**Man page**

Do a search (Ctrl+ f) home-hero and replace url (….) by url(./web.png)

**Snow**

**Paste anywhere in index.html with all other the script codes**

<script >

var snowStorm = (function(window, document) {

this.autoStart = true;

this.excludeMobile = true;

this.flakesMax = 128;

this.flakesMaxActive = 64;

this.animationInterval = 50;

this.useGPU = true;

this.className = null;

this.excludeMobile = true;

this.flakeBottom = null;

this.followMouse = false;

this.snowColor = '#fff';

this.snowCharacter = '&bull;';

this.snowStick = true;

this.targetElement = null;

this.useMeltEffect = true;

this.useTwinkleEffect = false;

this.usePositionFixed = false;

this.usePixelPosition = false;

this.freezeOnBlur = true;

this.flakeLeftOffset = 0;

this.flakeRightOffset = 0;

this.flakeWidth = 8;

this.flakeHeight = 8;

this.vMaxX = 5;

this.vMaxY = 4;

this.zIndex = 0;

var storm = this,

features,

isIE = navigator.userAgent.match(/msie/i),

isIE6 = navigator.userAgent.match(/msie 6/i),

isMobile = navigator.userAgent.match(/mobile|opera m(ob|in)/i),

isBackCompatIE = (isIE && document.compatMode === 'BackCompat'),

noFixed = (isBackCompatIE || isIE6),

screenX = null, screenX2 = null, screenY = null, scrollY = null, docHeight = null, vRndX = null, vRndY = null,

windOffset = 1,

windMultiplier = 2,

flakeTypes = 6,

fixedForEverything = false,

targetElementIsRelative = false,

opacitySupported = (function(){

try {

document.createElement('div').style.opacity = '0.5';

} catch(e) {

return false;

}

return true;

}()),

didInit = false,

docFrag = document.createDocumentFragment();

features = (function() {

var getAnimationFrame;

function timeoutShim(callback) {

window.setTimeout(callback, 1000/(storm.animationInterval || 20));

}

var \_animationFrame = (window.requestAnimationFrame ||

window.webkitRequestAnimationFrame ||

window.mozRequestAnimationFrame ||

window.oRequestAnimationFrame ||

window.msRequestAnimationFrame ||

timeoutShim);

getAnimationFrame = \_animationFrame ? function() {

return \_animationFrame.apply(window, arguments);

} : null;

var testDiv;

testDiv = document.createElement('div');

function has(prop) {

var result = testDiv.style[prop];

return (result !== undefined ? prop : null);

}

var localFeatures = {

transform: {

ie: has('-ms-transform'),

moz: has('MozTransform'),

opera: has('OTransform'),

webkit: has('webkitTransform'),

w3: has('transform'),

prop: null

},

getAnimationFrame: getAnimationFrame

};

localFeatures.transform.prop = (

localFeatures.transform.w3 ||

localFeatures.transform.moz ||

localFeatures.transform.webkit ||

localFeatures.transform.ie ||

localFeatures.transform.opera

);

testDiv = null;

return localFeatures;

}());

this.timer = null;

this.flakes = [];

this.disabled = false;

this.active = false;

this.meltFrameCount = 20;

this.meltFrames = [];

this.setXY = function(o, x, y) {

if (!o) {

return false;

}

if (storm.usePixelPosition || targetElementIsRelative) {

o.style.left = (x - storm.flakeWidth) + 'px';

o.style.top = (y - storm.flakeHeight) + 'px';

} else if (noFixed) {

o.style.right = (100-(x/screenX\*100)) + '%';

o.style.top = (Math.min(y, docHeight-storm.flakeHeight)) + 'px';

} else {

if (!storm.flakeBottom) {

// if not using a fixed bottom coordinate...

o.style.right = (100-(x/screenX\*100)) + '%';

o.style.bottom = (100-(y/screenY\*100)) + '%';

} else {

// absolute top.

o.style.right = (100-(x/screenX\*100)) + '%';

o.style.top = (Math.min(y, docHeight-storm.flakeHeight)) + 'px';

}

}

};

this.events = (function() {

var old = (!window.addEventListener && window.attachEvent), slice = Array.prototype.slice,

evt = {

add: (old?'attachEvent':'addEventListener'),

remove: (old?'detachEvent':'removeEventListener')

};

function getArgs(oArgs) {

var args = slice.call(oArgs), len = args.length;

if (old) {

args[1] = 'on' + args[1];

if (len > 3) {

args.pop();

}

} else if (len === 3) {

args.push(false);

}

return args;

}

function apply(args, sType) {

var element = args.shift(),

method = [evt[sType]];

if (old) {

element[method](args[0], args[1]);

} else {

element[method].apply(element, args);

}

}

function addEvent() {

apply(getArgs(arguments), 'add');

}

function removeEvent() {

apply(getArgs(arguments), 'remove');

}

return {

add: addEvent,

remove: removeEvent

};

}());

function rnd(n,min) {

if (isNaN(min)) {

min = 0;

}

return (Math.random()\*n)+min;

}

function plusMinus(n) {

return (parseInt(rnd(2),10)===1?n\*-1:n);

}

this.randomizeWind = function() {

var i;

vRndX = plusMinus(rnd(storm.vMaxX,0.2));

vRndY = rnd(storm.vMaxY,0.2);

if (this.flakes) {

for (i=0; i<this.flakes.length; i++) {

if (this.flakes[i].active) {

this.flakes[i].setVelocities();

}

}

}

};

this.scrollHandler = function() {

var i;

scrollY = (storm.flakeBottom ? 0 : parseInt(window.scrollY || document.documentElement.scrollTop || (noFixed ? document.body.scrollTop : 0), 10));

if (isNaN(scrollY)) {

scrollY = 0;

}

if (!fixedForEverything && !storm.flakeBottom && storm.flakes) {

for (i=0; i<storm.flakes.length; i++) {

if (storm.flakes[i].active === 0) {

storm.flakes[i].stick();

}

}

}

};

this.resizeHandler = function() {

if (window.innerWidth || window.innerHeight) {

screenX = window.innerWidth - 16 - storm.flakeRightOffset;

screenY = (storm.flakeBottom || window.innerHeight);

} else {

screenX = (document.documentElement.clientWidth || document.body.clientWidth || document.body.scrollWidth) - (!isIE ? 8 : 0) - storm.flakeRightOffset;

screenY = storm.flakeBottom || document.documentElement.clientHeight || document.body.clientHeight || document.body.scrollHeight;

}

docHeight = document.body.offsetHeight;

screenX2 = parseInt(screenX/2,10);

};

this.resizeHandlerAlt = function() {

screenX = storm.targetElement.offsetWidth - storm.flakeRightOffset;

screenY = storm.flakeBottom || storm.targetElement.offsetHeight;

screenX2 = parseInt(screenX/2,10);

docHeight = document.body.offsetHeight;

};

this.freeze = function() {

if (!storm.disabled) {

storm.disabled = 1;

} else {

return false;

}

storm.timer = null;

};

this.resume = function() {

if (storm.disabled) {

storm.disabled = 0;

} else {

return false;

}

storm.timerInit();

};

this.toggleSnow = function() {

if (!storm.flakes.length) {

storm.start();

} else {

storm.active = !storm.active;

if (storm.active) {

storm.show();

storm.resume();

} else {

storm.stop();

storm.freeze();

}

}

};

this.stop = function() {

var i;

this.freeze();

for (i=0; i<this.flakes.length; i++) {

this.flakes[i].o.style.display = 'none';

}

storm.events.remove(window,'scroll',storm.scrollHandler);

storm.events.remove(window,'resize',storm.resizeHandler);

if (storm.freezeOnBlur) {

if (isIE) {

storm.events.remove(document,'focusout',storm.freeze);

storm.events.remove(document,'focusin',storm.resume);

} else {

storm.events.remove(window,'blur',storm.freeze);

storm.events.remove(window,'focus',storm.resume);

}

}

};

this.show = function() {

var i;

for (i=0; i<this.flakes.length; i++) {

this.flakes[i].o.style.display = 'block';

}

};

this.SnowFlake = function(type,x,y) {

var s = this;

this.type = type;

this.x = x||parseInt(rnd(screenX-20),10);

this.y = (!isNaN(y)?y:-rnd(screenY)-12);

this.vX = null;

this.vY = null;

this.vAmpTypes = [1,1.2,1.4,1.6,1.8];

this.vAmp = this.vAmpTypes[this.type] || 1;

this.melting = false;

this.meltFrameCount = storm.meltFrameCount;

this.meltFrames = storm.meltFrames;

this.meltFrame = 0;

this.twinkleFrame = 0;

this.active = 1;

this.fontSize = (10+(this.type/5)\*10);

this.o = document.createElement('div');

this.o.innerHTML = storm.snowCharacter;

if (storm.className) {

this.o.setAttribute('class', storm.className);

}

this.o.style.color = storm.snowColor;

this.o.style.position = (fixedForEverything?'fixed':'absolute');

if (storm.useGPU && features.transform.prop) {

this.o.style[features.transform.prop] = 'translate3d(0px, 0px, 0px)';

}

this.o.style.width = storm.flakeWidth+'px';

this.o.style.height = storm.flakeHeight+'px';

this.o.style.fontFamily = 'arial,verdana';

this.o.style.cursor = 'default';

this.o.style.overflow = 'hidden';

this.o.style.fontWeight = 'normal';

this.o.style.zIndex = storm.zIndex;

docFrag.appendChild(this.o);

this.refresh = function() {

if (isNaN(s.x) || isNaN(s.y)) {

return false;

}

storm.setXY(s.o, s.x, s.y);

};

this.stick = function() {

if (noFixed || (storm.targetElement !== document.documentElement && storm.targetElement !== document.body)) {

s.o.style.top = (screenY+scrollY-storm.flakeHeight)+'px';

} else if (storm.flakeBottom) {

s.o.style.top = storm.flakeBottom+'px';

} else {

s.o.style.display = 'none';

s.o.style.bottom = '0%';

s.o.style.position = 'fixed';

s.o.style.display = 'block';

}

};

this.vCheck = function() {

if (s.vX>=0 && s.vX<0.2) {

s.vX = 0.2;

} else if (s.vX<0 && s.vX>-0.2) {

s.vX = -0.2;

}

if (s.vY>=0 && s.vY<0.2) {

s.vY = 0.2;

}

};

this.move = function() {

var vX = s.vX\*windOffset, yDiff;

s.x += vX;

s.y += (s.vY\*s.vAmp);

if (s.x >= screenX || screenX-s.x < storm.flakeWidth) { // X-axis scroll check

s.x = 0;

} else if (vX < 0 && s.x-storm.flakeLeftOffset < -storm.flakeWidth) {

s.x = screenX-storm.flakeWidth-1; // flakeWidth;

}

s.refresh();

yDiff = screenY+scrollY-s.y+storm.flakeHeight;

if (yDiff<storm.flakeHeight) {

s.active = 0;

if (storm.snowStick) {

s.stick();

} else {

s.recycle();

}

} else {

if (storm.useMeltEffect && s.active && s.type < 3 && !s.melting && Math.random()>0.998) {

s.melting = true;

s.melt();

}

if (storm.useTwinkleEffect) {

if (s.twinkleFrame < 0) {

if (Math.random() > 0.97) {

s.twinkleFrame = parseInt(Math.random() \* 8, 10);

}

} else {

s.twinkleFrame--;

if (!opacitySupported) {

s.o.style.visibility = (s.twinkleFrame && s.twinkleFrame % 2 === 0 ? 'hidden' : 'visible');

} else {

s.o.style.opacity = (s.twinkleFrame && s.twinkleFrame % 2 === 0 ? 0 : 1);

}

}

}

}

};

this.animate = function() {

s.move();

};

this.setVelocities = function() {

s.vX = vRndX+rnd(storm.vMaxX\*0.12,0.1);

s.vY = vRndY+rnd(storm.vMaxY\*0.12,0.1);

};

this.setOpacity = function(o,opacity) {

if (!opacitySupported) {

return false;

}

o.style.opacity = opacity;

};

this.melt = function() {

if (!storm.useMeltEffect || !s.melting) {

s.recycle();

} else {

if (s.meltFrame < s.meltFrameCount) {

s.setOpacity(s.o,s.meltFrames[s.meltFrame]);

s.o.style.fontSize = s.fontSize-(s.fontSize\*(s.meltFrame/s.meltFrameCount))+'px';

s.o.style.lineHeight = storm.flakeHeight+2+(storm.flakeHeight\*0.75\*(s.meltFrame/s.meltFrameCount))+'px';

s.meltFrame++;

} else {

s.recycle();

}

}

};

this.recycle = function() {

s.o.style.display = 'none';

s.o.style.position = (fixedForEverything?'fixed':'absolute');

s.o.style.bottom = 'auto';

s.setVelocities();

s.vCheck();

s.meltFrame = 0;

s.melting = false;

s.setOpacity(s.o,1);

s.o.style.padding = '0px';

s.o.style.margin = '0px';

s.o.style.fontSize = s.fontSize+'px';

s.o.style.lineHeight = (storm.flakeHeight+2)+'px';

s.o.style.textAlign = 'center';

s.o.style.verticalAlign = 'baseline';

s.x = parseInt(rnd(screenX-storm.flakeWidth-20),10);

s.y = parseInt(rnd(screenY)\*-1,10)-storm.flakeHeight;

s.refresh();

s.o.style.display = 'block';

s.active = 1;

};

this.recycle();

this.refresh();

};

this.snow = function() {

var active = 0, flake = null, i, j;

for (i=0, j=storm.flakes.length; i<j; i++) {

if (storm.flakes[i].active === 1) {

storm.flakes[i].move();

active++;

}

if (storm.flakes[i].melting) {

storm.flakes[i].melt();

}

}

if (active<storm.flakesMaxActive) {

flake = storm.flakes[parseInt(rnd(storm.flakes.length),10)];

if (flake.active === 0) {

flake.melting = true;

}

}

if (storm.timer) {

features.getAnimationFrame(storm.snow);

}

};

this.mouseMove = function(e) {

if (!storm.followMouse) {

return true;

}

var x = parseInt(e.clientX,10);

if (x<screenX2) {

windOffset = -windMultiplier+(x/screenX2\*windMultiplier);

} else {

x -= screenX2;

windOffset = (x/screenX2)\*windMultiplier;

}

};

this.createSnow = function(limit,allowInactive) {

var i;

for (i=0; i<limit; i++) {

storm.flakes[storm.flakes.length] = new storm.SnowFlake(parseInt(rnd(flakeTypes),10));

if (allowInactive || i>storm.flakesMaxActive) {

storm.flakes[storm.flakes.length-1].active = -1;

}

}

storm.targetElement.appendChild(docFrag);

};

this.timerInit = function() {

storm.timer = true;

storm.snow();

};

this.init = function() {

var i;

for (i=0; i<storm.meltFrameCount; i++) {

storm.meltFrames.push(1-(i/storm.meltFrameCount));

}

storm.randomizeWind();

storm.createSnow(storm.flakesMax); // create initial batch

storm.events.add(window,'resize',storm.resizeHandler);

storm.events.add(window,'scroll',storm.scrollHandler);

if (storm.freezeOnBlur) {

if (isIE) {

storm.events.add(document,'focusout',storm.freeze);

storm.events.add(document,'focusin',storm.resume);

} else {

storm.events.add(window,'blur',storm.freeze);

storm.events.add(window,'focus',storm.resume);

}

}

storm.resizeHandler();

storm.scrollHandler();

if (storm.followMouse) {

storm.events.add(isIE?document:window,'mousemove',storm.mouseMove);

}

storm.animationInterval = Math.max(20,storm.animationInterval);

storm.timerInit();

};

this.start = function(bFromOnLoad) {

if (!didInit) {

didInit = true;

} else if (bFromOnLoad) {

return true;

}

if (typeof storm.targetElement === 'string') {

var targetID = storm.targetElement;

storm.targetElement = document.getElementById(targetID);

if (!storm.targetElement) {

throw new Error('Snowstorm: Unable to get targetElement "'+targetID+'"');

}

}

if (!storm.targetElement) {

storm.targetElement = (document.body || document.documentElement);

}

if (storm.targetElement !== document.documentElement && storm.targetElement !== document.body) {

storm.resizeHandler = storm.resizeHandlerAlt;

storm.usePixelPosition = true;

}

storm.resizeHandler();

storm.usePositionFixed = (storm.usePositionFixed && !noFixed && !storm.flakeBottom);

if (window.getComputedStyle) {

try {

targetElementIsRelative = (window.getComputedStyle(storm.targetElement, null).getPropertyValue('position') === 'relative');

} catch(e) {

targetElementIsRelative = false;

}

}

fixedForEverything = storm.usePositionFixed;

if (screenX && screenY && !storm.disabled) {

storm.init();

storm.active = true;

}

};

function doDelayedStart() {

window.setTimeout(function() {

storm.start(true);

}, 20);

storm.events.remove(isIE?document:window,'mousemove',doDelayedStart);

}

function doStart() {

if (!storm.excludeMobile || !isMobile) {

doDelayedStart();

}

storm.events.remove(window, 'load', doStart);

}

if (storm.autoStart) {

storm.events.add(window, 'load', doStart, false);

}

return this;

}(window, document));

</script>

**Social Icons**

Do a search (Ctrl+ f) social-icons and

Twiter

replace <img src =…true”/> by <img src="Gino.png" class="preview-image1" aria-hidden="true"

LinkedIn

replace <img src =…true”/> by <img src="Josuah." class="preview-image2" aria-hidden="true"

Facebook

replace <img src =…true”/> by <img src="Oumar." class="preview-image3" aria-hidden="true"

Youtube

replace <img src =…true”/> by <img src="Dione." class="preview-image4" aria-hidden="true"

**Morph Style**

.preview-image1 {

-webkit-transition: all 0.5s ease;

-moz-transition: all 0.5s ease;

-o-transition: all 0.5s ease;

-ms-transition: all 0.5s ease;

transition: all 0.5s ease;

}

.preview-image1:hover {

border-radius: 50%;

-webkit-transform: rotate(360deg);

-moz-transform: rotate(360deg);

-o-transform: rotate(360deg);

-ms-transform: rotate(360deg);

transform: rotate(360deg);

}

.preview-image2 {

-webkit-transition: all 0.5s ease;

-moz-transition: all 0.5s ease;

-o-transition: all 0.5s ease;

-ms-transition: all 0.5s ease;

transition: all 0.5s ease;

}

.preview-image2:hover {

border-radius: 50%;

-webkit-transform: rotate(360deg);

-moz-transform: rotate(360deg);

-o-transform: rotate(360deg);

-ms-transform: rotate(360deg);

transform: rotate(360deg);

}

.preview-image3 {

-webkit-transition: all 0.5s ease;

-moz-transition: all 0.5s ease;

-o-transition: all 0.5s ease;

-ms-transition: all 0.5s ease;

transition: all 0.5s ease;

}

.preview-image3:hover {

border-radius: 50%;

-webkit-transform: rotate(360deg);

-moz-transform: rotate(360deg);

-o-transform: rotate(360deg);

-ms-transform: rotate(360deg);

transform: rotate(360deg);

}

.preview-image4 {

-webkit-transition: all 0.5s ease;

-moz-transition: all 0.5s ease;

-o-transition: all 0.5s ease;

-ms-transition: all 0.5s ease;

transition: all 0.5s ease;

}

.preview-image4:hover {

border-radius: 50%;

-webkit-transform: rotate(360deg);

-moz-transform: rotate(360deg);

-o-transform: rotate(360deg);

-ms-transform: rotate(360deg);

transform: rotate(360deg);

}