A Developer Diary
{about:"code learn and share"}

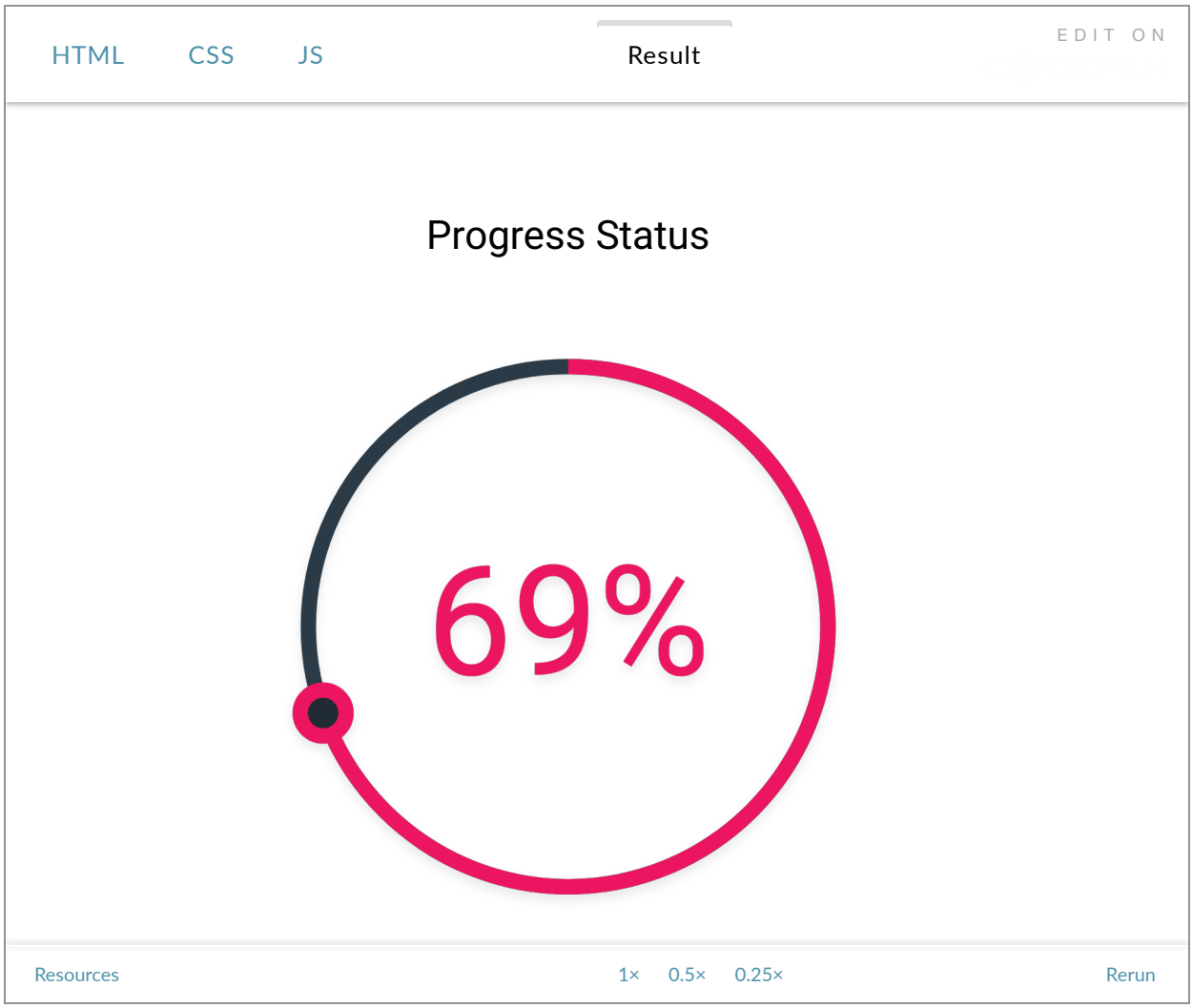
February 19, 2016 By Abhisek Jana — Leave a Comment (Edit)

Create custom progress chart using d3.js – Part1

So far we have learnt many ways to create progress chart using d3.js. In this **Create custom progress chart using d3.js – Part1** tutorial we will learn how to animate an object (e.g. Icon) along with the progress indicator.

You can refer my previous post on How to create Progress chart using d3.js in case you are not able understand any part here.

As always, lets look at the demo first to get an idea of what we will be creating today. We have a simple progress chart and another small circle is following the progress indicator.



Lets start with the basic first. Create two arcs, one for the background and another for the foreground.

```
var percent = 55;

var pie=d3.layout.pie()
        .value(function(d){return d})
        .sort(null);

var w=300,h=320;

var outerRadius=(w/2)-10;
var innerRadius=outerRadius-8;

var color = ['#ec1561','#2a3a46','#202b33'];
```

```
var arc=d3.svg.arc()
        .innerRadius(innerRadius)
        .outerRadius(outerRadius)
        .startAngle(0)
        .endAngle(2*Math.PI);
var arcLine=d3.svg.arc()
        .innerRadius(innerRadius)
        .outerRadius(outerRadius)
        .startAngle(0);
```

In addition to these, we will add another arc. Our small circle will follow this arc.

```
//The circle is following this
var arcDummy=d3.svg.arc()
        .innerRadius((outerRadius-
innerRadius)/2+innerRadius)
        .outerRadius((outerRadius-
innerRadius)/2+innerRadius)
        .startAngle(0);
```

Now add the `svg`, path for background, foreground and dummy circle.

```
var svg=d3.select("#chart")
        .append("svg")
        .attr({
            width:w,
            height:h,
            class:'shadow'
        }).append('g')
        .attr({
            transform:'translate('+w/2+','+h/2+')'
```

```
        });

//background
var path=svg.append('path')
        .attr({
            d:arc
        })
        .style({
            fill:color[1]
        });

var pathForeground=svg.append('path')
        .datum({endAngle:0})
        .attr({
            d:arcLine
        })
        .style({
            fill:color[0]
        });

//Dummy Arc for Circle
var pathDummy=svg.append('path')
        .datum({endAngle:0})
        .attr({
            d:arcDummy
        }).style({
            fill:color[0]
        });
```

Then create a circle svg element, We will position it appropriately to the top of the chart, which would be the starting point. We will use the `transform` property and `translate` function for this.

```
var endCircle=svg.append('circle')
        .attr({
            r:12,
            transform:'translate(0,'+ (-outerRadius+15)
+')'
        })
        .style({
            stroke:color[0],
            'stroke-width':8,
            fill:color[2]
        });
```

Next, lets add the text in the middle.

```
var middleTextCount=svg.append('text')
        .datum(0)
        .text(function(d){
            return d+'%';
        })

        .attr({
            class:'middleText',
            'text-anchor':'middle',
            dy:25,
            dx:0
        })
        .style({
            fill:'#ec1561',
            'font-size':'80px'

        });
```

The `arcTweenOld` function is the most critical one. We need to get the location of the `arcDummy` then use string manipulation to get the coordinate. If you print

the value of the `arcDummy(d)` you will get something like this in the console.

```
M8.327598234202001e-15,-136A136,136 0 1,1
-104.8109177516433,86.66413052733098L-
104.8109177516433,86.66413052733098A136,136 0 1,0
8.327598234202001e-15,-136Z
```

The data between L – A is the actual coordinate. Once we use the `spilt()` function it would look like the following.

```
-104.8109177516433, 86.66413052733098
```

Now we need to translate the circle to this coordinate to make it look like as it's following the progress foreground.

```
var arcTweenOld=function(transition, percent,oldValue) {
    transition.attrTween("d", function (d) {

        var newAngle=(percent/100)*(2*Math.PI);

        var interpolate = d3.interpolate(d.endAngle,
newAngle);

        var interpolateCount = d3.interpolate(oldValue,
percent);

        return function (t) {
            d.endAngle = interpolate(t);
            var pathForegroundCircle = arcLine(d);


middleTextCount.text(Math.floor(interpolateCount(t))+'%');
```

```
            var pathDummyCircle = arcDummy(d);
            var coordinate = pathDummyCircle.split("L")
[1].split("A")[0];

            endCircle.attr('transform', 'translate(' +
coordinate+ ')');

            return pathForegroundCircle;
        };
    });
};
```

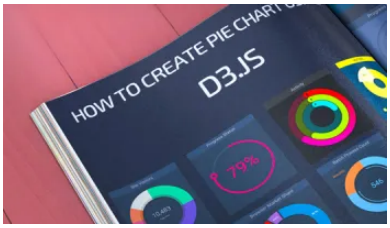
At the end, we have an animation & timeout function.

```
var oldValue=0;

var animate=function(){
    pathForeground.transition()
            .duration(750)
            .ease('cubic')
            .call(arcTweenOld,percent,oldValue);

    oldValue=percent;
    percent=(Math.random() * 60) + 20;
    setTimeout(animate,3000);
};

setTimeout(animate,0);
```

Please find the full code in Github.

Related

Create custom progress chart using d3.js – Part2
In "D3.js"



Create Pie Charts using D3.js
In "D3.js"

Create a Simple Pie Chart using D3.js
In "D3.js"

Filed Under: D3.js, JavaScript Progress Chart, Visualization

Tagged With: chart, d3.js, Donut Charts, JavaScript, Pie Chart,

Subscribe to stay in loop

* indicates required

Email Address *

Subscribe

Leave a Reply

Logged in as Abhisek Jana. Edit your profile. Log out? Required fields are marked *

Comment *

Post Comment

This site uses Akismet to reduce spam. Learn how your comment data is processed.