

D3 + ANGULAR

VISUAL AWESOMESAUCE

John Niedzwiecki

ABOUT THE GEEK

```
function getDeveloperInfo() {
  let dev = {
    name: 'John Niedzwiecki',
    desc: 'Your friendly neighborhood kilted coder.',
    company: 'VIPRE Security',
    role: 'Lead Software Engineer - UI',
    twitter: '@rhgeek',
    github: 'rhgeek',
    web: 'https://rhgeek.github.io',
    email: 'john.niedzwiecki.ii@gmail.com',
    currentDevLoves: ['Angular', 'JavaScript Everywhere', 'd3', 'CSS instead of JS'],
    nonDevAttr : {
      isHusband: true,
      isFather: true,
      hobbies: ['running', 'disney', 'cooking', 'video games', 'photography'],
      twitter: '@rgrdisney',
      web: 'http://www.rungeekrundisney.com',
    }
  };
  return dev;
}
```

THE WORLD IS DATA

WHAT WE'LL DO

- ★ What is D3?
- ★ General Update Pattern
- ★ What is Angular?
- ★ Combine D3 + Angular



WHAT IS D3?



WHAT IS D3?

Open source JavaScript library that provides powerful manipulation of the Document Object Model (DOM) driven by data.

<https://github.com/d3/d3>

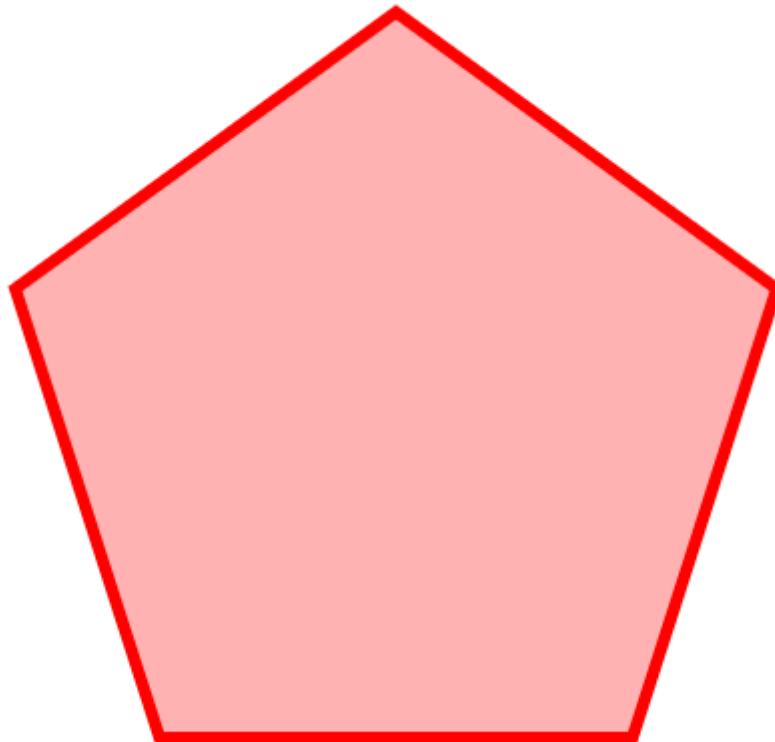
CREATING SHAPES

Easy to create whatever shapes you want.

CREATING SHAPES: PENTAGON

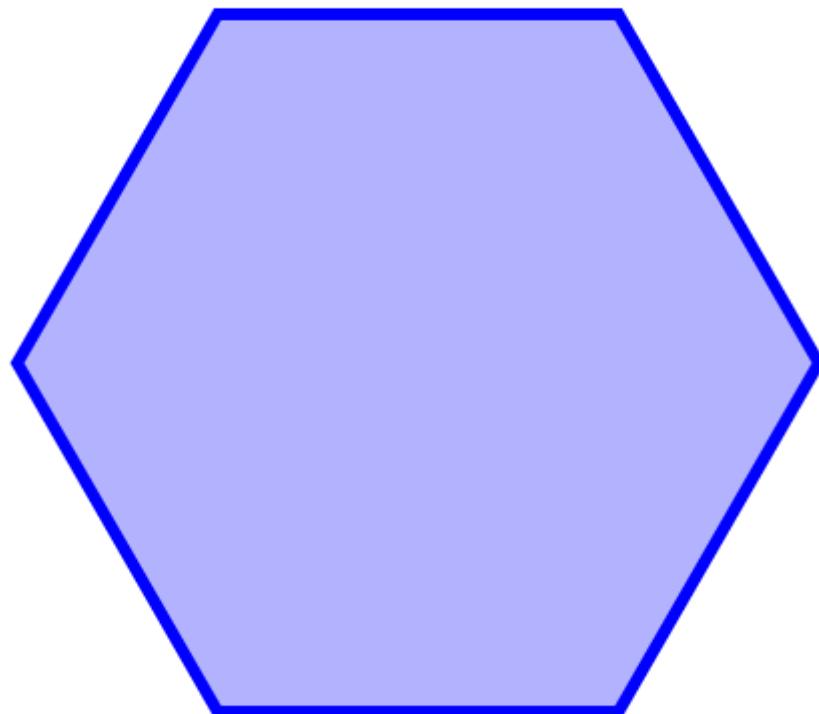
```
var pentagon
d3.line()
  .x(function(d) {
    return d.x;
  })
  .y(function(d) {
    return d.y;
  })
  .curve(d3.curveLinear);

var pentData = [
  { 'x': 100, 'y': 100 },
  { 'x': 150, 'y': 200 },
  { 'x': 200, 'y': 250 },
  { 'x': 250, 'y': 200 },
  { 'x': 300, 'y': 100 }
];
var svg =
```



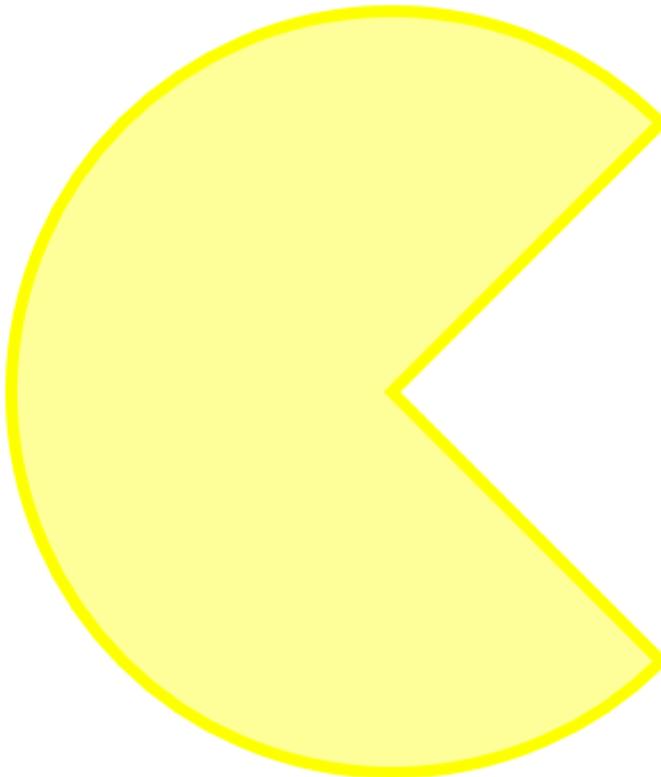
CREATING SHAPES: HEXAGON

```
var polygon =  
  d3.line()  
    .x(function(d) {  
      return d.x;  
    })  
    .y(function(d) {  
      return d.y;  
    })  
    .curv  
  
var hexData =  
  [ { 'x': 100, 'y': 100 }, { 'x': 150, 'y': 100 }, { 'x': 200, 'y': 100 }, { 'x': 250, 'y': 100 }, { 'x': 300, 'y': 100 }, { 'x': 350, 'y': 100 } ];
```

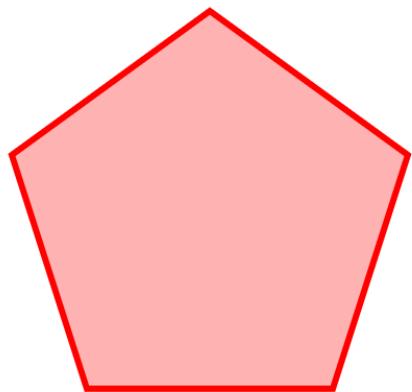


CREATING SHAPES: ARC

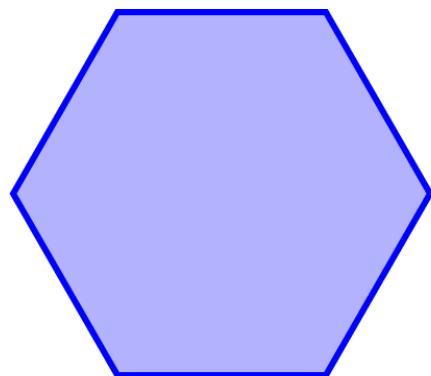
```
var arcGen =  
  .innerRad  
  .outerRad  
  .startAng  
  .endAngle  
  
var svg =  
  d3.select  
    .appe  
    .attr  
    .attr  
  
svg.append('p'  
  .attr('d'  
    a++r/ 's+
```



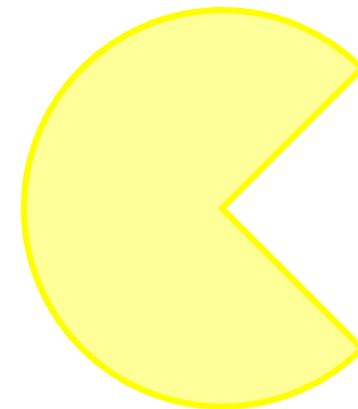
CREATING ALL THE 'GONS'



Pentagon



Hexagon



Omnomnomagon

EXAMPLE

Let's do more than your normal "draw a line" tutorial.

EASY AS SVG

Getting started is as easy as SVG

```
var width = 500,  
    height = 500;  
var svg = d3.select('body')  
  .append('svg')  
    .attr('width', width)  
    .attr('height', height)  
  .append('g')  
    .attr('transform', 'translate(' + width / 2 + ',' + height / 2)')
```

LET'S GET FUNCTIONAL

Power by defining a function to use with data. In our case arc.

```
var xScale = d3.scaleLinear()  
    .range([0, 2 * Math.PI]);  
var yScale = d3.scaleSqrt()  
    .range([0, radius]);  
  
var arc = d3.arc()  
    .startAngle(function(d) { return Math.max(0, Math.min(2 * Mat  
    .endAngle(function(d) { return Math.max(0, Math.min(2 * Math.  
    .innerRadius(function(d) { return Math.max(0, yScale(d['y0']))  
    .outerRadius(function(d) { return Math.max(0, yScale(d['y1']))
```

BRING ON THE DATA!

GENERAL UPDATE PATTERN

GENERAL UPDATE PATTERN

D3 has a pattern for how to deal with your data.

- ★ join: `data()`
- ★ enter: `append()`
- ★ update: `transition()`
- ★ exit: `exit()`

GENERAL UPDATE PATTERN

```
root = d3.hierarchy(tree);
root.sum(function(d) { return d.size; });

/* JOIN new data with old elements */
var path = svg.selectAll('path')
  .data(partition(root).descendants());

/* ENTER new elements present in new data */
/* UPDATE old elements present in new data */
path.enter()
  .append('path')
    .on('mouseover', mouseoverArc)
    .on('mousemove', mousemoveArc)
    .on('mouseout', mouseoutArc)
    .on('click', clickArc)
```



WHAT IS ANGULAR?



WHAT IS ANGULAR?

“Angular is a development platform for building mobile and desktop web applications.”

<https://github.com/angular/angular>

KEY PIECES OF ANGULAR

- ★ Components
- ★ Services
- ★ Router
- ★ Pipes
- ★ Modules

ANGULAR COMPONENTS

Angular components are your base building block for your UI. In the end, your Angular application is a tree of components, so, yeah, they're pretty important.

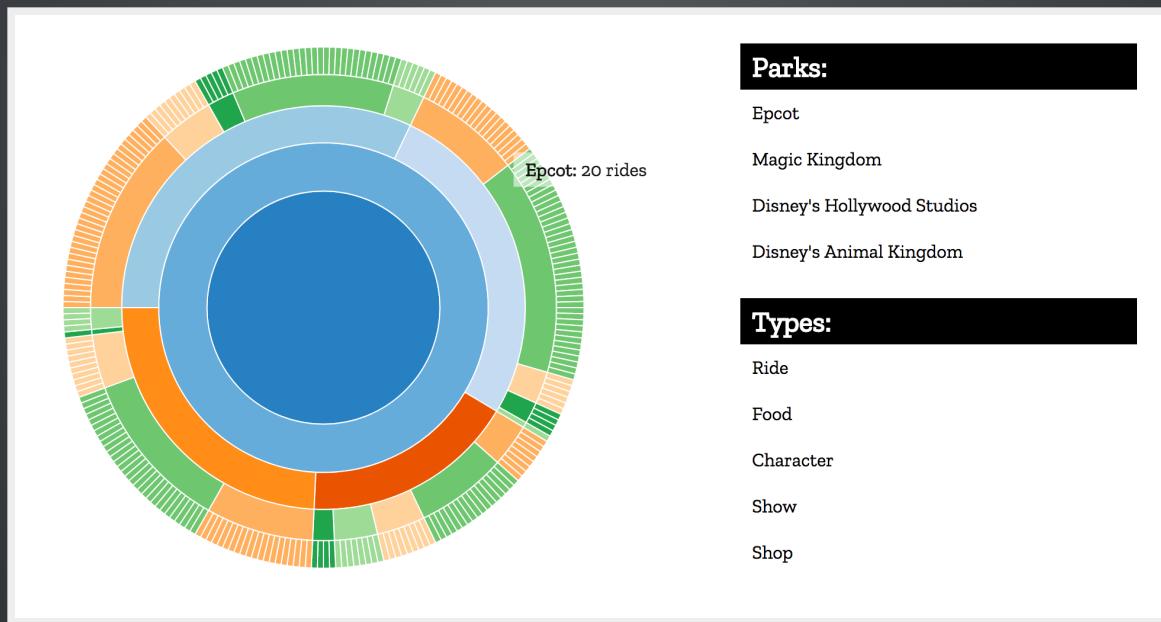


**WITH OUR POWERS
COMBINED...**



WITH OUR POWERS COMBINED...

Together we can create visual awesomesauce!



ADD D3 TO OUR ANGULAR APPLICATION

Install dependencies:

```
npm install --save d3
```

```
npm install --save-dev @types/d3
```

Import in our component:

```
import * as d3 from 'd3';
```

RESTRUCTURE OUR D3 CODE WITHIN THE COMPONENT

- ★ Create variables
- ★ Setup in `ngOnInit`
- ★ Move chart creating into own function

RESTRUCTURE: NGONINIT

```
ngOnInit() {
  this.created = false;
  this.attractionTree = {
    name : 'Theme Parks',
    children : []
  };
  this.margin = 10;
  this.createChart();
  this.updateChart(this.attractionTree);
}
```

RESTRUCTURE: CREATE CHART

```
@ViewChild('chart')
private chartContainer: ElementRef;

private createChart(): void {
  let element = this.chartContainer.nativeElement;
  this.width = element.offsetWidth - this.margin * 2;
  this.height = element.offsetHeight - this.margin * 2;
  this.radius = (Math.min(this.width, this.height) / 2) - this.

  /* create svg */
  this.svg = d3.select(element).append('svg')
    .attr('width', element.offsetWidth)
    .attr('height', element.offsetHeight)
    .append('g')
      .attr('transform', 'translate(' + this.width / 2 + ', ' +
```

RESTRUCTURE: UPDATE CHART

```
private updateChart(tree: any): void {
  this.root = d3.hierarchy(tree);
  this.root.sum(function(d) { return d.size; });

  /* JOIN new data with old elements */
  let path = this.svg.selectAll('path')
    .data(this.partition(this.root).descendants());

  /* ENTER new elements present in new data */
  /* UPDATE old elements present in new data */
  path.enter()
    .append('path')
    .on('mouseover', (d) => this.mouseoverArc(d))
    .on('mousemove', (d) => this.mousemoveArc())
    .on('mouseout', (d) => this.mouseoutArc())
```

COMMUNICATE BETWEEN COMPONENTS

- ★ Receive data in through @Input
- ★ Pass data back out with @Output emitter

@INPUT

```
@Input()
set attractions(attrs: ParkAttraction[ ]) {
    if (attrs) {
        this._attractions = attrs;
        this.attractionTree = this.dataToTree(attrs);
        if(this.created) {
            this.updateChart(this.attractionTree);
        }
    }
}
```

@OUTPUT

```
@Output()  
onArc: EventEmitter<any> = new EventEmitter<any>();  
  
private clickArc(d: any): void {  
    let clickedFilter = {};  
    if(d.data.level == 'parkName') {  
        clickedFilter['parkName'] = d.data.name;  
    } else if(d.data.level == 'type') {  
        clickedFilter['type'] = d.data.name;  
        clickedFilter['parkName'] = d.parent.data.name;  
    }  
    if(clickedFilter['parkName']) {  
        this.onArc.emit(clickedFilter);  
    }  
}
```

HTML TEMPLATE

```
<rhsun-park-sunburst [attractions]="attractions" [filters]="filte  
<rhsun-filter-bar [arcfilter]="arcFilter" (onfilter)="onFilterBar
```



LET'S GO TO THE
GAMETAPE, PHIL.

SOURCE

<https://github.com/RHGeek/d3-ng-visual-awesomesauce/example>

CONNECT . TECH

Find me: @rhgeek

<https://rhgeek.github.io/>

Code: <https://github.com/RHGeek/d3-ng-visual-awesomesauce>