**D3**

1. **可定制化程度高，但是复用性低，开发的流程相对固定。**
2. 创建基本流程：
   1. 引入d3库;
   2. 加载数据;
   3. 创建比例尺;
   4. 创建svg;
   5. 注入数据绘制图表，设置过渡动画，绑定交互事件;
   6. 添加辅助信息。(数轴，提示标签，气泡)。

3. 静态图表的创建

基本图形的创建

柱状图 条形图 饼图 折线图 面积图

4. 动态图表的创建

动态图表：图表在某一时刻形状，颜色，位置等的变化，用户可以看到变化的过程

实现过程用到过渡。

1. transition()
2. duration()
3. ease()

里面有四个参数：llinear cirde elastic bounce

1. delay()
2. 比例尺：
3. 线性比例尺：d3.scale.linear()

常用方法：

1) d3.scale.linear()   创建一个线性比例尺

          .domain([0,500]) 定义域

          .range([0,1000]) 值域

**2) linear(x)  输入定义域 返回 值域**

**3) linear.invert(y) 输入值域，返回定义域**

**4) linear.domain([numbers]) 设定或获取定义域**

**5) linear.range([values])设定或获取值域**

6) linear.rangeRound([valuses]) 四舍五入，结果是整数

7) linear.clamp([boolean])     false不收缩  true 收缩

8) linear.nice([count])  扩展成理想状态

9) linear.ticks([ticks]) 默认是10 用于选取坐标刻度

10) linear.ticksFormat(count,[format]) 设置定义域内具有代表的值的表示形式，如小数

1. 序数比例尺：d3.scale.ordinal()

常用方法：

1) d3.scale.orinal() 构建一个序数比例尺

2) ordinal(x) 返回y

3) ordinal.domain([values]) 设定和获取x

4) ordinal.range([values]) 设定或获取y

5) ordinal.rangePoint(interval,[padding])  设置值域

6) ordinal.rangeRoundPoints(interval,[padding]) 和ranggePoint()一样，但是会将结果取正

7) ordinal.rangeRoundBands(interval,[padding],[outerpadding]) 代替 range()设定值域

8) ordinal.rangeRoundRoundBands()  结果取整

9) ordinal.rangeBand() 返回使用rangeBands（） 设定后每一段的宽度

10) ordinal.rangeExtend()  返回一个数组，值域的最大值和最小值

6. update enter exit（当选择集和数据的数量关系不确定的情况）

(1) enter：数据比元素多，会用到enter()

(2) update

(3) exit

7. 图表交互：在图形元素上设置一个或者多个事件监听器

8. 饼图：

1) 声明数据dataset，但是不能直接用于绘制图形，绘制饼图的一部分需要知道一段弧起始角度和终 止角度。通过d3.layout.pie() 声明布局，他的返回值为一个函数，然后将dataset作为参数传给它， 返回的数据即为转换后可以用的数据，是一个对象（起始角度，终止角度，原数据）。

2) 生成器：

path() 通过定义一段路径来绘制各种图形

弧生成器：

d3.svg.arc() 弧生成器

.innerRadius() 设置内半径

.outerRadius() 设置外半径

9. 折线图：