

数据可视化

Week 8

Intro to D3 and Web technologies

虞思逸

D3

- **一个Javascript库**
- **Data-Driven Documents**
- **将数据与文档（HTML,SVG）联系起来的创建数据可视化图形的库。**
- **优点：**
 - **接口丰富、灵活、直接操作Web文档本身**
 - **有庞大的社区样例支持**
 - **代码简洁**

Web

- **服务器与客户端**（浏览器）之间的对话
- **Web服务器**：例如Apache软件
- **Web客户端**：Chrome Safari ..
- **URL**
 - **通信协议**：HTTP（Hypertext Transfer Protocol, 超文本传输协议）或HTTPS（Secure）
 - **资源所在域名**：bing.com
 - **端口号**：表示要连接到服务器哪个端口（默认80）
 - **其他定位信息**（如请求文件的路径）

Web

- B:尝试连接80端口
- S:同意连接
- B:请求访问页面
- S:发送HTML
- B:根据html引用的其他文件（CSS 图片）依次发送请求
- S:发回文件
- B:渲染（解析HTML结构→根据CSS匹配样式→插入图片→执行JavaScript代码）

HTML

- Hypertext Markup Language 超文本标记语言

- Html5

- 核心功能：标记

- 添加语义结构（层次、关系与含义）
- 通过标签创建元素

标题

段落文本

- 无序列表项
- 无序列表项

段落文本

- 有序列表项
- 有序列表项

1、段落

<p></p>

2、换行

3、标题

<h1>~<h6>

4、无序列表标记

5、有序列表标记

6、列表项标记

 (只能和或结合使用)

7、文本加粗

 或

8、文本倾斜

<i></i> 或

9、水平线

<hr>

10、上下标

11、定义大小

<small></small>

<big></big>

HTML

- 元素自闭合写法

- 属性

<标签名 属性="值"></标签名>

<a href=<http://bing.com>>The Bing website

- Class和id 字母开头!

<p class="sus">Hello, world</p>

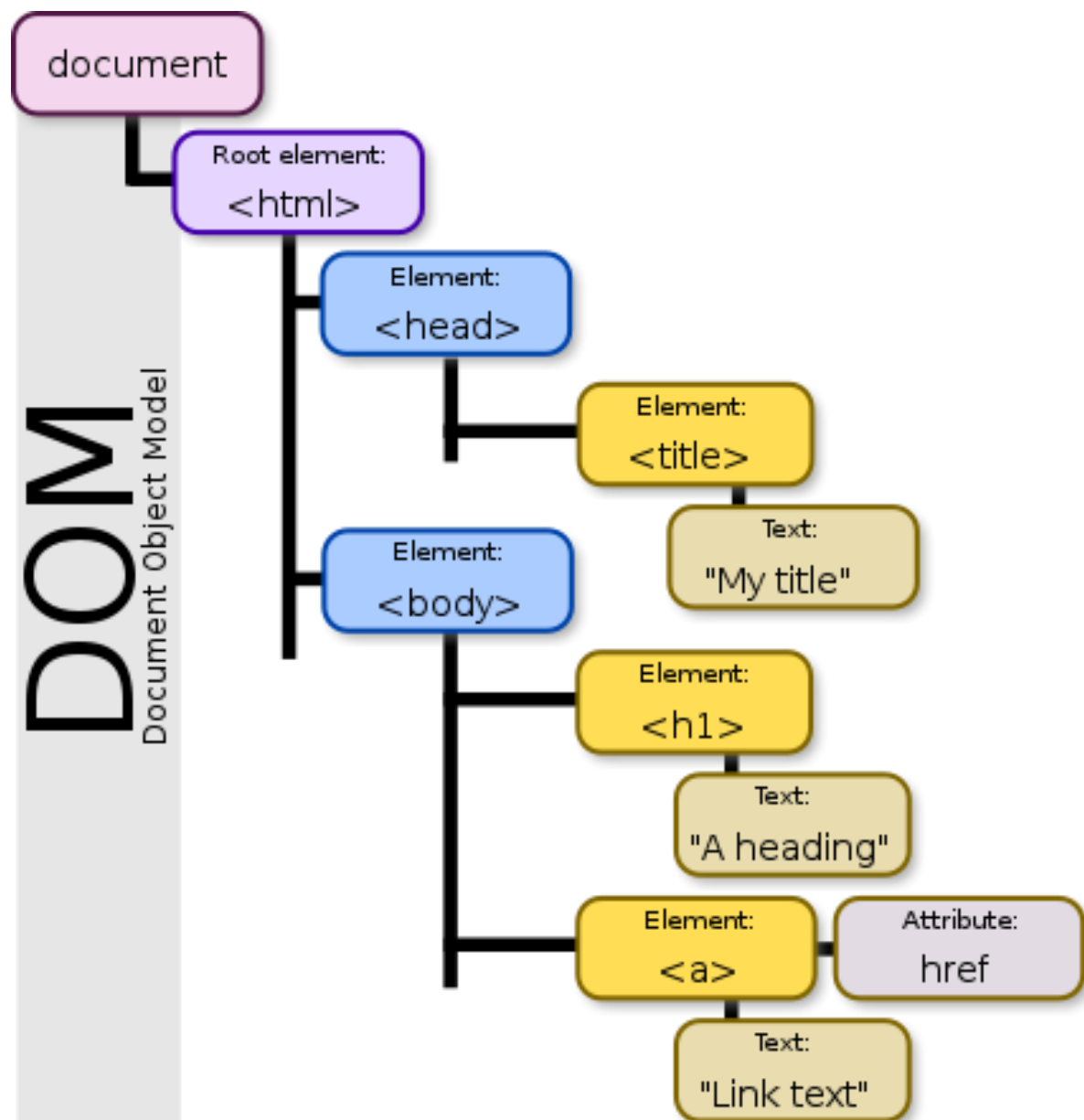
<p class="sus sem">Hello, world again</p>

<div id="content"></div>

DOM

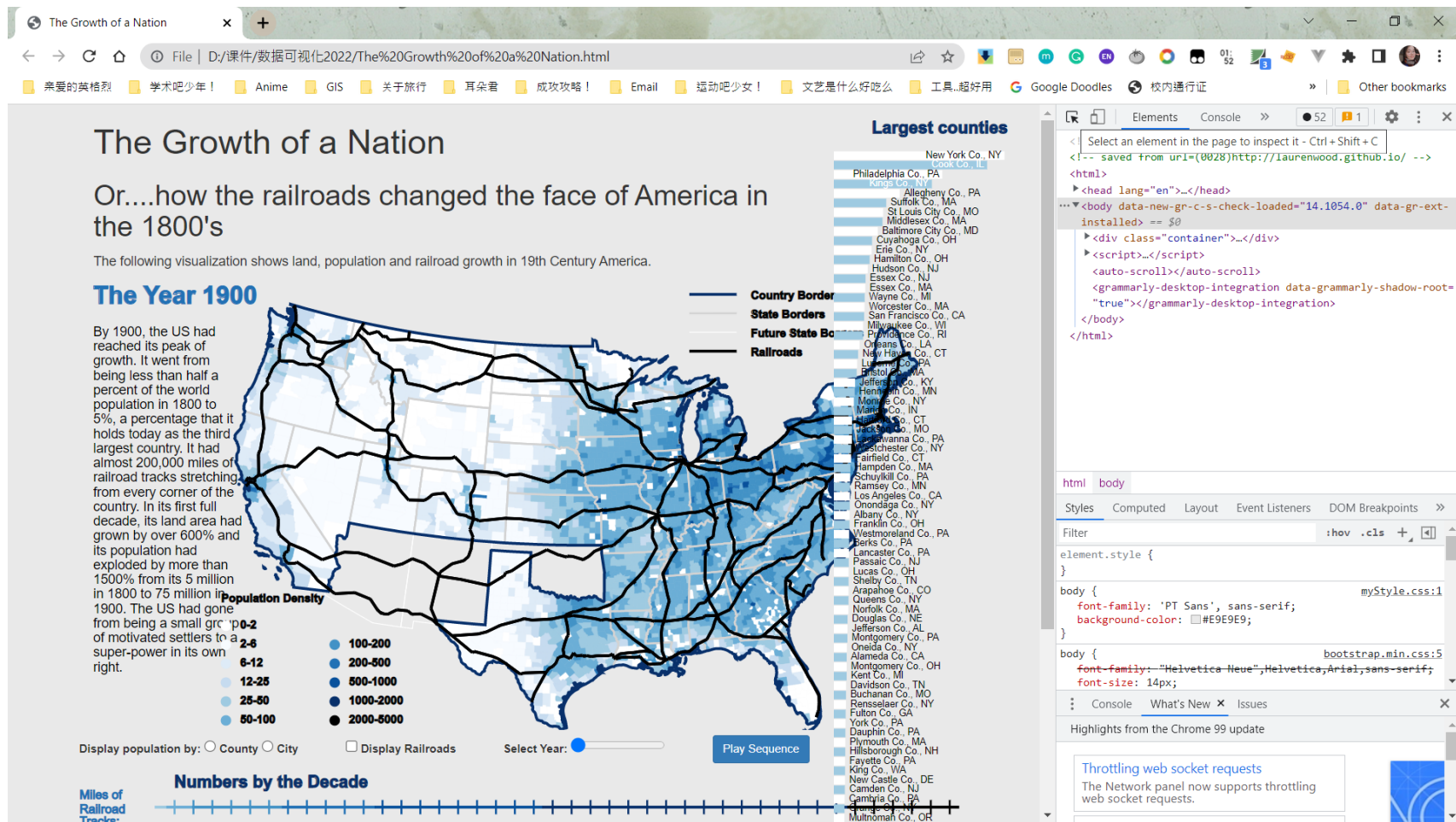
- Document Object Model 文本对象模型
- Html标签的层次结构
- 父元素、子元素
- 兄弟元素
- 祖先元素、后代元素

可视化需要在DOM层次中寻找元素，
添加样式和行为！



开发者工具

- 查看源代码
- 元素审查工具



渲染与盒模型

- **渲染：**

解析HTML并生成DOM→对DOM内容应用视觉规则→将像素绘制到屏幕

- **盒模型：**

浏览器渲染方式：一切都是盒子

块级元素： display:block 自动扩展

行内元素： strong, em, a, span

CSS

- Cascading Style Sheets 层叠样式表
- 一条CSS规则：包含选择符与属性值

选择符：用以表示要引用样式的具体元素

```
选择符 {  
    属性: 值;  
    属性: 值;  
    属性: 值;  
}
```

```
选择符A,  
选择符B,  
选择符C {  
    属性: 值;  
    属性: 值;  
    属性: 值;  
}
```

- 类型选择符

- h1
- p
- strong
- em
- div

- 后代选择符

- h1 em
- div p

- 类选择符(加个点)

- .caption
- .bar.highlight
- .label
- .axis.x
- .axis
- .axis.y

- ID选择符(加个#)

- #header
- #nav
- #export

div.Sidebar

#button.on

CSS

- Cascading Style Sheets 层叠样式表
- 一条CSS规则：包含选择符与属性值

属性和值：

```
选择符 {  
    属性： 值;  
    属性： 值;  
    属性： 值;  
}
```

```
margin: 10px;  
padding: 25px;  
background-color: yellow;
```

```
选择符A,  
选择符B,  
选择符C {  
    属性： 值;  
    属性： 值;  
    属性： 值;  
}
```

颜色的几种格式

- 颜色名： orange
- 十六进制： #3388aa或者#38a
- RGB值： rgb(10,150,20)
- 带透明通道的rgb： rgba(10,150,20,0.5)

CSS

- 引用格式

- 在HTML中嵌入

```
<head>  
  <style type="text/css">  
    body {background-color:yellow;}  
    p {color:blue;}  
  </style>  
</head>
```

- 引用外部样式表

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

- 插入行内样式——直接插入html元素标签

```
<body style="background-color:yellow;">
```

CSS

- **继承、层叠和特制度**

- **继承：**

- 没有指定时继承祖先元素
 - （有些属性不会继承）

- **层叠：**

- Cascading 后定义规则会覆盖前者

- **特制度（specificity）**

- `p.highlight {} > p {}`

- **引用外部样式表**

```
<head>
```

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

```
</head>
```

- **插入行内样式——直接插入html元素标签**

```
<body style="background-color:yellow;">
```

<http://www.csszengarden.com/>

JavaScript

- 一种脚本语言
- 通过操作DOM动态修改页面
 - 浏览器加载JavaScript代码文件
 - 浏览器控制台Console输入JavaScript代码

JavaScript

- **变量**

- `var number = 5;`
- 等号赋值，分号结尾

- **一些数据类型**

- **数组**

- `var numbers=[1,2,3,4,5,6,7];`
- `numbers[2]`
- `var names = ["Mike", "Jack"];`
- `var names =[1.3, 5, 8];`

虽然不推荐但是不同类型可以存在一个数组里

- **对象**

- `var fruit = {`
`kind: "grape",`
`color: "red",`
`quantity:12`
`}`
- `fruit.kind`

- **对象和数组组合**

- `var fruits = [`
`{`
`kind: "apple",`
`quantity:4`
`},`
`{`
`kind: "banana",`
`quantity:5`
`}`
`];`
- `fruits[1].quantity`

JavaScript

- **JSON (JavaScript Object Notation, JavaScript对象表示法)**
- 可用于JavaScript和AJAX请求
- 常见于API (Application Programming Interface)
- 可存在变量中

<pre>var fruit = { kind: "grape", color: "red", quantity:12 }</pre>		<pre>var fruit = { "kind": "grape", "color": "red", "quantity":12 }</pre>
---	--	---

JavaScript

- **GeoJSON**

- 专门为保存地理数据的优化JSON对象
- 保存点、线、多边形以及其他空间特征

- <https://geojson.in/>

经度,纬度 对应的是 y,x!

```
{
  "type": "FeatureCollection",
  "features": [
    { "type": "Feature",
      "properties": {},
      "geometry": {
        "type": "Point",
        "coordinates": [105.380859375, 31.57853542647338]
      }
    }
  ]
}
```

JavaScript

- 数学运算符

- 比较运算符

- ==

- !=

- 逻辑运算符

- && 与

- || 或

- 控制结构

- if(3<5){}

- for(var i=0;i<5;i++)
{console.log(i);}

- 函数

- var calculateGratuity = function(bill) {
return bill * 0.2;
}

- calculateGratuity(38)

- 引用

- 直接放于<script> 标签之间

- 引用外部脚本：在 <script>
标签的 src (source) 属性中设置脚本名称

- <script src="d3.js"></script>

JavaScript

- **函数级作用域**
 - **函数中声明的变量只能在所在函数的内部访问**
- **全局命名空间**
 - **问题：不同.js文件中变量同名冲突**
 - **解决方法：①只在函数中声明变量 ②声明一个全局对象，在全局对象的属性中进行保存**