Container3D

What is Container3D?

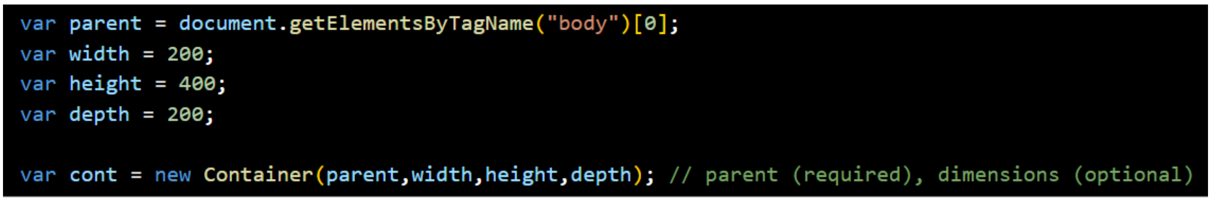
Container3D is a JavaScript framework, created using object oriented JavaScript and CSS and is supposed to help in the process of creating three-dimensional objects. Container3D is built with classes and developed using the ECMAScript 5 standard which entails that it is fully supported in all modern browsers.

For the time being it is only possible to create rectangular, six sided objects since parts of the methods adjusting the shape are currently manually controlled. This is hence the preliminary idea for the project was to create a three-dimensional

Tutorial

To use the framework the only thing needed is the compressed Container3D javascript file, for instance by adding it to an HTML page with a script (<script>) tag.

To create a three-dimensional object the Container-object is instantiated which creates an object according to the parameters it gets.



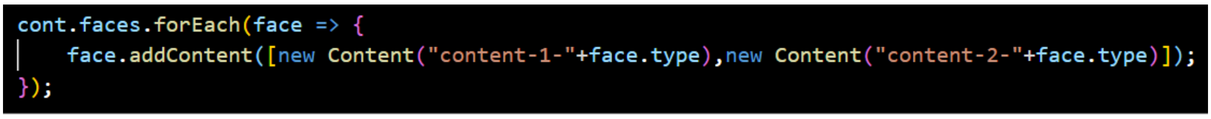
```
var parent = document.getElementsByTagName("body")[0];
var width = 200;
var height = 400;
var depth = 200;

var cont = new Container(parent,width,height,depth); // parent (required), dimensions (optional)
```

Instantiation of a 3D-object with the width and depth 200px, height 400px and is displayed in the body-element of the current HTML-document (picture text).

The container then creates its faces and appears within the parent element. To change the position or behaviour of the 3D-object it is the Container that is to be modified. Whenever the faces of the object are to be changed the faces can be changed through their parent Container or by reaching them through their classnames.

Through Container it is possible to use the showFace-method to change the face which is showing to the user by just calling the method and passing type of the face that is to be shown as a parameter.



```
cont.faces.forEach(face => {
    face.addContent([new Content("content-1-"+face.type),new Content("content-2-"+face.type)]);
});
```

Adding content to each face of the container and giving each a unique class name according to the face and order of contents

Container

These are the allowed parameters, the class-methods and properties of the Container class, accompanied by a description of what they do.

Parameters

Parent: Element (required)

- the 3D-objects parent element
- the **parent**-parameter needs to be an existing node in the DOM
- the 3D-object will be appended to the parent-node
- if no **parent** is received the object **will not** be displayed
- **NOTE:** add the “in”-class to show the object from the inside, default is to show it from the outside

Width: Number (optional)

- the required width of the object (specified in pixels)
- if no **width** is received the width is set to match the width of the **parent** element

Height: Number (optional)

- the required height of the object (specified in pixels)
- if no **height** is received it is set to match the height of the **parent** element

Depth: Number (optional)

- the required depth of the object (specified in pixels)
- if no **depth** is received but there is a width set, the depth will be set to the same as the width and otherwise it is set to the width of the **parent element**

Methods

showFace(String type)

- method to show one of the objects faces (sides)

- the received type should match the type of the face that should be shown
- NOTE: the face is shown from the outside as default

Properties

elem: Element

- a div-element representing the Container in the DOM
- has the classnames *cont* and *show*

faces[front,back,left,right,top,bottom]: Array

- array containing the objects faces (sides)
- the faces are objects; instances of the Face class (see the next class)
- can be used to change content, background, behaviour etc.

currentFace: String

- the name of the type of face that is currently showing

Face

Parameters

Type: String (required)

- the name of the type of face it is e.g. *front, back* or *bottom*
- the type is used to refer to or identify the faces while showing or changing the face

Methods

addContent([Element cont])

- adds content to the face-object
- the content should be passed in an array and can therefore contain one to many objects
- the content should be objects and have a element called *elem* to work with the method

Properties

type: String

- the type of the face, is the same as the parameter passed to the class

elem: Element

- a div-element representing the Face in the DOM
- has the classnames *face* and *face-* - **type** (which means that all faces has different classes)

content: Array

- array with all the content in the specific face

Content

Parameters

C: String (required)

- classname that is used to identify the content

Handler: String (optional)

- optional event handler, is added with the *click* listener

Methods

- there is currently no public methods for the Content class

Properties

elem: Element

- the div-element that is created during the instantiation of a Content-object
- has the classnames *content* and that of the **C** parameter

Container3D background

Container3D version 1.0 was released in 2023.

The framework is a project by Zimone Björnler, developed as an assignment while studying web technology at Linnaeus University.

The project is functional but not considered fully completed and is supposed to be further developed.

Container3D examples

https://elegant-sunburst-1de2b2.netlify.app/