Mini border radius

kounelios13

Published
with GitBook

Table of Contents

Introduction

Mini border radius generator

In this manual you will find anything you need to know about this javascript plugin

What is mini border radius

What is mini border radius?

Mini border radius is a small javascript plugin written in Jquery that allows you to add one or more border radius generator in a web page or in a web app and it is free .

How do I install it

How do I install it

Installation is not that hard.All you have to do is to include the following tags in your html file:

1. <script src="path/to/miniBorderRadius.js"></script>
2. <link rel="stylesheet" href="path/to/miniBorderRadius.css">

Dependencies

Dependencies

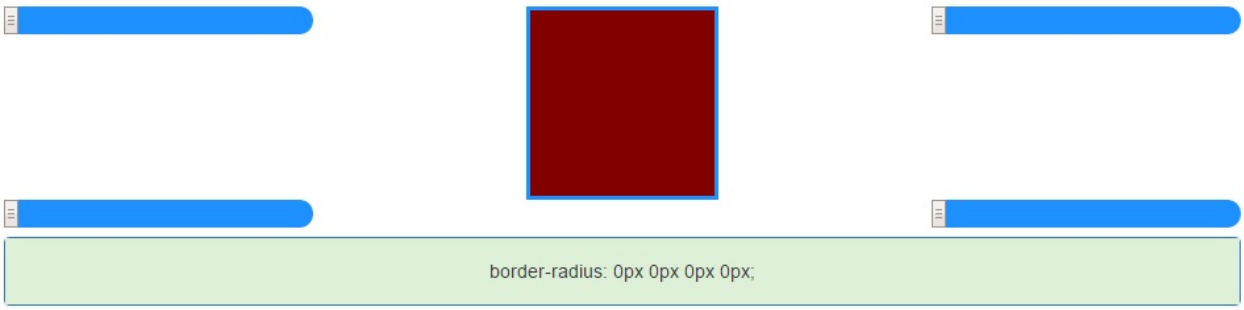
Since this plugin is written in jQuery you need to have jQuery in your web page.Also this plugin some bootstrap css components you need to also have bootstrap in your webpage

So now what?

So now what?

So you installed mini border radius.Right?Well there are a few things left to do.Each generator needs its own div to live on.So to create a new generator you have to do it like this:

`var generator=new Generator("#sample");` This is how a default generator looks like:



Instead of using the id of the host-parent div you can pass a list of 5 elements in the constructor like this:

```
generator=new
Generator([host_div_id,height,width,max_value,bg_color]);
```

Now the first parameter of the list is the id of the host-parent div.The next 2 parameters determine the size of the generator in the page.The fourth value determines the max value for the sliders inside the generator and the last one determines the background color of the generator.There is also a third constructor(3rd way to create a generator) which takes 2 arguments.The first one can be the id of the host-parent div or it can be a list like the previous list.The second one parameter is the id of an existing object in your page.What is this object?This object is going to replace the default object that the generator uses to apply the border-radius property.

Example: `generator=new Generator([host_div_id,height,width,max_value,bg_color],"#image");`

Let me tell you what the above code does.It look for an html element with the id "#image".It removes it from where it is and it replaces the default object used by the generator with this element.

Wait.Nothing happened.

Wait.Nothing happened.

So you have followed all the instructions up to here only to see that the generator is visible in your page but it is not working.This is absolutely normal.Creating a generator does not mean that is active.To activate generator you have to write the following code:

```
generator.activateGenerator();
```

After this you are ready to go

Generator methods

Generator methods

Each generator has some methods:

.activateGenerator()

This is the main function that activates the generator.Chainable:Yes.

.deactivateGenerator()

This function resets and deactivates the generator so before executing that method be sure you have saved somewhere the code produced by the generator.Chainable:Yes.

.reset()

Resets any slider inside a generator and set the border radius property of the object used by the generator to 0.Chainable:Yes

.replaceObject(object,restrict_size)

Replace the default object where border radius is applied with another one.If you want the new object to have the same size as the previous one pass true as the second parameter in the function. .Chainable:Yes

.setMax(value)

Set the max value for all the sliders inside a generator. Chainable:Yes

.setStep(value)

Set the curent step for each slider inside a generator.Chainable:Yes

.setSize(height,width)

Sets the size of a generator.Chainable:Yes

.setBackground(color,true)

Sets the background color of a generator.When true is passed as second argument a random color will be selected.Chainable:Yes

.init(options)

This function accepts a list of 4 default

options:height,width,max_value,default_background.Chainable:Yes

getBackup()

Return the elements that were removed when the generator was created.Chainable:No

restoreBackup(destination)

Restore the backup to the specified destination.Chainable:Yes

.getCode()

Gets the code generated from a miniRadiusGenerator.Chainable:No

.getId()

Returns the id of the host div.Chainable:No

.addToFavourites()

Adds the curent code produced by the generator into a list.Chainable:Yes

.getFavourites()

Returns favourites list.Chainable:No

.removeFavourites()

Removes favourites.Chainable:Yes

.toggleFavourites()

Toggle the list of favourites created by the current generator.Chainable:Yes

.downloadFavourites()

Download favourites.In order for this to work you need FileSaver.js.Chainable:Yes

.enablePreview()

When you execute this method you are able to preview any border radius in your current favourites list just by clicking it.Chainable:Yes

.enableBootstrapContainer

Replace the current container of the generator with a bootstrap panel.Chainable:No <<<<<<< HEAD If you want to know wheter a method is chaiable you can call the isChainable() function and pass the name of the method as an argument and you will get true or false.

For example: isChainable("activateGenerator") will return true

If you want to know wheter a function is chainable or not you can call the the isChainable function and pass the name of the function you want to know about.You will get a true or false.

For exampe: `isChainable("getId")` will return false.

Demos

Demo

Create a generator

You have 6 ways to create a generator

1. `var generator=new Generator("#sample")`
2. `var generator=new Generator(["#sample","10em","10em",200,'green']);`
3. `var generator=new Generator("#sample","#custom_generator_object");`
4. `var generator=new Generator("#sample","#custom_generator_object");`
5. `var generator=new Generator("#sample","#custom_generator_object",true);`
6. `var generator =new Generator("#sample",null,true);`

.init(options)

```
var options=["15em","15em",200,'#fa8103']
generator.init(["15em","15em",200,'#fa8103']);
```

.setStep(value)

7. `generator.setStep(-100);`

8. ###Wrong way
   ``` generator.setStep('100a');

   **Throws an error!!!**

### .setMax(max)

9. `generator.setMax(100);`
10. `generator.setMax(-100);`

### .setSize(height,width)

```
generator.setSize("12em","12em");
```

.setBackground(bg,true)

11. `generator.setBackground('red')`
12. `generator.setBackground("#faa718");`
13. `generator.setBackground("rgb(20,98,12)");`
```

```
14. generator.setBackground("rgba(20,187,87,0.9");
15. generator.setBackground("url('path/to/image.png')no-repeat
    center center fixed");
16. generator.setBackground('red',true)
17. generator.setBackground('',true)
18. generator.setBackground(null,true)
```

.downloadFavourites()

```
generator.downloadFavourites
```