This repository Search Pull requests Issues Gist

rajgoel / reveal.js-plugins

Watch 15 Star 85 Fork 48

Code Issues 11 Pull requests 1 Projects 0 Wiki Pulse Graphs

Branch: master reveal.js-plugins / anything / Create new file Upload files Find file History

rajgoel committed on GitHub Update README.md Latest commit 50d1547 9 days ago

..

README.md Update README.md 9 days ago

anything.js Support functions in JSON string a year ago

README.md

Anything

A plugin for Reveal.js allowing to add anything inside an HTML object using a JSON string and a javascript function. The plugin allows you to define a class for which the content of all HTML object of this class will be modified by a given javascript function. Inside the HTML object you can provide a comment containing a JSON string that will be used by function to customise the content.

Check out the live demo

Installation

Copy the files `anything.js` into the plugin folder of your reveal.js presentation, i.e. `plugin/anything`.

Add the plugins to the dependencies in your presentation, as below.

```
Reveal.initialize({
        // ...
        dependencies: [
                // ...
                { src: 'plugin/anything/anything.js' },
                // ...
        ]
});
```

Configuration & basic usage

The plugin can be configured by providing an `anything` option containing an array of `className`, `defaults`, and `f` within the reveal.js initialization options.

```
Reveal.initialize({
        // ...
        anything: [
         {
          className: "random",
          defaults: {min: 0, max: 9},
          initialize: (function(container, options){
```

```
            container.innerHTML = Math.trunc( options.min + Math.random()*(options.max-options.min + 1) );
            })
        },
        // ...
    ],
```

With the above configuration the plugin searches for all HTML object with class `random`. For each of the HTML objects it checks whether there is a JSON string within a comment inside the HTML object. Then, it calls the function `function(container, options)` where `container` is the HTML object and `options` is the JSON string. It is possible to specify the `defaults` parameter to be used if no JSON string is provided or not all values required by the function are given in the JSON string.

The code

```
<p>
        Today's winning 3 digit number is :
        <span class="random"></span>,
        <span class="random"></span>,
        <span class="random"></span>.
</p>
```

produces the output

```
<p>
        Today's winning 3 digit number is :
        <span class="random">3</span>,
        <span class="random">8</span>,
        <span class="random">0</span>.
</p>
```

The code

```
<p>
        Today's roll of a die is:
        <span class="random"><!-- { "min": 1, "max": 6 } --></span>.
</p>
```

produces the output

```
<p>
        Today's roll of a die is:
        <span class="random">4</span>.
</p>
```

Advanced usage

The plugin can be used to easily integrate external javascript libraries.

Charts.js

With the plugin charts created by Chart.js v2.0 can easily be included in the slides.

```
Reveal.initialize({
        // ...
        anything: [
         {
```

```
                className: "chart",
                initialize: (function(container, options){ container.chart = new Chart(container.getContext("2d"), op
         },
         // ...
        ],
        dependencies: [
                // ...
                { src: 'Chart.min.js' },
                { src: 'plugin/anything/anything.js' },
                // ...
        ]
});
```

A chart can be included in a slide by adding a `canvas` element and a JSON string specifying the chart options.

```
<canvas class="chart stretch">
<!--
{
 "type": "line",
 "data": {
  "labels": ["January"," February"," March"," April"," May"," June"," July"],
  "datasets":[
   {
    "data":[" 65"," 59"," 80"," 81"," 56"," 55"," 40"],
    "label":"My first dataset","backgroundColor":"rgba(20,220,220,.8)"
   },
   {
    "data":[" 28"," 48"," 40"," 19"," 86"," 27"," 90"],
    "label":"My second dataset","backgroundColor":"rgba(220,120,120,.8)"
   }
  ]
 },
 "options": { "responsive": "true" }
}
-->
</canvas>
```

Note, that the Chart plugin provides an easier way to use Chart.js.

Function-plot.js

With the plugin plots of functions created by Function-plot.js can easily be included in the slides.

```
Reveal.initialize({
        // ...
        anything: [
         {
                className: "plot",
                defaults: {width:500, height: 500, grid:true},
                initialize: (function(container, options){ options.target = "#"+container.id; functionPlot(options) }
         },
         // ...
        ],
        dependencies: [
                // ...
                { src: 'reveal.js-plugins/function-plot/site/js/vendor/jquery-1.11.2.min.js' },
                { src: 'reveal.js-plugins/function-plot/site/js/vendor/d3.js' },
                { src: 'reveal.js-plugins/function-plot/site/js/function-plot.js' },
                { src: 'plugin/anything/anything.js' },
                // ...
        ]
});
```

A plot can be included in a slide by adding a `div` element and a JSON string specifying the options.

```
<div class="plot" id="myplot1" style="background-color:#fff; width:800px; height:400px; margin: 0 auto;">
<!--
{
 "target":"#myplot1",
 "height":400,
 "width":"800",
 "xAxis":{"domain":[-10,10]},
 "yAxis":{"domain":[-5,5]},
 "grid":true,
 "data":[{"fn":"sin(x)","color":"darkred"}]
}
-->
</div>
```

With the above `defaults`, the input can be eased, e.g.

```
<div class="plot" id="myplot2" style="background-color:#fff; width:500px; height:500px; margin: 0 auto;">
<!--
 {
  "xAxis": {"domain": ["-10", "10"]},
  "yAxis": {"domain": ["0", "10"]},
  "data": [{ "fn": "10 -x * x/10" }]
}
-->
</div>
```

More advanced usage

The plugin allows to define functions within the JSON options.

Example

In the following example, the function `options.initialize(container)` is called for each element of the class `anything`. The function is defined within the JSON string.

```
Reveal.initialize({
        // ...
        anything: [
         {
                className: "anything",
                initialize: (function(container, options){ if (options && options.initialize) { options.initialize(cc
         },
         // ...
        ],
        dependencies: [
                // ...
                { src: 'reveal.js-plugins/anything/d3/d3.v3.min.js' },
                { src: 'reveal.js-plugins/anything/d3/topojson.v1.min.js' },
                { src: 'plugin/anything/anything.js' },
                // ...
        ]
});
```

The d3.js library can now be used to draw a globe within a canvas element.

```
<canvas width=500 height=500 class="anything">
<!--
{
```

```
  "initialize": "function(container) {
        var width = container.width,
            height = container.height;
        var radius = height / 2 - 5,
            scale = radius,
            velocity = .02;
        var projection = d3.geo.orthographic()
            .translate([width / 2, height / 2])
            .scale(scale)
            .clipAngle(90);
        var context = container.getContext('2d');
        var path = d3.geo.path()
            .projection(projection)
            .context(context);

        d3.json('reveal.js-plugins/anything/d3/world-110m.json', function(error, world) {
          if (error) throw error;
          var land = topojson.feature(world, world.objects.land);
          d3.timer(function(elapsed) {
            context.clearRect(0, 0, width, height);
            projection.rotate([velocity * elapsed, 0]);
            context.beginPath();
            path(land);
            context.fillStyle = '#fff';
            context.fill();
            context.beginPath();
            context.arc(width / 2, height / 2, radius, 0, 2 * Math.PI, true);
            context.lineWidth = 2.5;
            context.strokeStyle = '#fff';
            context.stroke();
          });
        });
        d3.select(self.frameElement).style('height', height + 'px');

    }"
}
-->
</canvas>
```

License

MIT licensed

Copyright (C) 2016 Asvin Goel

© 2017 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About