

数据可视化

1. 什么是数据可视化

1.1 数据可视化

- 数据可视化主要目的：借助于图形化手段，清洗有效地传达与沟通信息
- 数据可视化可以把数据从冰冷的数字转换成图形，揭示蕴含在数据中的规律和道理。

1.2 数据可视化的场景

目前互联网公司通常由这么几大类的可视化需求：

- 通用报表
- 移动端图标
- 大屏可视化
- 图编辑&图分析
- 地理可视化

1.3 常见的数据可视化库

- D3.js 目前 Web 端评价最高的 JavaScript 可视化工具库（入手难）
- ECharts.js 百度出品的一个开源 JavaScript 数据可视化库
- Highcharts.js 国外的前端数据可视化库，非商用免费，被许多国外大公司所使用
- AntV 蚂蚁金服全新一代数据可视化解决方案
- 等等

Highcharts 和 Echarts 就像是 Office 和 WPS 的关系

1.4 小结

- 数据可视化主要目的：借助于图形化手段，清洗有效地传达与沟通信息
- 数据可视化在我们互联网公司中经常用于通用数据报表，移动端图标，大屏可视化，图编辑等
- 数据可视化库有很多，接下来我们重点学习ECharts

2. 数据可视化项目概述

2.1 项目目的

市场需求：

应对现在数据可视化的趋势，越来越多企业需要在很多场景（营销数据，生产数据，用户数据）下使用可视化图表来展示体现数据，让数据更加直观，数据特点更加突出

学习阶段需求：

项目有着承上启下的作用。

承上

- 复习以前学习内容
- HTML5 + CSS3 布局相关技术
- JavaScript / jQuery 相关技术

启下

为学习服务器编程做铺垫

如何把服务器里面的数据渲染到页面里面？

2.2 项目技术

- HTML5 + CSS3 布局
- CSS3 动画、渐变
- jQuery库 + 原生 JavaScript
- flex布局 和 rem适配方案
- 图片边框 **border-image**
- ES6 模板字符
- ECharts可视化库等等、

大数据可视化项目代码比例

JavaScript: 27.97%

HTML: 31.47%

CSS: 40.56%

2.3 小结

- 数据可视化项目展示
- 学习这个项目的目的：市场需求和学习阶段需求
- 项目用到的技术：以前学过的技术 和 新技术
 - CSS 动画、渐变
 - jQuery库 + 原生JavaScript

- flex布局 和 rem适配方案
- 图片边框 border-image
- ESS模板字符
- ECharts 可视化库等等

3.ECharts 简介

ECharts 是一个使用 JavaScript 实现的开源可视化库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器（IE8/9/10/11,Chrome,Firefox,Safari等），底层依赖矢量图形库ZRender，提供直观、交互丰富、可高度个性化定制的数据可视化图表。

- 丰富的可视化类型
- 多种数据格式支持
- 流数据的支持
- 移动端优化
- 跨平台使用
- 绚丽的特效
- 详细文档说明

特点：

开源免费：遵循 Apache-2.0 开源协议，免费商用

功能丰富：涵盖各行业图表，满足各种需求

社区活跃

官网地址：<https://www.echartsjs.com/zh/index.html>

4. ECharts的基本使用

4.1 ECharts使用五步曲

步骤1：下载并引入echarts.js文件（图表依赖这个**js库**）

步骤2：准备一个具备大小的DOM容器（生成的图表会放入这个**容器**内）

步骤3：初始化echarts实例对象（**实例化**echarts对象）

步骤4：指定配置项和数据(option)（根据具体需求修改**配置**选项）

步骤5：将配置项设置给echarts实例对象（让u给你echarts对象根据修改好的配置**生效**）

4.2 选择不同类型图表

官网→实例→官方实例

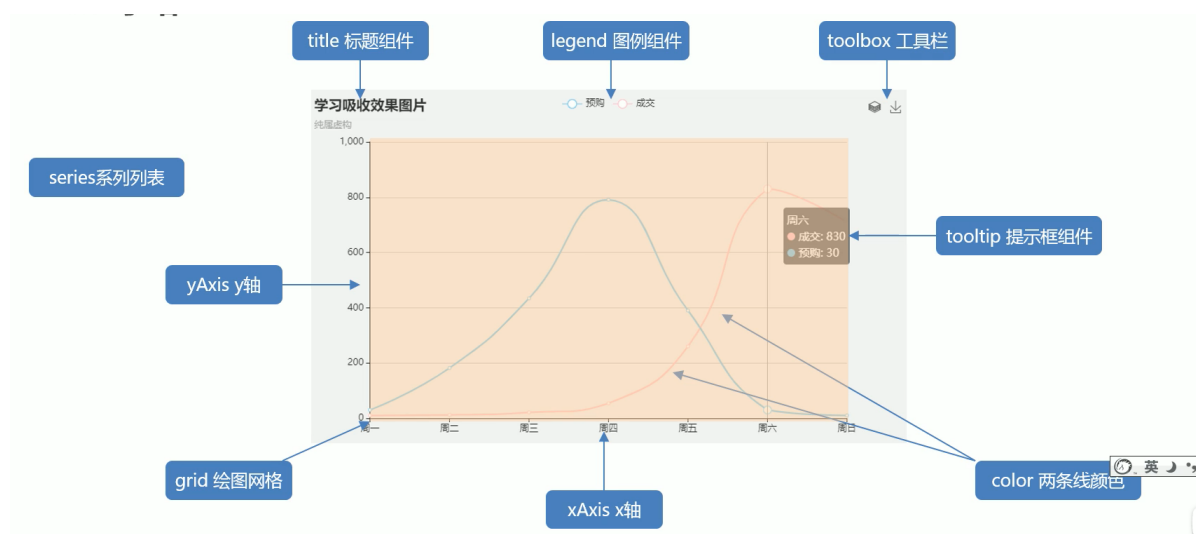
4.3 相关配置讲解

- title: 标题组件
- tooltip: 提示框组件
- legend: 图例组件
- toolbox: 工具栏
- grid: 直角坐标系内绘图网格
- xAxis: 直角坐标系 grid 中的 x 轴
- yAxis: 直角坐标系 grid 中的 y 轴
- series: 系列列表。每个系列通过 type 决定自己的图表类型（什么类型的图表）
- color: 调色盘颜色列表

先了解以上9个配置的作用，其余配置还有具体细节我们查阅文档：文档→配置项手册

学会echarts 关键在于学会查阅文档，根据需求修改配置

4.4 小结



数据可视化项目

1. 数据可视化项目适配方案

1.1 项目需求

- 设计稿是1920px
- PC端适配：宽度在 1024~1920之间页面元素宽高自适应'

1.2 适配方案

flexible.js + rem 单位 + flex布局

- flexible.js: 检测浏览器宽度，修改html 文字大小
- rem 单位: 页面元素根据 rem 适配大小，配合 cssrem 插件
- 页面快速布局

1. flexible.js 把屏幕分为 24 等份
2. PC端的效果图是1920px
3. cssrem 插件的基准值是 80px
4. rem值自动生成

要把屏幕宽度约束在1024~1920之间有适配

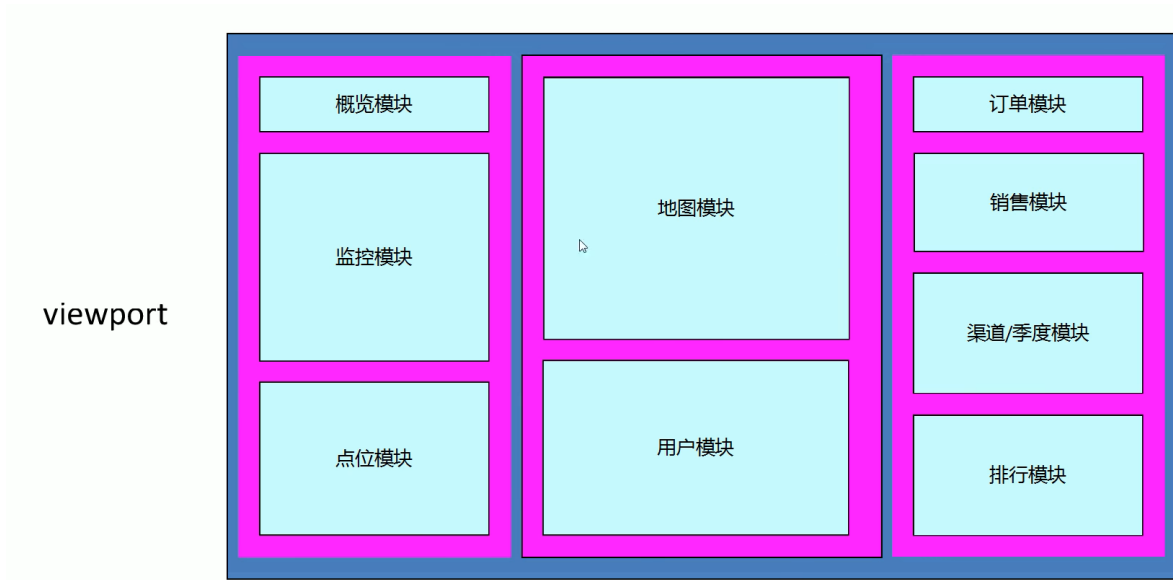
```
@media screen and (max-width:1024px) {  
  html {  
    font-size:42.66px!important;  
  }  
}  
@media screen and (min-width:1920px) {  
  html {  
    font-size:80px!important;  
  }  
}
```

保存自动刷新浏览器插件

Preview on Web Server

项目核心：先布好局，之后利用ECharts填充图表

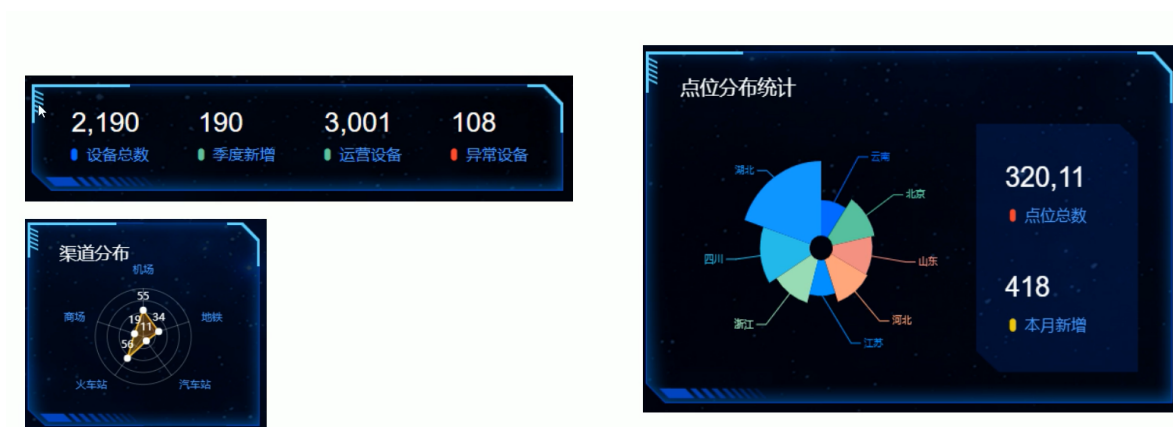
项目基础布局



2. 边框图片

1. 边框图片的应用场景

盒子大小不一，但是边框样式相同，此时就需要边框图片来完成

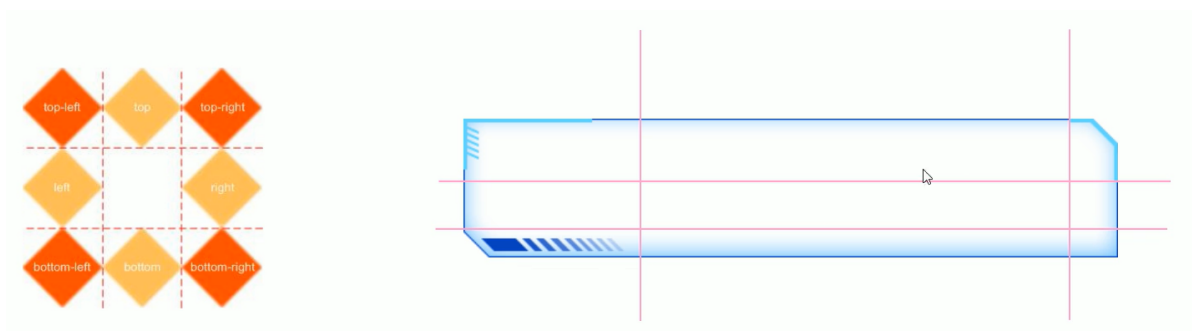


为了实现丰富多彩的边框效果，在CSS3中，新增了 border-image属性，这个新属性允许指定一幅图像作为元素的边框。



2. 边框图片的切图原理（重要）

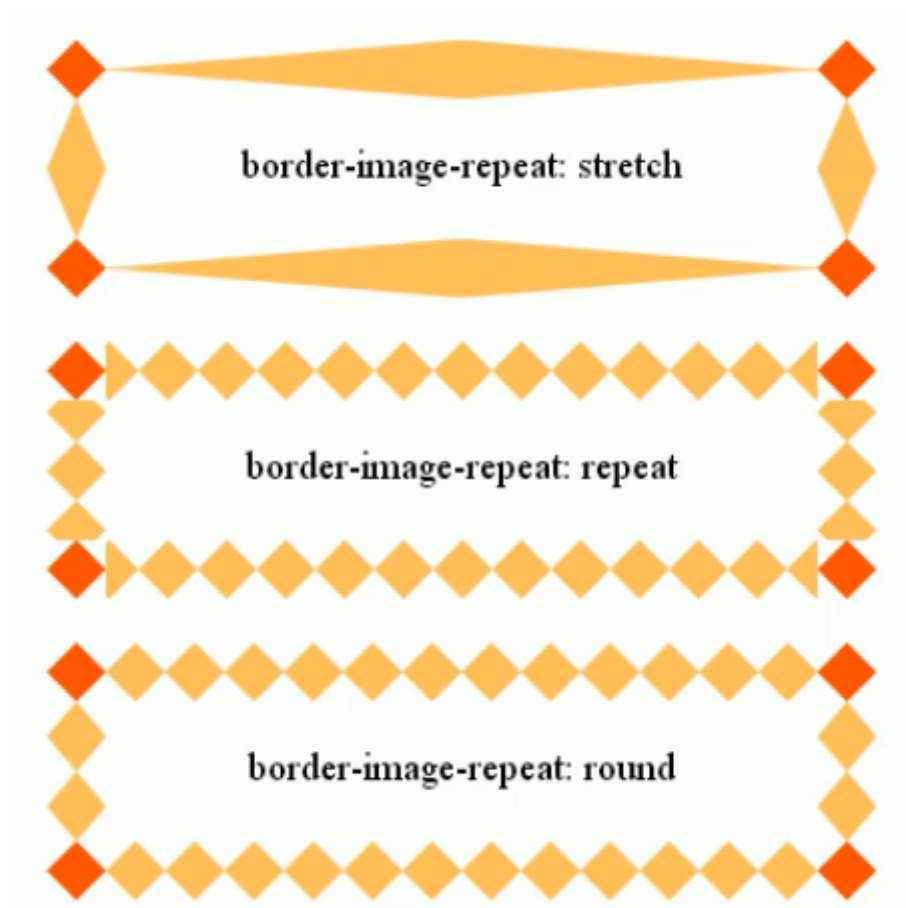
把四个角切出去（九宫格的由来），中间部分可以铺排、拉伸或者环绕



按照 上右下左 顺序切割

3. 边框图片语法（重要）

属性	描述
<code>border-image-source</code>	用在边框图片的路径。（哪个图片）
<code>border-image-slice</code>	图片边框向内偏移。（裁剪的尺寸，一定不加单位，上右下左顺序）
<code>border-image-width</code>	图片边框的宽度（需要加单位），默认是border的宽度。（不是边框的宽度，是边框图片的宽度）
<code>border-image-repeat</code>	图像边框是否平铺（repeat）、铺满（round）或拉伸（stretch）默认拉伸



注意：

border-image-width是边框图片的宽度，不会挤压文字

内容压住盒子边框实现：盒子里面再放一个盒子，把内容放到内部盒子里，内部盒子采取绝对定位，使得每个边都往外走一点，

内部盒子样式：

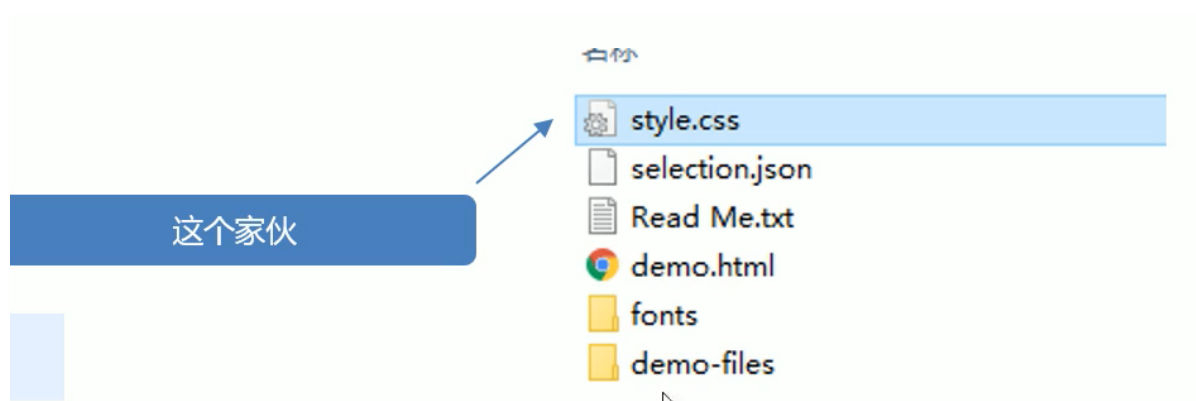
```
.inner {  
  position: absolute;  
  top: -20px;  
  left: -15px;  
  right: -10px;  
  bottom: -15px;  
}
```

//这样就能使得内部盒子中的内容贴着外部盒子最外边显示

2.4 通过类名调用字体图标

1. HTML页面引入字体图标中CSS文件。

```
<link rel="stylesheet" href="fonts/style.css">
```



2. 标签直接调用图标对应的类名即可。（类名在css文件中标注）

```
<span class="icon-angelist"></span>
```




引入css文件和声明字体图标的时候，一定要注意路径问题。

2.5 立即执行函数用法

JS文件中，会有大量的变量命名，特别是Echarts使用中，需要大量初始化Echarts对象。

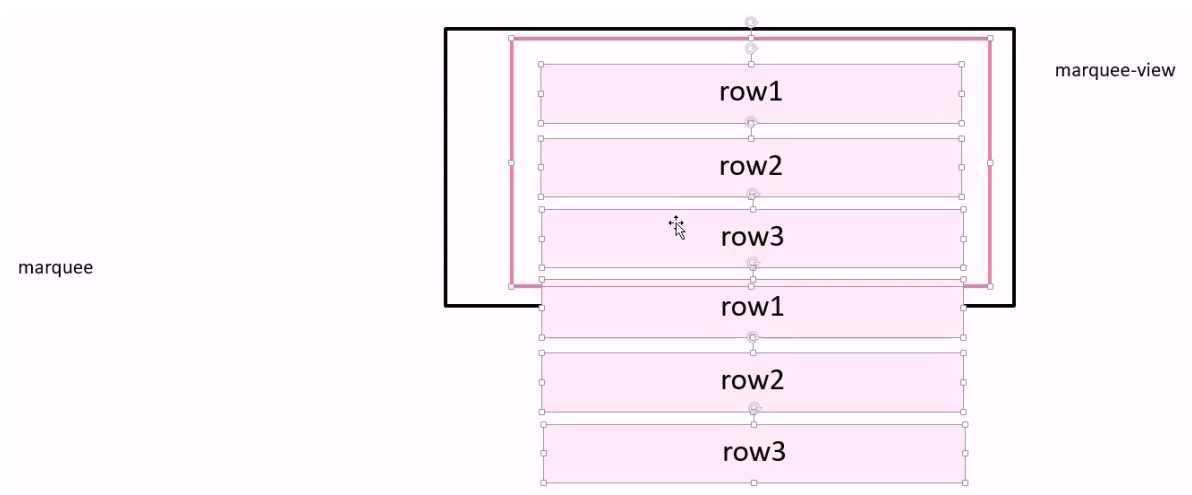
为了防止变量名冲突（变量污染），我们采用立即执行函数：

```
(function() {} )();
```

```
(function() {  
    var num = 10;  
})();  
(function() {  
    var num = 10;  
})();
```

注意多个立即执行函数中间必须加分号隔开。

2.6 无缝滚动原理



步骤:

1. 先克隆marquee里面所有的行 (row)
2. 通过CSS3动画滚动marquee
3. 鼠标经过marquee 就停止动画:

```
animation-play-state: paused;
```

2.7 点位分析模块-使用Echarts图表

1. 先去官网找到雷士的图表引入到页面中
2. 根据需求修改具体的配置

```
option = {
  tooltip: {
    //图形上触发 trigger 触发方式。 非轴图形，使用item的意思是放到数据对应的
    trigger: 'item',
    // 格式化提示内容:
    // a 代表series系列图表名称
    // b 代表series数据名称 data 里面的 name
    // c 代表series数据值 data 里面的value
    // d 代表当前数据/总数居的比例
    formatter: "{a} <br/> {b} : {c} ({d}%)"
  }
  series: [
    // 图表名称
    name: '点位统计',
    //图表类型
    type: 'pie',
    // 南丁格尔玫瑰图 有两个圆半径10% 外圆半径70%
    // 饼形图半径。 可以是像素，也可以是百分比（基于DOM容器大小）第一项是内半径，第二项是
    // 外半径（通过它可以实现饼形图大小）
    radius: ['10%', '70%'],
    // 图表中心位置 left 50% top 50% 距离图表DOM容器
    center: ['50%', '50%']
    // radius 半径模式，另外一种 是 area 面积模式
```

```
roseType: 'radius',  
// 数据集 value 数据的值 name 数据的名称  
data: [  
    {value:10, name:'rose1'},  
    {value:5, name:'rose2'}  
]  
]
```

```
}
```