project by Alvaro Trigo

\*/

(function($, window, document, Math, undefined) {

$.fn.multiscroll = function(options) {

var MS = $.fn.multiscroll;

// Create some defaults, extending them with any options that were provided

options = $.extend({

'verticalCentered' : true,

'scrollingSpeed': 700,

'easing': 'easeInQuart',

'menu': false,

'sectionsColor': [],

'anchors':[],

'navigation': false,

'navigationPosition': 'right',

'navigationColor': '#000',

'navigationTooltips': [],

'loopBottom': false,

'loopTop': false,

'css3': false,

'paddingTop': 0,

'paddingBottom': 0,

'fixedElements': null,

'normalScrollElements': null,

'keyboardScrolling': true,

'touchSensitivity': 5,

// Custom selectors

'sectionSelector': '.ms-section',

'leftSelector': '.ms-left',

'rightSelector': '.ms-right',

//events

'afterLoad': null,

'onLeave': null,

'afterRender': null,

'afterResize': null

}, options);

//Defines the delay to take place before being able to scroll to the next section

//BE CAREFUL! Not recommened to change it under 400 for a good behavior in laptops and

//Apple devices (laptops, mouses...)

var scrollDelay = 600;

var isTouchDevice = navigator.userAgent.match(/(iPhone|iPod|iPad|Android|playbook|silk|BlackBerry|BB10|Windows Phone|Tizen|Bada|webOS|IEMobile|Opera Mini)/);

var isTouch = (('ontouchstart' in window) || (navigator.msMaxTouchPoints > 0) || (navigator.maxTouchPoints));

// adding class namef for right and left blocks

if (options.rightSelector !== '.ms-right') {

$(options.rightSelector).addClass('ms-right');

}

if (options.leftSelector !== '.ms-left') {

$(options.leftSelector).addClass('ms-left');

}

var numberSections = $('.ms-left').find('.ms-section').length;

var isMoving = false;

var nav;

var windowHeight = $(window).height();

var MSPointer = getMSPointer();

var events = {

touchmove: 'ontouchmove' in window ? 'touchmove' : MSPointer.move,

touchstart: 'ontouchstart' in window ? 'touchstart' : MSPointer.down

};

//timeouts

var resizeId;

addMouseWheelHandler();

addTouchHandler();

//if css3 is not supported, it will use jQuery animations

if(options.css3){

options.css3 = support3d();

}

$('html, body').css({

'overflow' : 'hidden',

'height' : '100%'

});

//adding class names to each sections

if (options.sectionSelector !== '.ms-section') {

$(options.sectionSelector).each(function(){

$(this).addClass('ms-section');

});

}

//creating the navigation dots

if (options.navigation) {

$('body').append('<div id="multiscroll-nav"><ul></ul></div>');

nav = $('#multiscroll-nav');

nav.css('color', options.navigationColor);

nav.addClass(options.navigationPosition);

}

$('.ms-right, .ms-left').css({

'width': '50%',

'position': 'absolute',

'height': '100%',

'-ms-touch-action': 'none'

});

$('.ms-right').css({

'right': '1px', //http://stackoverflow.com/questions/23675457/chrome-and-opera-creating-small-padding-when-using-displaytable

'top': '0',

'-ms-touch-action': 'none',

'touch-action': 'none'

});

$('.ms-left').css({

'left': '0',

'top': '0',

'-ms-touch-action': 'none',

'touch-action': 'none'

});

$('.ms-left .ms-section, .ms-right .ms-section').each(function(){

var sectionIndex = $(this).index();

if(options.paddingTop || options.paddingBottom){

$(this).css('padding', options.paddingTop + ' 0 ' + options.paddingBottom + ' 0');

}

if (typeof options.sectionsColor[sectionIndex] !== 'undefined') {

$(this).css('background-color', options.sectionsColor[sectionIndex]);

}

if (typeof options.anchors[sectionIndex] !== 'undefined') {

$(this).attr('data-anchor', options.anchors[sectionIndex]);

}

if(options.verticalCentered){

addTableClass($(this));

}

//only for the left panel

if($(this).closest('.ms-left').length && options.navigation) {

var link = '';

if(options.anchors.length){

link = options.anchors[sectionIndex];

}

var tooltip = options.navigationTooltips[sectionIndex];

if(typeof tooltip === 'undefined'){

tooltip = '';

}

if (options.navigation) {

nav.find('ul').append('<li data-tooltip="' + tooltip + '"><a href="#' + link + '"><span></span></a></li>');

}

}

});

//inverting the right panel

$('.ms-right').html( $('.ms-right').find('.ms-section').get().reverse());

$('.ms-left .ms-section, .ms-right .ms-section').each(function(){

var sectionIndex = $(this).index();

$(this).css({

'height': '100%'

});

if(!sectionIndex && options.navigation ){

//activating the navigation bullet

nav.find('li').eq(sectionIndex).find('a').addClass('active');

}

}).promise().done(function(){

//if no active section is defined, the 1st one will be the default one

if(!$('.ms-left .ms-section.active').length){

$('.ms-right').find('.ms-section').last().addClass('active');

$('.ms-left').find('.ms-section').first().addClass('active');

}

//vertical centered of the navigation + first bullet active

if(options.navigation){

nav.css('margin-top', '-' + (nav.height()/2) + 'px');

}

$.isFunction( options.afterRender ) && options.afterRender.call( this);

//scrolling to the defined active section and adjusting right and left panels

silentScroll();

//setting the class for the body element

setBodyClass();

$(window).on('load', function() {

scrollToAnchor();

});

});

//detecting any change on the URL to scroll to the given anchor link

//(a way to detect back history button as we play with the hashes on the URL)

$(window).on('hashchange', hashChangeHandler);

function hashChangeHandler(){

var value = window.location.hash.replace('#', '');

var sectionAnchor = value;

if(sectionAnchor.length){

var section = $('.ms-left').find('[data-anchor="'+sectionAnchor+'"]');

var isFirstScrollMove = (typeof lastScrolledDestiny === 'undefined' );

if (isFirstScrollMove || sectionAnchor !== lastScrolledDestiny){

scrollPage(section);

}

}

};

/\*\*

\* Sliding with arrow keys, both, vertical and horizontal

\*/

$(document).keydown(keydownHandler);

var keydownId;

function keydownHandler(e) {

clearTimeout(keydownId);

var activeElement = $(document.activeElement);

if(!activeElement.is('textarea') && !activeElement.is('input') && !activeElement.is('select') &&

options.keyboardScrolling){

var keyCode = e.which;

//preventing the scroll with arrow keys & spacebar & Page Up & Down keys

var keyControls = [40, 38, 32, 33, 34];

if($.inArray(keyCode, keyControls) > -1){

e.preventDefault();

}

keydownId = setTimeout(function(){

onkeydown(e);

},150);

}

}

/\*\*

\* Sliding with arrow keys, both, vertical and horizontal

\*/

function onkeydown(e){

var shiftPressed = e.shiftKey;

switch (e.which) {

//up

case 38:

case 33:

MS.moveSectionUp();

break;

//down

case 32: //spacebar

if(shiftPressed){

MS.moveSectionUp();

break;

}

case 40:

case 34:

MS.moveSectionDown();

break;

//Home

case 36:

MS.moveTo(1);

break;

//End

case 35:

MS.moveTo( $('.ms-left .ms-section').length);

break;

default:

return; // exit this handler for other keys

}

}

/\*\*

\* Disabling any action when pressing of the mouse wheel (Chrome, IE, Opera, Safari)

\*/

$(document).mousedown(function(e) {

if(e.button == 1){

e.preventDefault();

return false;

}

});

function navClickHandler(e) {

e.preventDefault();

var index = $(this).parent().index();

scrollPage($('.ms-left .ms-section').eq(index));

}

//navigation action

$(document).on('click', '#multiscroll-nav a', navClickHandler);

function navMouseEnterHandler() {

var tooltip = $(this).data('tooltip');

$('<div class="multiscroll-tooltip ' + options.navigationPosition +'">' + tooltip + '</div>').hide().appendTo($(this)).fadeIn(200);

}

function navMouseLeaveHandler() {

$(this).find('.multiscroll-tooltip').fadeOut(200, function() {

$(this).remove();

});

}

//navigation tooltips

$(document).on({

mouseenter: navMouseEnterHandler,

mouseleave: navMouseLeaveHandler

}, '#multiscroll-nav li');

if(options.normalScrollElements){

$(document).on('mouseenter', options.normalScrollElements, function () {

MS.setMouseWheelScrolling(false);

});

$(document).on('mouseleave', options.normalScrollElements, function(){

MS.setMouseWheelScrolling(true);

});

}

//when resizing the site, we adjust the heights of the sections

$(window).on('resize', resizeHandler);

var previousHeight = windowHeight;

/\*\*

\* When resizing is finished, we adjust the slides sizes and positions

\*/

function resizeHandler() {

// rebuild immediately on touch devices

if (isTouchDevice) {

var activeElement = $(document.activeElement);

//if the keyboard is NOT visible

if (!activeElement.is('textarea') && !activeElement.is('input') && !activeElement.is('select')) {

var currentHeight = $(window).height();

//making sure the change in the viewport size is enough to force a rebuild. (20 % of the window to avoid problems when hidding scroll bars)

if( Math.abs(currentHeight - previousHeight) > (20 \* Math.max(previousHeight, currentHeight) / 100) ){

reBuild(true);

previousHeight = currentHeight;

}

}

}else{

//in order to call the functions only when the resize is finished

//http://stackoverflow.com/questions/4298612/jquery-how-to-call-resize-event-only-once-its-finished-resizing

clearTimeout(resizeId);

resizeId = setTimeout(function(){

reBuild(true);

}, 350);

}

}

/\*\*

\* When resizing is finished, we adjust the slides sizes and positions

\*/

function reBuild(resizing){

windowHeight = $(window).height();

$('.ms-tableCell').each(function() {

$(this).css({ height: getTableHeight($(this).parent()) });

});

if(options.scrollOverflow){

scrollBarHandler.createScrollBarForAll();

}

silentScroll();

$.isFunction( options.afterResize ) && options.afterResize.call( this);

}

function silentScroll(){

//moving the right section to the bottom

if(options.css3){

transformContainer($('.ms-left'), 'translate3d(0px, -' + $('.ms-left').find('.ms-section.active').position().top + 'px, 0px)', false);

transformContainer($('.ms-right'), 'translate3d(0px, -' + $('.ms-right').find('.ms-section.active').position().top + 'px, 0px)', false);

}else{

$('.ms-left').css('top', -$('.ms-left').find('.ms-section.active').position().top );

$('.ms-right').css('top', -$('.ms-right').find('.ms-section.active').position().top );

}

}

MS.moveSectionUp = function(){

var prev = $('.ms-left .ms-section.active').prev('.ms-section');

if(!prev.length && options.loopTop){

prev = $('.ms-left .ms-section').last();

}

if (prev.length) {

scrollPage(prev);

}

};

MS.moveSectionDown = function (){

var next = $('.ms-left .ms-section.active').next('.ms-section');

if(!next.length && options.loopBottom ){

next = $('.ms-left .ms-section').first();

}

if(next.length){

scrollPage(next);

}

};

MS.moveTo = function (section){

var destiny = '';

if(isNaN(section)){

destiny = $('.ms-left [data-anchor="'+section+'"]');

}else{

destiny = $('.ms-left .ms-section').eq( (section -1) );

}

scrollPage(destiny);

};

function scrollPage(leftDestination){

var leftDestinationIndex = leftDestination.index();

var rightDestination = $('.ms-right').find('.ms-section').eq( numberSections -1 - leftDestinationIndex);

var rightDestinationIndex = numberSections - 1 - leftDestinationIndex;

var anchorLink = leftDestination.data('anchor');

var activeSection = $('.ms-left .ms-section.active');

var leavingSection = activeSection.index() + 1;

var yMovement = getYmovement(leftDestination);

//preventing from activating the MouseWheelHandler event

//more than once if the page is scrolling

isMoving = true;

var topPos = {

'left' : leftDestination.position().top,

'right': rightDestination.position().top

};

rightDestination.addClass('active').siblings().removeClass('active');

leftDestination.addClass('active').siblings().removeClass('active');

setURLHash(anchorLink);

// Use CSS3 translate functionality or...

if (options.css3){

//callback (onLeave)

$.isFunction(options.onLeave) && options.onLeave.call(this, leavingSection, (leftDestinationIndex + 1), yMovement);

var translate3dLeft = 'translate3d(0px, -' + topPos['left'] + 'px, 0px)';

var translate3dRight = 'translate3d(0px, -' + topPos['right'] + 'px, 0px)';

transformContainer($('.ms-left'), translate3dLeft, true);

transformContainer($('.ms-right'), translate3dRight, true);

setTimeout(function () {

//callback (afterLoad)

$.isFunction(options.afterLoad) && options.afterLoad.call(this, anchorLink, (leftDestinationIndex + 1));

setTimeout(function () {

isMoving = false;

}, scrollDelay);

}, options.scrollingSpeed);

}else{

//callback (onLeave)

$.isFunction(options.onLeave) && options.onLeave.call(this, leavingSection, (leftDestinationIndex + 1), yMovement);

$('.ms-left').animate({

'top': -topPos['left']

}, options.scrollingSpeed, options.easing, function(){

$.isFunction(options.afterLoad) && options.afterLoad.call(this, anchorLink, (leftDestinationIndex + 1));

setTimeout(function () {

isMoving = false;

}, scrollDelay);

});

$('.ms-right').animate({

'top': -topPos['right']

}, options.scrollingSpeed, options.easing);

}

//flag to avoid callingn `scrollPage()` twice in case of using anchor links

lastScrolledDestiny = anchorLink;

activateMenuElement(anchorLink);

activateNavDots(anchorLink, leftDestinationIndex);

}

/\*\*

\* Removes the auto scrolling action fired by the mouse wheel and tackpad.

\* After this function is called, the mousewheel and trackpad movements won't scroll through sections.

\*/

function removeMouseWheelHandler(){

if (document.addEventListener) {

document.removeEventListener('mousewheel', MouseWheelHandler, false); //IE9, Chrome, Safari, Oper

document.removeEventListener('wheel', MouseWheelHandler, false); //Firefox

} else {

document.detachEvent("onmousewheel", MouseWheelHandler); //IE 6/7/8

}

}

/\*\*

\* Adds the auto scrolling action for the mouse wheel and tackpad.

\* After this function is called, the mousewheel and trackpad movements will scroll through sections

\*/

function addMouseWheelHandler(){

if (document.addEventListener) {

document.addEventListener("mousewheel", MouseWheelHandler, false); //IE9, Chrome, Safari, Oper

document.addEventListener("wheel", MouseWheelHandler, false); //Firefox

} else {

document.attachEvent("onmousewheel", MouseWheelHandler); //IE 6/7/8

}

}

/\*\*

\* Detecting mousewheel scrolling

\*

\* http://blogs.sitepointstatic.com/examples/tech/mouse-wheel/index.html

\* http://www.sitepoint.com/html5-javascript-mouse-wheel/

\*/

function MouseWheelHandler(e) {

// cross-browser wheel delta

e = window.event || e;

var delta = Math.max(-1, Math.min(1,

(e.wheelDelta || -e.deltaY || -e.detail)));

if (!isMoving) { //if theres any #

//scrolling down?

if (delta < 0) {

MS.moveSectionDown();

}

//scrolling up?

else {

MS.moveSectionUp();

}

}

return false;

}

/\*\*

\* Adds a css3 transform property to the container class with or without animation depending on the animated param.

\*/

function transformContainer(container, translate3d, animated){

container.toggleClass('ms-easing', animated);

container.css(getTransforms(translate3d));

}

/\*\*

\* Returns the transform styles for all browsers

\*/

function getTransforms(translate3d){

return {

'-webkit-transform': translate3d,

'-moz-transform': translate3d,

'-ms-transform':translate3d,

'transform': translate3d

};

}

/\*\*

\* Activating the website navigation dots according to the given slide name.

\*/

function activateNavDots(name, sectionIndex){

if(options.navigation){

$('#multiscroll-nav').find('.active').removeClass('active');

if(name){

$('#multiscroll-nav').find('a[href="#' + name + '"]').addClass('active');

}else{

$('#multiscroll-nav').find('li').eq(sectionIndex).find('a').addClass('active');

}

}

}

/\*\*

\* Activating the website main menu elements according to the given slide name.

\*/

function activateMenuElement(name){

if(options.menu){

$(options.menu).find('.active').removeClass('active');

$(options.menu).find('[data-menuanchor="'+name+'"]').addClass('active');

}

}

/\*\*

\* Retuns `up` or `down` depending on the scrolling movement to reach its destination

\* from the current section.

\*/

function getYmovement(destiny){

var fromIndex = $('.ms-left .ms-section.active').index();

var toIndex = destiny.index();

if(fromIndex > toIndex){

return 'up';

}

return 'down';

}

/\*\*

\* Sets the URL hash for a section with slides

\*/

function setURLHash(anchorLink){

if(options.anchors.length){

location.hash = anchorLink;

}

setBodyClass();

}

/\*\*

\* Sets a class for the body of the page depending on the active section / slide

\*/

function setBodyClass(){

var section = $('.ms-left .ms-section.active');

var sectionAnchor = section.data('anchor');

var sectionIndex = section.index();

var text = String(sectionIndex);

if(options.anchors.length){

text = sectionAnchor;

}

//changing slash for dash to make it a valid CSS style

text = text.replace('/', '-').replace('#','');

//removing previous anchor classes

var classRe = new RegExp('\\b\\s?' + 'ms-viewing' + '-[^\\s]+\\b', "g");

$('body')[0].className = $('body')[0].className.replace(classRe, '');

//adding the current anchor

$('body').addClass('ms-viewing-' + text);

}

/\*\*

\* Checks for translate3d support

\* @return boolean

\* http://stackoverflow.com/questions/5661671/detecting-transform-translate3d-support

\*/

function support3d() {

var el = document.createElement('p'),

has3d,

transforms = {

'webkitTransform':'-webkit-transform',

'OTransform':'-o-transform',

'msTransform':'-ms-transform',

'MozTransform':'-moz-transform',

'transform':'transform'

};

// Add it to the body to get the computed style.

document.body.insertBefore(el, null);

for (var t in transforms) {

if (el.style[t] !== undefined) {

el.style[t] = "translate3d(1px,1px,1px)";

has3d = window.getComputedStyle(el).getPropertyValue(transforms[t]);

}

}

document.body.removeChild(el);

return (has3d !== undefined && has3d.length > 0 && has3d !== "none");

}

/\*\*

\* Wraps an element in order to center it vertically by using a class style.

\*/

function addTableClass(element){

element.addClass('ms-table').wrapInner('<div class="ms-tableCell" style="height: ' + getTableHeight(element) + 'px" />');

}

/\*\*

\* Gets the height of the section after removing the paddings.

\*/

function getTableHeight(section){

var sectionHeight = windowHeight;

if(options.paddingTop || options.paddingBottom){

var paddings = parseInt(section.css('padding-top')) + parseInt(section.css('padding-bottom'));

sectionHeight = (windowHeight - paddings);

}

return sectionHeight;

}

/\*\*

\* Scrolls the page to the existent anchor in the URL

\*/

function scrollToAnchor(){

//getting the anchor link in the URL and deleting the `#`

var sectionAnchor = window.location.hash.replace('#', '');

var section = $('.ms-left .ms-section[data-anchor="'+sectionAnchor+'"]');

if(sectionAnchor.length){ //if theres any #

scrollPage(section);

}

}

/\*\*

\* Adds or remove the possiblity of scrolling through sections by using the keyboard arrow keys

\*/

MS.setKeyboardScrolling = function (value){

options.keyboardScrolling = value;

};

/\*\*

\* Adds or remove the possiblity of scrolling through sections by using the mouse wheel or the trackpad.

\*/

MS.setMouseWheelScrolling = function (value){

if(value){

addMouseWheelHandler();

}else{

removeMouseWheelHandler();

}

};

/\*\*

\* Defines the scrolling speed

\*/

MS.setScrollingSpeed = function(value){

options.scrollingSpeed = value;

};

var touchStartY = 0;

var touchStartX = 0;

var touchEndY = 0;

var touchEndX = 0;

/\* Detecting touch events

\* As we are changing the top property of the page on scrolling, we can not use the traditional way to detect it.

\* This way, the touchstart and the touch moves shows an small difference between them which is the

\* used one to determine the direction.

\*/

function touchMoveHandler(e){

if(isReallyTouch(e)){

//preventing the easing on iOS devices

event.preventDefault();

var activeSection = $('.ms-left .ms-section.active');

if (!isMoving) { //if theres any #

var touchEvents = getEventsPage(e);

touchEndY = touchEvents['y'];

touchEndX = touchEvents['x'];

//is the movement greater than the minimum resistance to scroll?

if (Math.abs(touchStartY - touchEndY) > ($(window).height() / 100 \* options.touchSensitivity)) {

if (touchStartY > touchEndY) {

MS.moveSectionDown();

} else if (touchEndY > touchStartY) {

MS.moveSectionUp();

}

}

}

}

}

/\*\*

\* As IE >= 10 fires both touch and mouse events when using a mouse in a touchscreen

\* this way we make sure that is really a touch event what IE is detecting.

\*/

function isReallyTouch(e){

//if is not IE || IE is detecting `touch` or `pen`

return typeof e.pointerType === 'undefined' || e.pointerType != 'mouse';

}

/\*\*

\* Handler to get he coordinates of the starting touch

\*/

function touchStartHandler(e){

if(isReallyTouch(e)){

var touchEvents = getEventsPage(e);

touchStartY = touchEvents['y'];

touchStartX = touchEvents['x'];

}

}

/\*\*

\* Adds the possibility to auto scroll through sections on touch devices.

\*/

function addTouchHandler(){

if(isTouch || isTouchDevice){

document.removeEventListener(events.touchstart, touchStartHandler);

document.removeEventListener(events.touchmove, touchMoveHandler, {passive: false});

document.addEventListener(events.touchstart, touchStartHandler);

document.addEventListener(events.touchmove, touchMoveHandler, {passive: false});

}

}

/\*\*

\* Removes the auto scrolling for touch devices.

\*/

function removeTouchHandler(){

if(isTouch || isTouchDevice){

document.removeEventListener(events.touchstart, touchStartHandler);

document.removeEventListener(events.touchmove, touchMoveHandler, {passive: false});

}

}

/\*

\* Returns and object with Microsoft pointers (for IE<11 and for IE >= 11)

\* http://msdn.microsoft.com/en-us/library/ie/dn304886(v=vs.85).aspx

\*/

function getMSPointer(){

var pointer;

//IE >= 11

if(window.PointerEvent){

pointer = { down: "pointerdown", move: "pointermove"};

}

//IE < 11

else{

pointer = { down: "MSPointerDown", move: "MSPointerMove"};

}

return pointer;

}

/\*\*

\* Gets the pageX and pageY properties depending on the browser.

\* https://github.com/alvarotrigo/fullPage.js/issues/194#issuecomment-34069854

\*/

function getEventsPage(e){

var events = [];

events.y = (typeof e.pageY !== 'undefined' && (e.pageY || e.pageX) ? e.pageY : e.touches[0].pageY);

events.x = (typeof e.pageX !== 'undefined' && (e.pageY || e.pageX) ? e.pageX : e.touches[0].pageX);

//in touch devices with scrollBar:true, e.pageY is detected, but we have to deal with touch events. #1008

if(isTouch && isReallyTouch(e) && typeof e.touches !== 'undefined'){

events.y = e.touches[0].pageY;

events.x = e.touches[0].pageX;

}

return events;

}

/\*\*

\* Destroy multiscroll.js plugin's events

\*/

MS.destroy = function() {

MS.setKeyboardScrolling(false);

MS.setMouseWheelScrolling(false);

removeTouchHandler();

$(window)

.off('hashchange', hashChangeHandler)

.off('resize', resizeHandler);

$(document)

.off('mouseenter', '#multiscroll-nav li')

.off('mouseleave', '#multiscroll-nav li')

.off('click', '#multiscroll-nav a');

};

/\*\*

\* Build multiscroll.js plugin's events after destroy

\*/

MS.build = function() {

MS.setKeyboardScrolling(true);

MS.setMouseWheelScrolling(true);

addTouchHandler();

$(window)

.on('hashchange', hashChangeHandler)

.on('resize', resizeHandler);

$(document)

.on('mouseenter', '#multiscroll-nav li', navMouseEnterHandler)

.on('mouseleave', '#multiscroll-nav li', navMouseLeaveHandler)

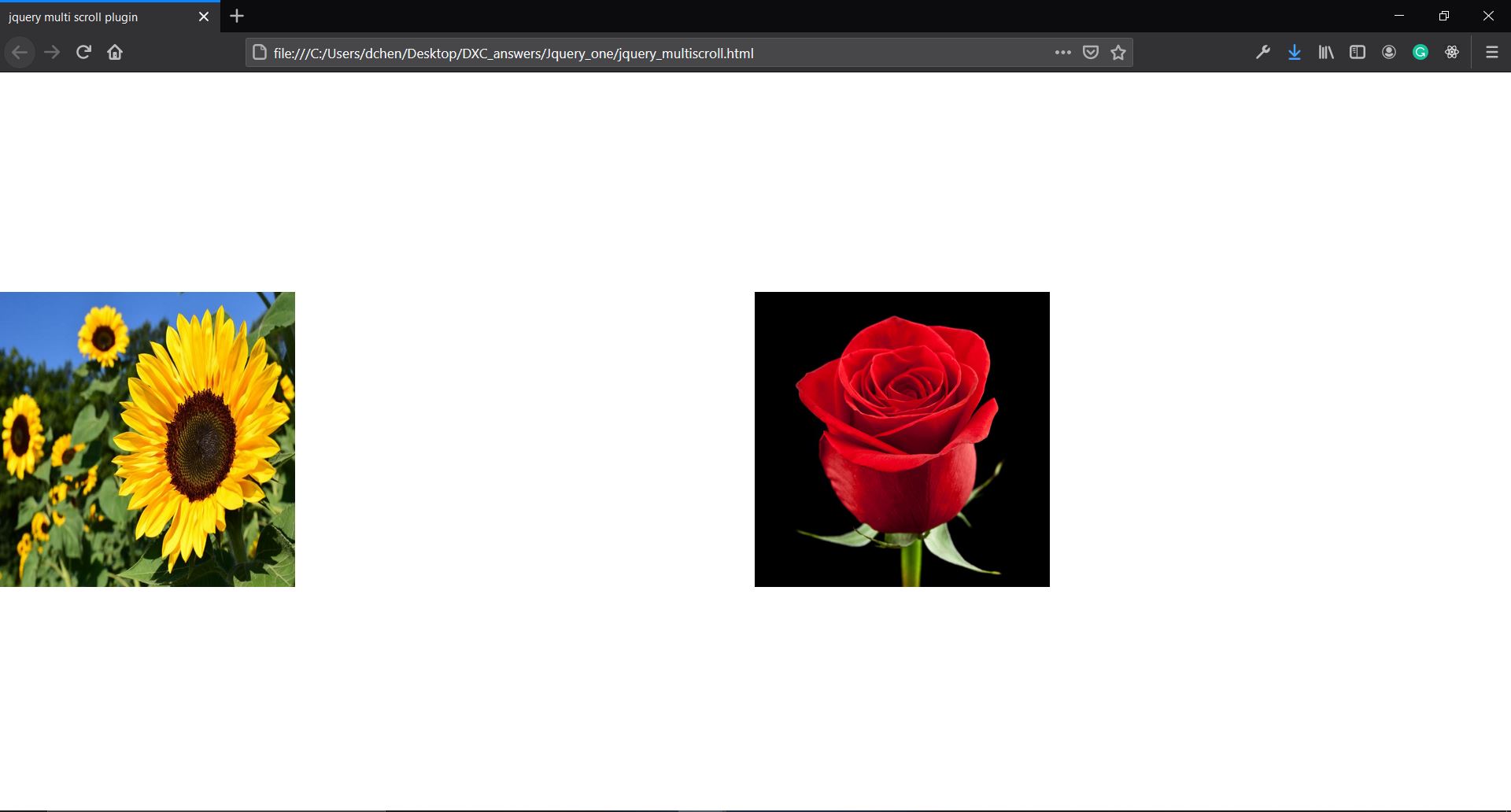
.on('click', '#multiscroll-nav a', navClickHandler);

};

};

})(jQuery, window, document, Math);

Screenshot:



Jquery\_two

.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Jquery plugin</title>

<link rel="stylesheet" type="text/css" href="jquery.pagepiling.css" />

</head>

<body>

<div id="pagepiling">

<div class="section"><img src="../sunflower.jpg" alt=""></div>

<div class="section"><img src="../rose.jpg" alt=""></div>

<div class="section"><img src="../lilly.jpg" alt=""></div>

</div>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script type="text/javascript" src="jquery.pagepiling.js"></script>

<script type="text/javascript">

$(document).ready(function() {

$('#pagepiling').pagepiling();

});

</script>

</body>

</html>

Css

/\*!

\* pagepiling.js 1.5.6

\*

\* https://github.com/alvarotrigo/pagePiling.js

\* @license MIT licensed

\*

\* Copyright (C) 2016 alvarotrigo.com - A project by Alvaro Trigo

\*/

html, body {

overflow:hidden;

margin:0;

padding:0;

/\*Avoid flicker on slides transitions for mobile phones #336 \*/

-webkit-tap-highlight-color: rgba(0,0,0,0);

}

.pp-section {

height:100%;

position:absolute;

width:100%;

/\* fixes flickering in firefox\*/

backface-visibility: hidden;

}

.pp-easing {

-webkit-transition: all 1000ms cubic-bezier(0.550, 0.085, 0.000, 0.990);

-moz-transition: all 1000ms cubic-bezier(0.550, 0.085, 0.000, 0.990);

-o-transition: all 1000ms cubic-bezier(0.550, 0.085, 0.000, 0.990);

transition: all 1000ms cubic-bezier(0.550, 0.085, 0.000, 0.990);

/\* custom \*/

-webkit-transition-timing-function: cubic-bezier(0.550, 0.085, 0.000, 0.990);

-moz-transition-timing-function: cubic-bezier(0.550, 0.085, 0.000, 0.990);

-o-transition-timing-function: cubic-bezier(0.550, 0.085, 0.000, 0.990);

transition-timing-function: cubic-bezier(0.550, 0.085, 0.000, 0.990);

/\* custom \*/

}

#pp-nav {

position: fixed;

z-index: 100;

margin-top: -32px;

top: 50%;

opacity: 1;

}

#pp-nav.right {

right: 17px;

}

#pp-nav.left {

left: 17px;

}

.pp-section.pp-table{

display: table;

}

.pp-tableCell {

display: table-cell;

vertical-align: middle;

width: 100%;

height: 100%;

}

.pp-slidesNav{

position: absolute;

z-index: 4;

left: 50%;

opacity: 1;

}

.pp-slidesNav.bottom {

bottom: 17px;

}

.pp-slidesNav.top {

top: 17px;

}

#pp-nav ul,

.pp-slidesNav ul {

margin: 0;

padding: 0;

}

#pp-nav li,

.pp-slidesNav li {

display: block;

width: 14px;

height: 13px;

margin: 7px;

position:relative;

}

.pp-slidesNav li {

display: inline-block;

}

#pp-nav li a,

.pp-slidesNav li a {

display: block;

position: relative;

z-index: 1;

width: 100%;

height: 100%;

cursor: pointer;

text-decoration: none;

}

#pp-nav li .active span,

.pp-slidesNav .active span {

background: #333;

}

#pp-nav span,

.pp-slidesNav span {

top: 2px;

left: 2px;

width: 8px;

height: 8px;

border: 1px solid #000;

background: rgba(0, 0, 0, 0);

border-radius: 50%;

position: absolute;

z-index: 1;

}

.pp-tooltip {

position: absolute;

top: -2px;

color: #fff;

font-size: 14px;

font-family: arial, helvetica, sans-serif;

white-space: nowrap;

max-width: 220px;

}

.pp-tooltip.right {

right: 20px;

}

.pp-tooltip.left {

left: 20px;

}

.pp-scrollable{

overflow-y: scroll;

height: 100%;

}

Js

/\*!

\* pagepiling.js 1.5.6

\*

\* https://github.com/alvarotrigo/pagePiling.js

\* @license MIT licensed

\*

\* Copyright (C) 2016 alvarotrigo.com - A project by Alvaro Trigo

\*/

(function ($, document, window, undefined) {

'use strict';

$.fn.pagepiling = function (custom) {

var PP = $.fn.pagepiling;

var container = $(this);

var lastScrolledDestiny;

var lastAnimation = 0;

var isTouch = (('ontouchstart' in window) || (navigator.msMaxTouchPoints > 0) || (navigator.maxTouchPoints));

var touchStartY = 0, touchStartX = 0, touchEndY = 0, touchEndX = 0;

var scrollings = [];

//Defines the delay to take place before being able to scroll to the next section

//BE CAREFUL! Not recommened to change it under 400 for a good behavior in laptops and

//Apple devices (laptops, mouses...)

var scrollDelay = 600;

// Create some defaults, extending them with any options that were provided

var options = $.extend(true, {

direction: 'vertical',

menu: null,

verticalCentered: true,

sectionsColor: [],

anchors: [],

scrollingSpeed: 700,

easing: 'easeInQuart',

loopBottom: false,

loopTop: false,

css3: true,

navigation: {

textColor: '#000',

bulletsColor: '#000',

position: 'right',

tooltips: []

},

normalScrollElements: null,

normalScrollElementTouchThreshold: 5,

touchSensitivity: 5,

keyboardScrolling: true,

sectionSelector: '.section',

animateAnchor: false,

//events

afterLoad: null,

onLeave: null,

afterRender: null

}, custom);

//easeInQuart animation included in the plugin

$.extend($.easing,{ easeInQuart: function (x, t, b, c, d) { return c\*(t/=d)\*t\*t\*t + b; }});

/\*\*

\* Defines the scrolling speed

\*/

PP.setScrollingSpeed = function(value){

options.scrollingSpeed = value;

};

/\*\*

\* Adds or remove the possiblity of scrolling through sections by using the mouse wheel or the trackpad.

\*/

PP.setMouseWheelScrolling = function (value){

if(value){

addMouseWheelHandler();

}else{

removeMouseWheelHandler();

}

};

/\*\*

\* Adds or remove the possiblity of scrolling through sections by using the mouse wheel/trackpad or touch gestures.

\*/

PP.setAllowScrolling = function (value){

if(value){

PP.setMouseWheelScrolling(true);

addTouchHandler();

}else{

PP.setMouseWheelScrolling(false);

removeTouchHandler();

}

};

/\*\*

\* Adds or remove the possiblity of scrolling through sections by using the keyboard arrow keys

\*/

PP.setKeyboardScrolling = function (value){

options.keyboardScrolling = value;

};

/\*\*

\* Moves sectio up

\*/

PP.moveSectionUp = function () {

var prev = $('.pp-section.active').prev('.pp-section');

//looping to the bottom if there's no more sections above

if (!prev.length && options.loopTop) {

prev = $('.pp-section').last();

}

if (prev.length) {

scrollPage(prev);

}

};

/\*\*

\* Moves sectio down

\*/

PP.moveSectionDown = function () {

var next = $('.pp-section.active').next('.pp-section');

//looping to the top if there's no more sections below

if(!next.length && options.loopBottom){

next = $('.pp-section').first();

}

if (next.length) {

scrollPage(next);

}

};

/\*\*

\* Moves the site to the given anchor or index

\*/

PP.moveTo = function (section){

var destiny = '';

if(isNaN(section)){

destiny = $(document).find('[data-anchor="'+section+'"]');

}else{

destiny = $('.pp-section').eq( (section -1) );

}

if(destiny.length > 0){

scrollPage(destiny);

}

};

//adding internal class names to void problem with common ones

$(options.sectionSelector).each(function(){

$(this).addClass('pp-section');

});

//if css3 is not supported, it will use jQuery animations

if(options.css3){

options.css3 = support3d();

}

$(container).css({

'overflow' : 'hidden',

'-ms-touch-action': 'none', /\* Touch detection for Windows 8 \*/

'touch-action': 'none' /\* IE 11 on Windows Phone 8.1\*/

});

//init

PP.setAllowScrolling(true);

//creating the navigation dots

if (!$.isEmptyObject(options.navigation) ) {

addVerticalNavigation();

}

var zIndex = $('.pp-section').length;

$('.pp-section').each(function (index) {

$(this).data('data-index', index);

$(this).css('z-index', zIndex);

//if no active section is defined, the 1st one will be the default one

if (!index && $('.pp-section.active').length === 0) {

$(this).addClass('active');

}

if (typeof options.anchors[index] !== 'undefined') {

$(this).attr('data-anchor', options.anchors[index]);

}

if (typeof options.sectionsColor[index] !== 'undefined') {

$(this).css('background-color', options.sectionsColor[index]);

}

if(options.verticalCentered && !$(this).hasClass('pp-scrollable')){

addTableClass($(this));

}

zIndex = zIndex - 1;

}).promise().done(function(){

//vertical centered of the navigation + first bullet active

if(options.navigation){

$('#pp-nav').css('margin-top', '-' + ($('#pp-nav').height()/2) + 'px');

$('#pp-nav').find('li').eq($('.pp-section.active').index('.pp-section')).find('a').addClass('active');

}

$(window).on('load', function() {

scrollToAnchor();

});

$.isFunction( options.afterRender ) && options.afterRender.call( this);

});

/\*\*

\* Enables vertical centering by wrapping the content and the use of table and table-cell

\*/

function addTableClass(element){

element.addClass('pp-table').wrapInner('<div class="pp-tableCell" style="height:100%" />');

}

/\*\*

\* Retuns `up` or `down` depending on the scrolling movement to reach its destination

\* from the current section.

\*/

function getYmovement(destiny){

var fromIndex = $('.pp-section.active').index('.pp-section');

var toIndex = destiny.index('.pp-section');

if(fromIndex > toIndex){

return 'up';

}

return 'down';

}

/\*\*

\* Scrolls the page to the given destination

\*/

function scrollPage(destination, animated) {

var v ={

destination: destination,

animated: animated,

activeSection: $('.pp-section.active'),

anchorLink: destination.data('anchor'),

sectionIndex: destination.index('.pp-section'),

toMove: destination,

yMovement: getYmovement(destination),

leavingSection: $('.pp-section.active').index('.pp-section') + 1

};

//quiting when activeSection is the target element

if(v.activeSection.is(destination)){ return; }

if(typeof v.animated === 'undefined'){

v.animated = true;

}

if(typeof v.anchorLink !== 'undefined'){

setURLHash(v.anchorLink, v.sectionIndex);

}

v.destination.addClass('active').siblings().removeClass('active');

v.sectionsToMove = getSectionsToMove(v);

//scrolling down (moving sections up making them disappear)

if (v.yMovement === 'down') {

v.translate3d = getTranslate3d();

v.scrolling = '-100%';

if(!options.css3){

v.sectionsToMove.each(function(index){

if(index != v.activeSection.index('.pp-section')){

$(this).css(getScrollProp(v.scrolling));

}

});

}

v.animateSection = v.activeSection;

}

//scrolling up (moving section down to the viewport)

else {

v.translate3d = 'translate3d(0px, 0px, 0px)';

v.scrolling = '0';

v.animateSection = destination;

}

$.isFunction(options.onLeave) && options.onLeave.call(this, v.leavingSection, (v.sectionIndex + 1), v.yMovement);

performMovement(v);

activateMenuElement(v.anchorLink);

activateNavDots(v.anchorLink, v.sectionIndex);

lastScrolledDestiny = v.anchorLink;

var timeNow = new Date().getTime();

lastAnimation = timeNow;

}

/\*\*

\* Performs the movement (by CSS3 or by jQuery)

\*/

function performMovement(v){

if(options.css3){

transformContainer(v.animateSection, v.translate3d, v.animated);

v.sectionsToMove.each(function(){

transformContainer($(this), v.translate3d, v.animated);

});

setTimeout(function () {

afterSectionLoads(v);

}, options.scrollingSpeed);

}else{

v.scrollOptions = getScrollProp(v.scrolling);

if(v.animated){

v.animateSection.animate(

v.scrollOptions,

options.scrollingSpeed, options.easing, function () {

readjustSections(v);

afterSectionLoads(v);

});

}else{

v.animateSection.css(getScrollProp(v.scrolling));

setTimeout(function(){

readjustSections(v);

afterSectionLoads(v);

},400);

}

}

}

/\*\*

\* Actions to execute after a secion is loaded

\*/

function afterSectionLoads(v){

//callback (afterLoad) if the site is not just resizing and readjusting the slides

$.isFunction(options.afterLoad) && options.afterLoad.call(this, v.anchorLink, (v.sectionIndex + 1));

}

function getSectionsToMove(v){

var sectionToMove;

if(v.yMovement === 'down'){

sectionToMove = $('.pp-section').map(function(index){

if (index < v.destination.index('.pp-section')){

return $(this);

}

});

}else{

sectionToMove = $('.pp-section').map(function(index){

if (index > v.destination.index('.pp-section')){

return $(this);

}

});

}

return sectionToMove;

}

/\*\*

\* Returns the sections to re-adjust in the background after the section loads.

\*/

function readjustSections(v){

if(v.yMovement === 'up'){

v.sectionsToMove.each(function(index){

$(this).css(getScrollProp(v.scrolling));

});

}

}

/\*\*

\* Gets the property used to create the scrolling effect when using jQuery animations

\* depending on the plugin direction option.

\*/

function getScrollProp(propertyValue){

if(options.direction === 'vertical'){

return {'top': propertyValue};

}

return {'left': propertyValue};

}

/\*\*

\* Scrolls the site without anymations (usually used in the background without the user noticing it)

\*/

function silentScroll(section, offset){

if (options.css3) {

transformContainer(section, getTranslate3d(), false);

}

else{

section.css(getScrollProp(offset));

}

}

/\*\*

\* Sets the URL hash for a section with slides

\*/

function setURLHash(anchorLink, sectionIndex){

if(options.anchors.length){

location.hash = anchorLink;

setBodyClass(location.hash);

}else{

setBodyClass(String(sectionIndex));

}

}

/\*\*

\* Sets a class for the body of the page depending on the active section / slide

\*/

function setBodyClass(text){

//removing the #

text = text.replace('#','');

//removing previous anchor classes

$('body')[0].className = $('body')[0].className.replace(/\b\s?pp-viewing-[^\s]+\b/g, '');

//adding the current anchor

$('body').addClass('pp-viewing-' + text);

}

//TO DO

function scrollToAnchor(){

//getting the anchor link in the URL and deleting the `#`

var value = window.location.hash.replace('#', '');

var sectionAnchor = value;

var section = $(document).find('.pp-section[data-anchor="'+sectionAnchor+'"]');

if(section.length > 0){ //if theres any #

scrollPage(section, options.animateAnchor);

}

}

/\*\*

\* Determines if the transitions between sections still taking place.

\* The variable `scrollDelay` adds a "save zone" for devices such as Apple laptops and Apple magic mouses

\*/

function isMoving(){

var timeNow = new Date().getTime();

// Cancel scroll if currently animating or within quiet period

if (timeNow - lastAnimation < scrollDelay + options.scrollingSpeed) {

return true;

}

return false;

}

//detecting any change on the URL to scroll to the given anchor link

//(a way to detect back history button as we play with the hashes on the URL)

$(window).on('hashchange', hashChangeHandler);

/\*\*

\* Actions to do when the hash (#) in the URL changes.

\*/

function hashChangeHandler(){

var value = window.location.hash.replace('#', '').split('/');

var sectionAnchor = value[0];

if(sectionAnchor.length){

/\*in order to call scrollpage() only once for each destination at a time

It is called twice for each scroll otherwise, as in case of using anchorlinks `hashChange`

event is fired on every scroll too.\*/

if (sectionAnchor && sectionAnchor !== lastScrolledDestiny) {

var section;

if(isNaN(sectionAnchor)){

section = $(document).find('[data-anchor="'+sectionAnchor+'"]');

}else{

section = $('.pp-section').eq( (sectionAnchor -1) );

}

scrollPage(section);

}

}

}

/\*\*

\* Cross browser transformations

\*/

function getTransforms(translate3d) {

return {

'-webkit-transform': translate3d,

'-moz-transform': translate3d,

'-ms-transform': translate3d,

'transform': translate3d

};

}

/\*\*

\* Adds a css3 transform property to the container class with or without animation depending on the animated param.

\*/

function transformContainer(element, translate3d, animated) {

element.toggleClass('pp-easing', animated);

element.css(getTransforms(translate3d));

}

/\*\*

\* Sliding with arrow keys, both, vertical and horizontal

\*/

$(document).keydown(function (e) {

if(options.keyboardScrolling && !isMoving()){

//Moving the main page with the keyboard arrows if keyboard scrolling is enabled

switch (e.which) {

//up

case 38:

case 33:

PP.moveSectionUp();

break;

//down

case 40:

case 34:

PP.moveSectionDown();

break;

//Home

case 36:

PP.moveTo(1);

break;

//End

case 35:

PP.moveTo($('.pp-section').length);

break;

//left

case 37:

PP.moveSectionUp();

break;

//right

case 39:

PP.moveSectionDown();

break;

default:

return; // exit this handler for other keys

}

}

});

/\*\*

\* If `normalScrollElements` is used, the mouse wheel scrolling will scroll normally

\* over the defined elements in the option.

\*/

if(options.normalScrollElements){

$(document).on('mouseenter', options.normalScrollElements, function () {

PP.setMouseWheelScrolling(false);

});

$(document).on('mouseleave', options.normalScrollElements, function(){

PP.setMouseWheelScrolling(true);

});

}

/\*\*

\* Detecting mousewheel scrolling

\*

\* http://blogs.sitepointstatic.com/examples/tech/mouse-wheel/index.html

\* http://www.sitepoint.com/html5-javascript-mouse-wheel/

\*/

var prevTime = new Date().getTime();

function MouseWheelHandler(e) {

var curTime = new Date().getTime();

// cross-browser wheel delta

e = e || window.event;

var value = e.wheelDelta || -e.deltaY || -e.detail;

var delta = Math.max(-1, Math.min(1, value));

var horizontalDetection = typeof e.wheelDeltaX !== 'undefined' || typeof e.deltaX !== 'undefined';

var isScrollingVertically = (Math.abs(e.wheelDeltaX) < Math.abs(e.wheelDelta)) || (Math.abs(e.deltaX ) < Math.abs(e.deltaY) || !horizontalDetection);

//Limiting the array to 150 (lets not waste memory!)

if(scrollings.length > 149){

scrollings.shift();

}

//keeping record of the previous scrollings

scrollings.push(Math.abs(value));

//time difference between the last scroll and the current one

var timeDiff = curTime-prevTime;

prevTime = curTime;

//haven't they scrolled in a while?

//(enough to be consider a different scrolling action to scroll another section)

if(timeDiff > 200){

//emptying the array, we dont care about old scrollings for our averages

scrollings = [];

}

if(!isMoving()){

var activeSection = $('.pp-section.active');

var scrollable = isScrollable(activeSection);

var averageEnd = getAverage(scrollings, 10);

var averageMiddle = getAverage(scrollings, 70);

var isAccelerating = averageEnd >= averageMiddle;

if(isAccelerating && isScrollingVertically){

//scrolling down?

if (delta < 0) {

scrolling('down', scrollable);

//scrolling up?

}else if(delta>0){

scrolling('up', scrollable);

}

}

return false;

}

}

/\*\*

\* Gets the average of the last `number` elements of the given array.

\*/

function getAverage(elements, number){

var sum = 0;

//taking `number` elements from the end to make the average, if there are not enought, 1

var lastElements = elements.slice(Math.max(elements.length - number, 1));

for(var i = 0; i < lastElements.length; i++){

sum = sum + lastElements[i];

}

return Math.ceil(sum/number);

}

/\*\*

\* Determines the way of scrolling up or down:

\* by 'automatically' scrolling a section or by using the default and normal scrolling.

\*/

function scrolling(type, scrollable){

var check;

var scrollSection;

if(type == 'down'){

check = 'bottom';

scrollSection = PP.moveSectionDown;

}else{

check = 'top';

scrollSection = PP.moveSectionUp;

}

if(scrollable.length > 0 ){

//is the scrollbar at the start/end of the scroll?

if(isScrolled(check, scrollable)){

scrollSection();

}else{

return true;

}

}else{

//moved up/down

scrollSection();

}

}

/\*\*

\* Return a boolean depending on whether the scrollable element is at the end or at the start of the scrolling

\* depending on the given type.

\*/

function isScrolled(type, scrollable){

if(type === 'top'){

return !scrollable.scrollTop();

}else if(type === 'bottom'){

return scrollable.scrollTop() + 1 + scrollable.innerHeight() >= scrollable[0].scrollHeight;

}

}

/\*\*

\* Determines whether the active section or slide is scrollable through and scrolling bar

\*/

function isScrollable(activeSection){

return activeSection.filter('.pp-scrollable');

}

/\*\*

\* Removes the auto scrolling action fired by the mouse wheel and tackpad.

\* After this function is called, the mousewheel and trackpad movements won't scroll through sections.

\*/

function removeMouseWheelHandler(){

if (container.get(0).addEventListener) {

container.get(0).removeEventListener('mousewheel', MouseWheelHandler, false); //IE9, Chrome, Safari, Oper

container.get(0).removeEventListener('wheel', MouseWheelHandler, false); //Firefox

} else {

container.get(0).detachEvent('onmousewheel', MouseWheelHandler); //IE 6/7/8

}

}

/\*\*

\* Adds the auto scrolling action for the mouse wheel and tackpad.

\* After this function is called, the mousewheel and trackpad movements will scroll through sections

\*/

function addMouseWheelHandler(){

if (container.length) {

if (container.get(0).addEventListener) {

container.get(0).addEventListener('mousewheel', MouseWheelHandler, false); //IE9, Chrome, Safari, Oper

container.get(0).addEventListener('wheel', MouseWheelHandler, false); //Firefox

} else {

container.get(0).attachEvent('onmousewheel', MouseWheelHandler); //IE 6/7/8

}

}

}

/\*\*

\* Adds the possibility to auto scroll through sections on touch devices.

\*/

function addTouchHandler(){

if(isTouch){

//Microsoft pointers

var MSPointer = getMSPointer();

container.off('touchstart ' + MSPointer.down).on('touchstart ' + MSPointer.down, touchStartHandler);

container.off('touchmove ' + MSPointer.move).on('touchmove ' + MSPointer.move, touchMoveHandler);

}

}

/\*\*

\* Removes the auto scrolling for touch devices.

\*/

function removeTouchHandler(){

if(isTouch){

//Microsoft pointers

var MSPointer = getMSPointer();

container.off('touchstart ' + MSPointer.down);

container.off('touchmove ' + MSPointer.move);

}

}

/\*

\* Returns and object with Microsoft pointers (for IE<11 and for IE >= 11)

\* http://msdn.microsoft.com/en-us/library/ie/dn304886(v=vs.85).aspx

\*/

function getMSPointer(){

var pointer;

//IE >= 11 & rest of browsers

if(window.PointerEvent){

pointer = { down: 'pointerdown', move: 'pointermove', up: 'pointerup'};

}

//IE < 11

else{

pointer = { down: 'MSPointerDown', move: 'MSPointerMove', up: 'MSPointerUp'};

}

return pointer;

}

/\*\*

\* Gets the pageX and pageY properties depending on the browser.

\* https://github.com/alvarotrigo/fullPage.js/issues/194#issuecomment-34069854

\*/

function getEventsPage(e){

var events = new Array();

events.y = (typeof e.pageY !== 'undefined' && (e.pageY || e.pageX) ? e.pageY : e.touches[0].pageY);

events.x = (typeof e.pageX !== 'undefined' && (e.pageY || e.pageX) ? e.pageX : e.touches[0].pageX);

return events;

}

/\*\*

\* As IE >= 10 fires both touch and mouse events when using a mouse in a touchscreen

\* this way we make sure that is really a touch event what IE is detecting.

\*/

function isReallyTouch(e){

//if is not IE || IE is detecting `touch` or `pen`

return typeof e.pointerType === 'undefined' || e.pointerType != 'mouse';

}

/\*\*

\* Getting the starting possitions of the touch event

\*/

function touchStartHandler(event){

var e = event.originalEvent;

if(isReallyTouch(e)){

var touchEvents = getEventsPage(e);

touchStartY = touchEvents.y;

touchStartX = touchEvents.x;

}

}

/\* Detecting touch events

\*/

function touchMoveHandler(event){

var e = event.originalEvent;

// additional: if one of the normalScrollElements isn't within options.normalScrollElementTouchThreshold hops up the DOM chain

if ( !checkParentForNormalScrollElement(event.target) && isReallyTouch(e) ) {

var activeSection = $('.pp-section.active');

var scrollable = isScrollable(activeSection);

if(!scrollable.length){

event.preventDefault();

}

if (!isMoving()) {

var touchEvents = getEventsPage(e);

touchEndY = touchEvents.y;

touchEndX = touchEvents.x;

//$('body').append('<span style="position:fixed; top: 250px; left: 20px; z-index:88; font-size: 25px; color: #000;">touchEndY: ' + touchEndY + '</div>');

//X movement bigger than Y movement?

if (options.direction === 'horizontal' && Math.abs(touchStartX - touchEndX) > (Math.abs(touchStartY - touchEndY))) {

//is the movement greater than the minimum resistance to scroll?

if (Math.abs(touchStartX - touchEndX) > (container.width() / 100 \* options.touchSensitivity)) {

if (touchStartX > touchEndX) {

scrolling('down', scrollable);

} else if (touchEndX > touchStartX) {

scrolling('up', scrollable);

}

}

} else {

if (Math.abs(touchStartY - touchEndY) > (container.height() / 100 \* options.touchSensitivity)) {

if (touchStartY > touchEndY) {

scrolling('down', scrollable);

} else if (touchEndY > touchStartY) {

scrolling('up', scrollable);

}

}

}

}

}

}

/\*\*

\* recursive function to loop up the parent nodes to check if one of them exists in options.normalScrollElements

\* Currently works well for iOS - Android might need some testing

\* @param {Element} el target element / jquery selector (in subsequent nodes)

\* @param {int} hop current hop compared to options.normalScrollElementTouchThreshold

\* @return {boolean} true if there is a match to options.normalScrollElements

\*/

function checkParentForNormalScrollElement (el, hop) {

hop = hop || 0;

var parent = $(el).parent();

if (hop < options.normalScrollElementTouchThreshold &&

parent.is(options.normalScrollElements) ) {

return true;

} else if (hop == options.normalScrollElementTouchThreshold) {

return false;

} else {

return checkParentForNormalScrollElement(parent, ++hop);

}

}

/\*\*

\* Creates a vertical navigation bar.

\*/

function addVerticalNavigation(){

$('body').append('<div id="pp-nav"><ul></ul></div>');

var nav = $('#pp-nav');

nav.css('color', options.navigation.textColor);

nav.addClass(options.navigation.position);

for(var cont = 0; cont < $('.pp-section').length; cont++){

var link = '';

if(options.anchors.length){

link = options.anchors[cont];

}

if(options.navigation.tooltips !== 'undefined'){

var tooltip = options.navigation.tooltips[cont];

if(typeof tooltip === 'undefined'){

tooltip = '';

}

}

nav.find('ul').append('<li data-tooltip="' + tooltip + '"><a href="#' + link + '"><span></span></a></li>');

}

nav.find('span').css('border-color', options.navigation.bulletsColor);

}

/\*\*

\* Scrolls to the section when clicking the navigation bullet

\*/

$(document).on('click touchstart', '#pp-nav a', function(e){

e.preventDefault();

var index = $(this).parent().index();

scrollPage($('.pp-section').eq(index));

});

/\*\*

\* Navigation tooltips

\*/

$(document).on({

mouseenter: function(){

var tooltip = $(this).data('tooltip');

$('<div class="pp-tooltip ' + options.navigation.position +'">' + tooltip + '</div>').hide().appendTo($(this)).fadeIn(200);

},

mouseleave: function(){

$(this).find('.pp-tooltip').fadeOut(200, function() {

$(this).remove();

});

}

}, '#pp-nav li');

/\*\*

\* Activating the website navigation dots according to the given slide name.

\*/

function activateNavDots(name, sectionIndex){

if(options.navigation){

$('#pp-nav').find('.active').removeClass('active');

if(name){

$('#pp-nav').find('a[href="#' + name + '"]').addClass('active');

}else{

$('#pp-nav').find('li').eq(sectionIndex).find('a').addClass('active');

}

}

}

/\*\*

\* Activating the website main menu elements according to the given slide name.

\*/

function activateMenuElement(name){

if(options.menu){

$(options.menu).find('.active').removeClass('active');

$(options.menu).find('[data-menuanchor="'+name+'"]').addClass('active');

}

}

/\*\*

\* Checks for translate3d support

\* @return boolean

\* http://stackoverflow.com/questions/5661671/detecting-transform-translate3d-support

\*/

function support3d() {

var el = document.createElement('p'),

has3d,

transforms = {

'webkitTransform':'-webkit-transform',

'OTransform':'-o-transform',

'msTransform':'-ms-transform',

'MozTransform':'-moz-transform',

'transform':'transform'

};

// Add it to the body to get the computed style.

document.body.insertBefore(el, null);

for (var t in transforms) {

if (el.style[t] !== undefined) {

el.style[t] = 'translate3d(1px,1px,1px)';

has3d = window.getComputedStyle(el).getPropertyValue(transforms[t]);

}

}

document.body.removeChild(el);

return (has3d !== undefined && has3d.length > 0 && has3d !== 'none');

}

/\*\*

\* Gets the translate3d property to apply when using css3:true depending on the `direction` option.

\*/

function getTranslate3d(){

if (options.direction !== 'vertical') {

return 'translate3d(-100%, 0px, 0px)';

}

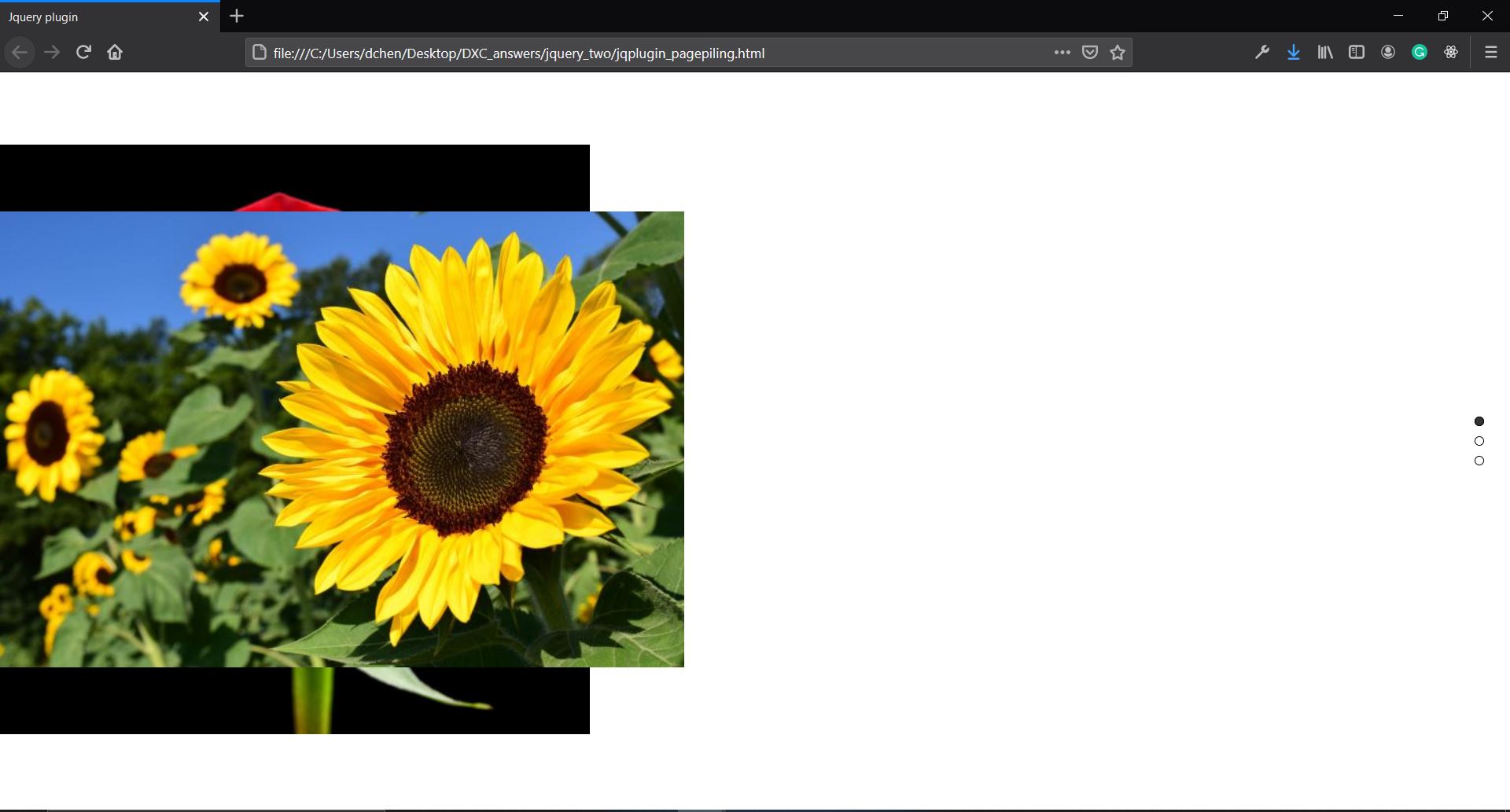
return 'translate3d(0px, -100%, 0px)';

}

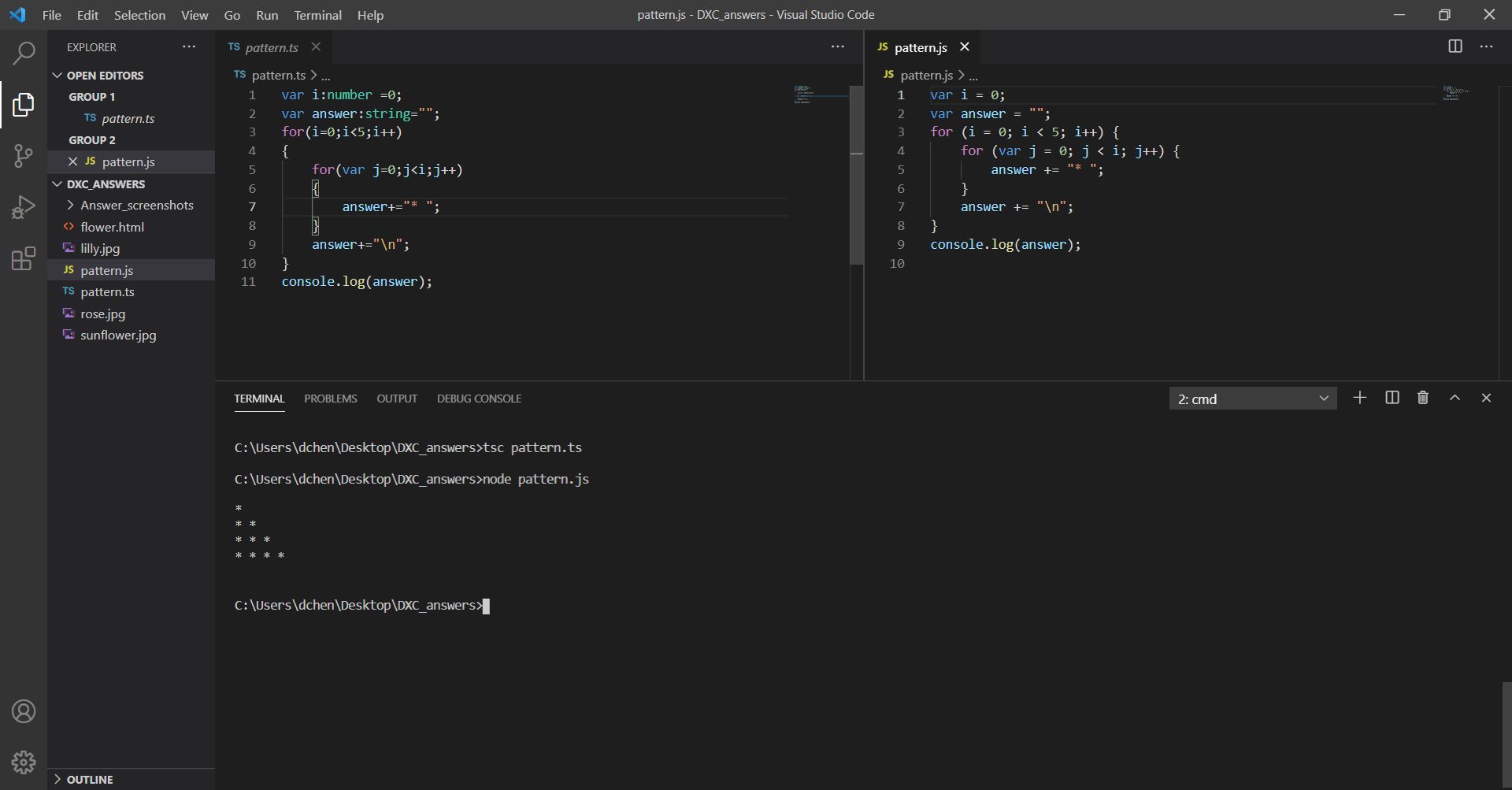
};

})(jQuery, document, window);

Screenshot:



3.



4.

eshwar-component.html

<div id="hide" \*ngIf="flag">

<h1>Eshwar's Details</h1>

<h1>My Name is {{name}}.</h1>

<h1>This is my facebook page <a href="{{link}}" target="\_blank">Link</a></h1>

<h1>This is my linkedin <a href="{{link\_linkedin}}" target="\_blank">Link</a></h1>

</div>

<button (click)="hide()">Hide</button>

<button (click)="unhide()">Unhide</button>

Eshwar-component.ts

import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app-eshwar',

templateUrl: './eshwar.component.html',

styleUrls: ['./eshwar.component.css']

})

export class EshwarComponent implements OnInit {

flag:boolean=true;

name="Eshwar Grandhe";

link ="https://www.facebook.com/eswar.anirudh";

link\_linkedin ="https://www.linkedin.com/in/eshwar-anirudh-782288178";

public hide(){

this.flag = false;

}

public unhide(){

this.flag = true;

}

constructor() { }

ngOnInit(): void {

}

}

App.component.css

<app-eshwar></app-eshwar>

5.

a)

1.To create fruit table

CREATE TABLE "FruitPrice" (

"fruit" TEXT NOT NULL,

"price" REAL NOT NULL,

"unit" TEXT NOT NULL

);

CREATE TABLE "FruitComsumption" (

"date" TEXT NOT NULL,

"fruit" TEXT NOT NULL,

"quantity" INTEGER NOT NULL

);

2.Insert into fruit table

INSERT INTO FruitPrice(fruit, price, unit)

values

("mango", 125, "kg"),

("Banana", 45, "kg"),

("guava", 85, "kg"),

("Pineapple", 65, "kg");

Insert into FruitComsumption (date,fruit,quantity)

values

("2021-apr-01", "mango", 5),

("2021-apr-02", "mango", 6),

("2021-apr-03", "mango", 3),

("2021-apr-04"," mango", 2),

("2021-apr-05", "mango", 5),

("2021-apr-01", "guava", 15),

("2021-apr-02", "guava", 12),

("2021-apr-03", "guava", 6),

("2021-apr-04", "guava", 7),

("2021-apr-05", "guava", 8),

("2021-apr-01", "Banana", 1),

("2021-apr-02", "Banana", 2),

("2021-apr-03", "Banana", 1),

("2021-apr-04", "Banana", 2),

("2021-apr-05", "Banana", 2);

CREATE VIEW day\_wise\_amount as

SELECT date, sum(quantity\*price) as total\_amount FROM

FruitComsumption, FruitPrice WHERE FruitPrice.fruit = FruitComsumption.fruit GROUP

by date;

CREATE VIEW day\_wise\_quantity as

SELECT date,sum(quantity) as total\_weight from FruitComsumption

GROUP by date

CREATE VIEW total\_fruits as

SELECT fruit,sum(quantity) as total\_comsumption FROM

FruitComsumption GROUP by fruit;