

# D3js Tutorial

<http://www.washingtonpost.com/wp-srv/special/politics>

# Contents

- Some important aspects of JavaScript
- Introduction to SVG
- CSS
- D3js
- Hands On
- Further Reading

# JavaScript - Datatypes

- Variable declaration

```
var a = "Arr";  
var b = 23;
```

- Datatypes are handled automatically

→ may lead to problems!

```
var a = 1;  
var b = 2;  
var c = a + b;           // c = 3
```

# JavaScript - Datatypes

- Variable declaration

```
var a = "Arr";  
var b = 23;
```

- Datatypes are handled automatically

→ may lead to problems!

```
var a = 1;  
var b = 2;  
var c = a + b;           // c = 3
```

```
var a = 1;  
var b = "2";  
var c = a + b;           // c = 12 (!)
```

# JavaScript - Datatypes

- Variable declaration

```
var a = "Arr";  
var b = 23;
```

- Datatypes are handled automatically

→ may lead to problems!

```
var a = 1;  
var b = 2;  
var c = a + b;           // c = 3
```

```
var a = 1;  
var b = "2";  
var c = a + +b;         // c = 3 (casted b to a number)
```

# JavaScript - Arrays

- declaration

```
var a = [];  
var a = new Array();
```

- iterating:

```
for(var i = 0; i < a.length; i++)  
{  
    alert(a[i]);  
}
```

```
var myfunction = function(element)  
{  
    alert(element);  
}
```

```
a.forEach( myfunction );
```

# JavaScript - Objects

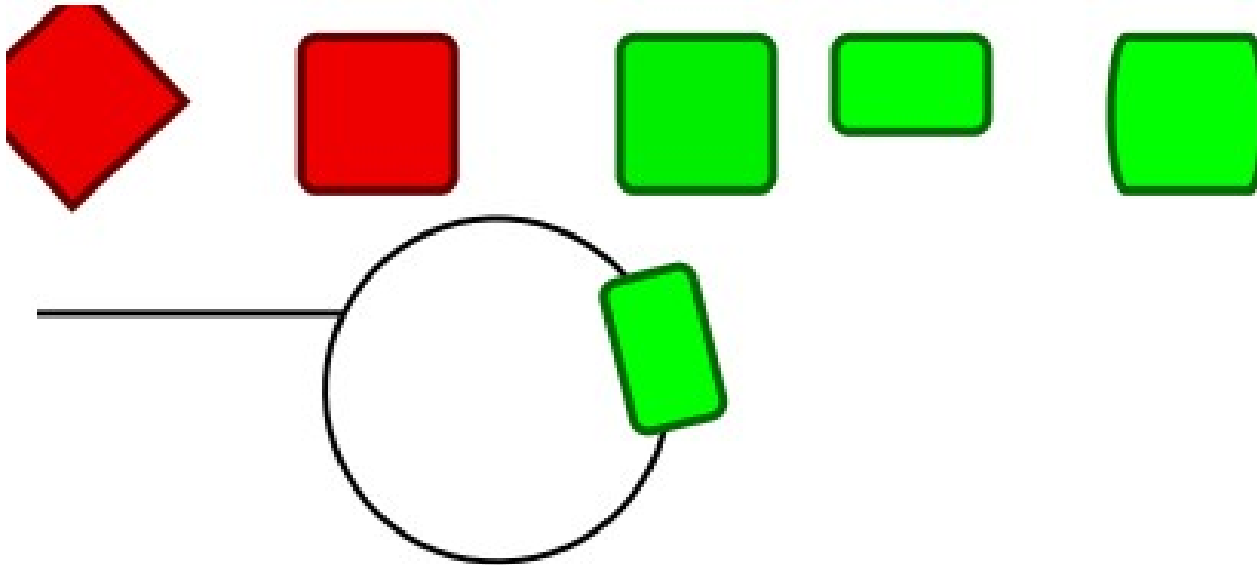
- Everything is a object
- Objects can be extended easily

```
var obj = {};  
                // we have an empty object now  
  
obj.speed = 10;  
obj["arr"] = 1;  
                // obj has now a member!  
                // objects can be used as a map
```



# Scalable Vector Graphics

- Extending HTML with graphics

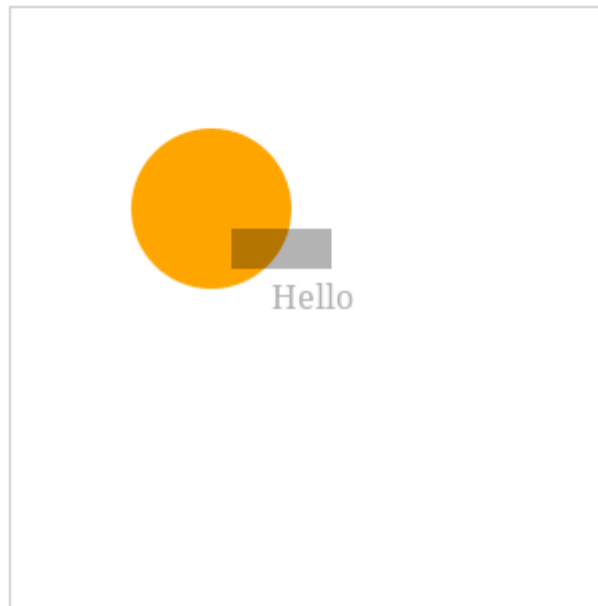


- New Tags `<svg>`, `<g>`, `<rect>`, `<circle>`, `<text>`
- <http://www.w3schools.com/SVG>

# SVG

- How to use?

```
<svg width="300" height="300" style="border: 1px solid #CCCCCC">  
  <circle cx="100" cy="100" r="40" fill="orange"/>  
  <g id="testgroup" opacity="0.3">  
    <rect x="110" y="110" width="50" height="20" />  
    <text x="130" y="150">Hello</text>  
  </g>  
</svg>
```

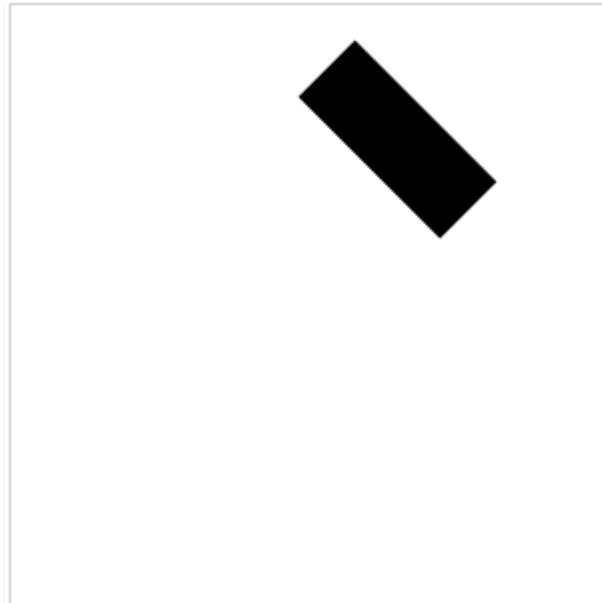


# SVG - transformations

- Translate, Rotate, Scale:

```
<svg width="300" height="300" style="border: 1px solid #CCCCCC">  
  <rect width="50" height="20"  
    transform="translate(50 50) scale(2 1) rotate(45 25,5)" />  
</svg>
```

- Take care of the order!



# CSS

- Styling your graphics
- Write a css-file containing e.g.:

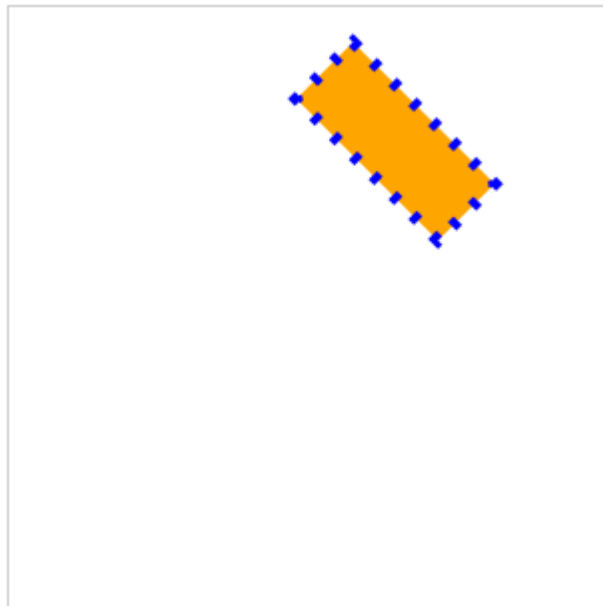
```
svg {  
  border: 1px solid grey;  
}  
  
rect {  
  stroke: blue;  
  stroke-dasharray: 2,5;  
  stroke-width: 3;  
  fill: orange;  
}
```

- Include it in your html-file:

```
<link href="test.css" rel="stylesheet" type="text/css" />
```

# CSS

```
svg {  
  border: 1px solid grey;  
}  
  
rect {  
  stroke: blue;  
  stroke-dasharray: 2,5;  
  stroke-width: 3;  
  fill: orange;  
}
```



# CSS

- What can you manipulate?

field	description
<code>stroke: {color}</code>	Sets the stroke-color (used to outline the shape)
<code>stroke-width: {in pixel}</code>	Sets the width of the stroke
<code>stroke-opacity: {0.0 .. 1.0}</code>	Sets the opacity
<code>stroke-linecap: {round, butt or square}</code>	Sets the line-caps
<code>fill: {color}</code>	Sets the fill-color
<code>fill-opacity: {0.0 .. 1.0}</code>	Sets the fill-opacity of the shape

# D3js

- JavaScript-Library like jQuery
- <http://www.d3js.org>
- Modify the HTML-DOM easily
- Support for animations
- Compatibility: only problems with old versions of IE (< v9) and Android Browser (< v3)

# D3js – function chaining

```
var a = d3.select("#barchart");  
var b = a.append("g");  
var c = b.attr("transform", "translate(10,10)");
```

```
d3.select("#barchart").append("g").attr("transform",  
"translate(10,10)");
```

-or-

```
d3.select("#barchart")  
  .append("g")  
  .attr("transform", "translate(10,10)");
```



# D3js – create elements

```
var chart = d3.select("chart");  
  
chart.selectAll(".group")  
  .data(myData)  
  .enter().append("rect");
```

- Automatically **adds new** elements for the given data
- If myData has elements from a previous call new elements won't be created
  - enter() only applied onto not-existent data-elements

# D3js – load data

- Loading CSV/TSV-files

```
d3.csv(filename, doneCallback);  
d3.tsv(filename, doneCallback);
```

- is loaded asynchronously

→ callback is called, when all data are read

```
d3.csv(filename, function(error, data)  
    {  
        data.forEach(function(d)  
            {  
                d.year = +d.year;  
            }  
            // ...  
        }  
    );
```

# D3js – Mapping Coordinates

- We have data in data-space → how to map them to screen-space?
- D3 offers simple helpers:

```
var y = d3.scale.linear()  
  .range([height, 0])  
  .domain([minimum, maximum]);
```

function(d) {return y(d); } will give the correct screen-coordinate for the given value d

- linear, log, ordinal, category10

# Interaction - Drag

```
function dragged() {  
  chart.selectAll(".bar")  
    .attr("y", d3.event.y)  
    .attr("height", height - d3.event.y);  
}  
  
// Add a drag handle to change the return bar  
chart.selectAll(".bar")  
  .call(d3.behavior.drag().on("drag", dragged));
```

# Hands On / Live Demo

# Further Information

- Tools and Toolkits
  - Zurb Foundation
  - Twitter Bootstrap
  - JQuery
- See also Resources-list on course-site

Thank you for  
your attention