Scatter v0.1 Documentation

Scatter is a JavaScript library for randomly arranging HTML elements within a containing element.

See the file **readme.md** for project information, requirements, licensing and credits.

This documentation aims to provide everything you need to know to use Scatter - please raise an issue if you find any bugs/errors/omissions, or have any questions.

Hyperlinks that are internal to the PDF are blue and dashed, those to external websites are red and underlined.

Scatter Documentation

Scatter is a JavaScript library for randomly arranging HTML elements within a containing element.

The main use case is an image gallery, but the library purposefully does not restrict what it can be used for - it can provide a scattered polaroid photo effect as easily as using decorative images to provide a randomized background, or any other reasons for scattering items your imagination can come up with.

Restrictions on markup and styling are intentionally kept to a minimum - the elements must be top-level children of the container, and the container's position style needs to be relative, absolute, or fixed (not static).

Usage

A Scatter instance is created by loading the JavaScript library, then calling `new Scatter(target)`, where target is the container of the elements you wish to scatter:

```
<div id="container">...</div>
<script src="scatter.js"></script>
<script>
 new Scatter('#container');
</script>
```

Scatter doesn't care where the src script tag is placed (e.g. in head, at end of body, or elsewhere), and likewise, the new Scatter call can be placed in the page, in a document ready function, in an event handler, or anywhere else that makes sense.

The instance can be assigned to a variable, allowing methods to be called - such as to first scatter and then select the first item:

```
<div id="container">...</div>
<script src="scatter.js"></script>
<script>
 var MyScatter = new Scatter('#container');
 MyScatter.select(0);
</script>
```

To change the default behaviour, use the second argument to pass in an object containing options, for example:

```
<div id="container">...</div>
<script src="scatter.js"></script>
<script>
 new Scatter('#container',{Mode:'pile'});
</script>
```

The "demos/" directory contains various working examples showing different ways Scatter can be used.

Methods

new Scatter (Target , Options)

Creates a new Scatter instance.

Target can be either:

- a CSS selector identifying a single element
- a single HTML Node (i.e. result of document.getElementByld or similar functions)

Options is an object, see Options page.

configure (Options)

Change Options after an instance has been created.

arrange (Mode , Reset=true)

Arranges items into their chosen positions based on Mode.

select (Index)

Selects the specified item - moves to center, increases z-index, and applies scaling and class as per options.

Index is one of three things:

- The numeric index of the Element within the container
- a HTML Node of an Element
- An Event with currentTarget pointing at an Element.

When an existing item is selected, discard is called automatically.

discard ()

De-selects the specified item, returns to original position and removes class.

prev ()

Selects the previous item.

If called when first item selected, loops around to last item.

next ()

Selects the next item

If called when last item selected, loops around to first item.

Options

This page describes the options which can be set in the second argument when creating a new Scatter instance.

Mode

Controls how items are arranged.

- center - items are placed in center of container, no rotation.
- stack - same as center, but with rotation applied.
- pile - same as stack, but each item is slightly moved.
- radial - items are spread a random distance from the container center.
- random - simple random positioning.
- grid - items are organised in rows and columns.
- fixed - items are placed as per specified positions.

Default *radial*

InitialMode

Mode that applies initially, until arrange is called.

Default *unset* (i.e. inherits Mode value)

InitialSelect

If set, select is immediately called on the item at specified index.

That is, this:

```
ScatterObj = new Scatter(Container,{InitialSelect:0})
```

is equivalent to this:

```
ScatterObj = new Scatter(Container,{});
ScatterObj.select(0)
```

Default *unset*

SelectedClass

Class to add to the item when selected (and remove when discarded).

Default "*selected*"

SelectedScale

A selected item is scaled to this value.

Default *1.5*

Scale

All unselected items are scaled to this value.

Default *1.0*

MaxRotation

How much each item can be rotated.

This value is a magnitude, so range of rotation is twice (from negative to positive).

Default *10*

Shuffle

Whether item ordering should be shuffled.

Default *true* for random,radial; *false* for all other modes.

ContainerEvents

An object indicating functions to bind to specific events on the container.

The built-in methods must be defined as strings, but for all other functions, pass in the function itself.

Default value is `{click:"arrange"}` - i.e. clicking the container arranges items.

To disable default behaviour without specifying a new function, use `{click:void 0}`

ChildEvents

An object indicating functions to bind to specific events on the children of the container.

The built-in methods must be defined as strings, but for all other functions, pass in the function itself.

Default is `{click:"select"}` - i.e. clicking a container child selects that item.

To disable default behaviour without specifying a new function, use `{click:void 0}`

Mode-specific options

The options listed below only apply when arranging items by the mode the option name starts with (i.e. grid, pile, radial, fixed).

GridAlignCenter

Boolean, specifies whether items in grid mode should be centered (true), or in top left (false).

Default *true*

GridSpacing

Integer, how many pixels of space should seperate items in grid mode.

Default *1*

GridSpacingX

Override GridSpacing horizontally.

Default *unset*

GridSpacingY

Override GridSpacing vertically.

Default *unset*

PileOffsetFactor

Numeric. The width of an item is divided by this number to determine maximum random offset.

Default *3*

RadialWidthFactor

Numeric. The random horizontal offset for radial mode is multiplied by this number.

Default *0.85*

RadialHeightFactor

Numeric. The random vertical offset for radial mode is multiplied by this number.

Default *1.2*

RadialMinDistance

Numeric. The smallest random offset allowed - prevents too much clustering in the center.

Default *0.5*

FixedRelativeTo

Whether fixed mode positions are relative to the top left *corner*, or *center* of the container.

Default *center*

FixedPositions

Array of item positions; accepts arrays or objects (but not both in same array).

[x,y], *[x,y,angle]*, *{left:x,top:y}*, *{left:x,top:y,angle:a}*

Default: *unset*