

D3.js – Hello World

张松海、张少魁、周文洋、蔡韵

数据可视化 – D3.js

清华大学 可视媒体研究中心

Hello World

- 学习D3前，需要了解（本次课的主要内容）：
 - HTML与文档对象模型（DOM）
 - JavaScript
 - Flask（后端）
 - 配置一个Web开发环境
 - 只会 **Cover D3.js**会涉及到的内容
- D3 -> Data-Driven Documents
- Why D3?
 - 接口丰富 - 充足的官方文档和样例支持（社区很庞大 😊）
 - 代码简洁 - D3有特殊的“语法结构”（尽管入门的几天会略微痛苦 😊）

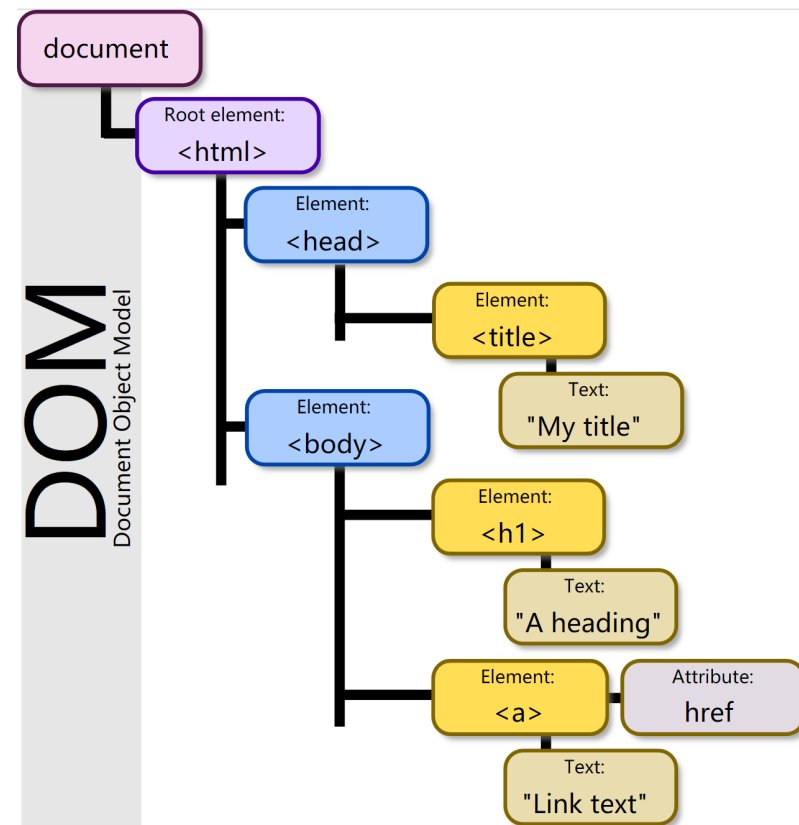
HTML

- 超文本标记语言
- HyperText Markup Language
- HTML不是编程语言!
 - 由多个标签构成的标记语言
 - 通过浏览器来解析
- HTML用来描述我们常见的网页
- HTML包含大量元素（标签）
 - 元素与元素的类别不同，如矩形、直线、文本、圆……
 - 元素包含属性，如位置、大小、色调、文本风格……

```
<!DOCTYPE html>
<html>
  <head>
    <title>Data Visualization!</title>
  </head>
  <body>
    Hello World!
  </body>
</html>
```

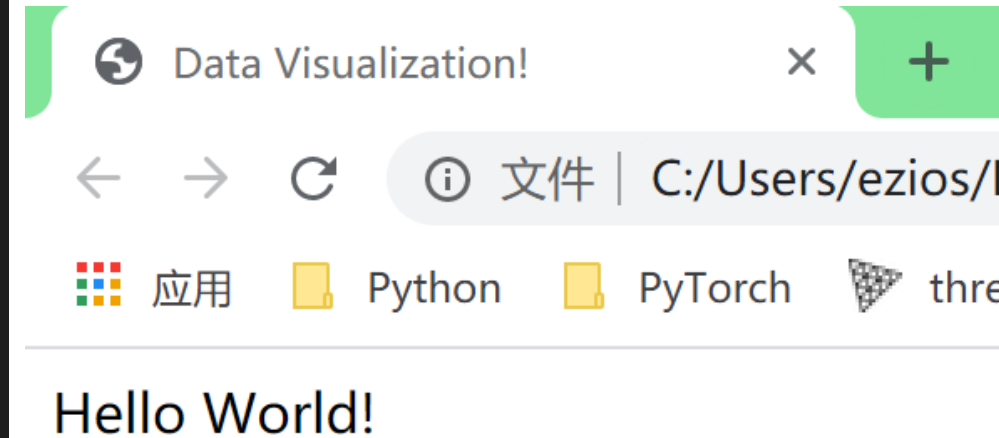
HTML - Tags

- 本课程只需要大家了解以下标签即可!
- `<html>`: 最外层的主标签, 每个HTML文件都需要有!
- `<head>`: 标题内容, 包含HTML的文件链接、标题等
- `<body>`: HTML的主体, 包括各种其中的各种元素
- `<title>`: 标题 (显示在浏览器的标签栏)
- `<script>`: JavaScript脚本或对于脚本的链接
 - D3.js的编程主要写在此标签中
- `<svg>`: 对于D3最为重要的标签, 主要操作的对象 (见下页)
- `<link rel="stylesheet" href="/static/css/style.css">`
- `<!DOCTYPE html>`
- 所有标签都需要以`</tag>`结束



HTML – Hello World

```
<!DOCTYPE html>
<html>
  <head>
    <title>Data Visualization!</title>
  </head>
  <body>
    Hello World!
  </body>
</html>
```

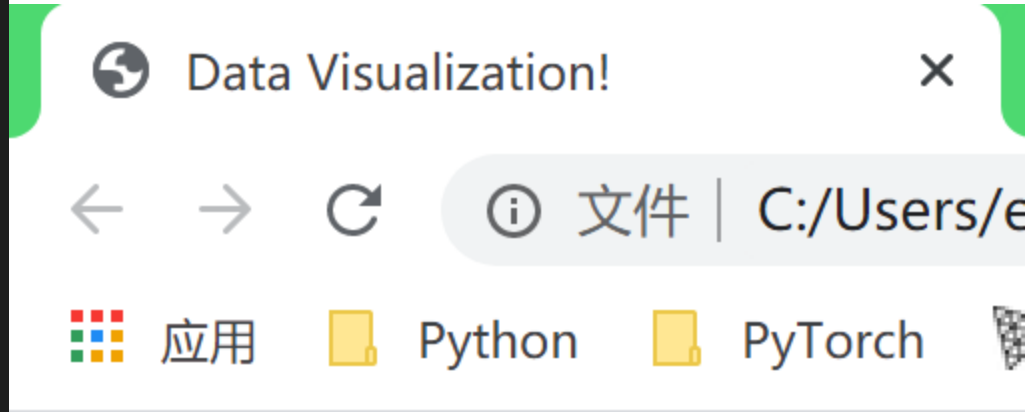


(关于HTML文件)

- HTML文件通常由浏览器解析
- 打开HTML文件
 - 由文本编辑器打开会看到其“源代码”
 - 由浏览器打开会看到其正常的页面
- 使用document, jQuery, D3等库可以动态增删改查页面的元素
 - D3的操作会非常方便!

HTML – Hello World

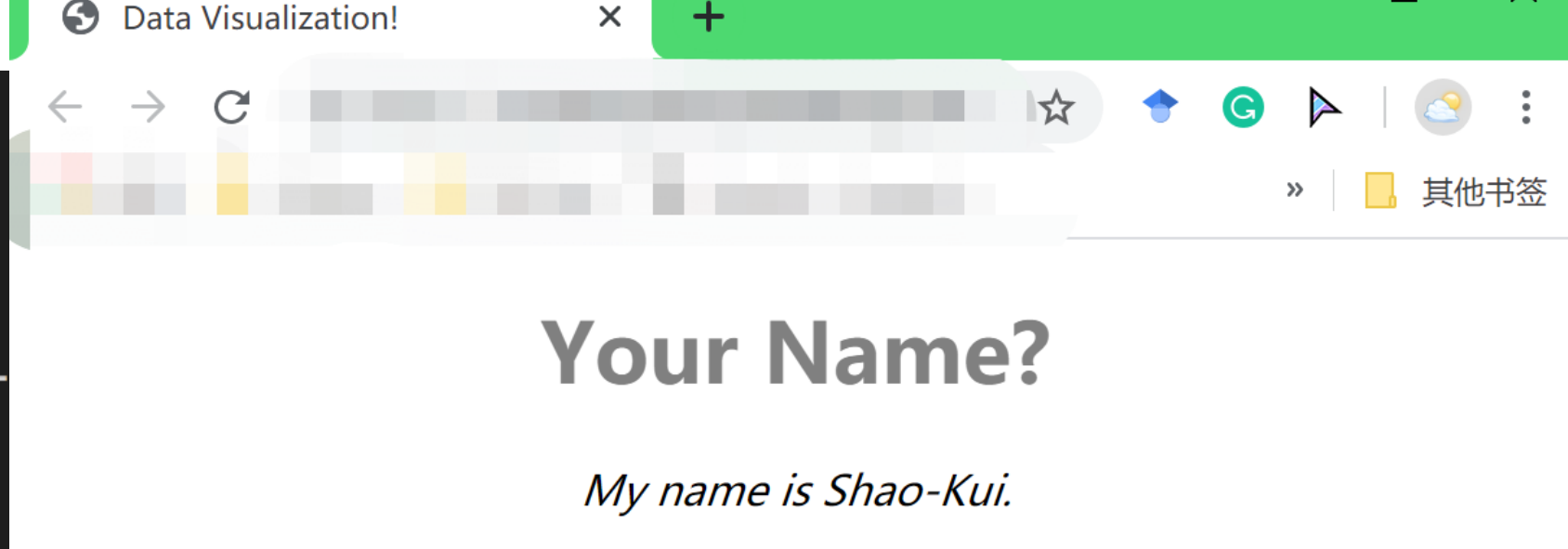
```
<!DOCTYPE html>
<html>
  <head>
    <title>Data Visualization!</title>
  </head>
  <body>
    <h1>Your Name? </h1>
    <p>My name is Shao-Kui. </p>
  </body>
</html>
```



Your Name?

My name is Shao-Kui.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Data Visual
  </head>
  <body>
    <h1 style="text-align:center; color: gray;">
      Your Name?
    </h1>
    <p style="text-align:center; font-style: italic;">
      My name is Shao-Kui.
    </p>
  </body>
</html>
```



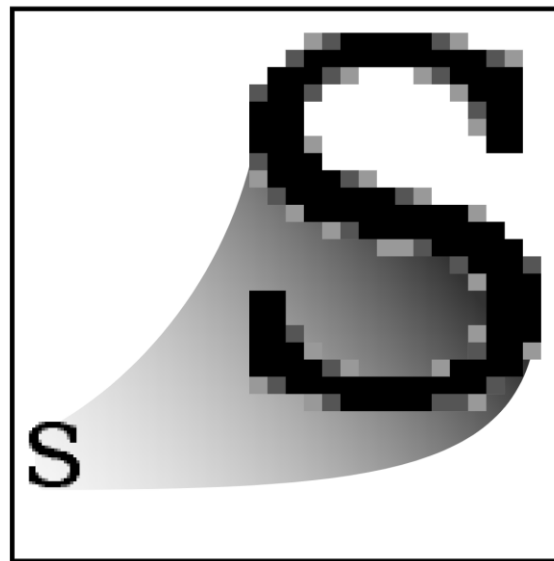
HTML - 导入D3.js

- 使用库，必须先将其导入，如：
 - Python的import，C/C++的include、
 - Java的import、node.js的require……
- 通过Script标签导入
 - 直接通过互联网链接
 - <https://d3js.org/d3.v5.min.js>
 - 通过本地服务器链接（推荐）
 - /static/js/d3.min.js （本地路径会在后续介绍）
 - 通过unpkg链接
 - <https://unpkg.com/browse/d3@5.15.0/dist/d3.js>
 - 为什么推荐本地服务器的链接？

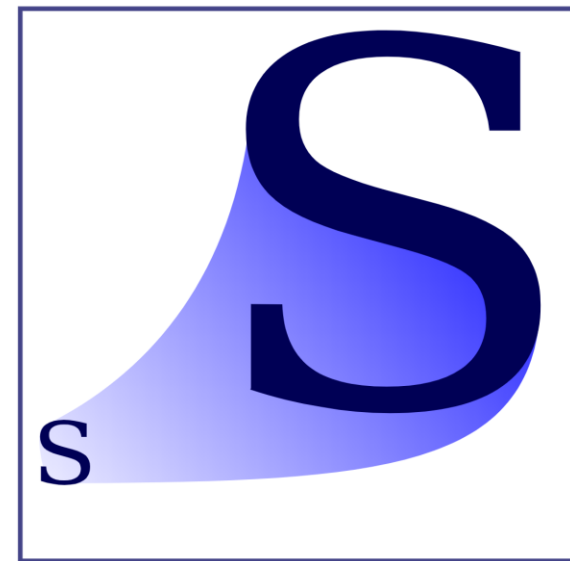
```
<!DOCTYPE html>
<html>
  <head>
    <title>Scatter</title>
    <link rel="stylesheet" href="/static/css/nCov.css">
    <script src="/static/js/d3.min.js"></script>
  </head>
  <body style="text-align: center">
```

SVG – 可缩放矢量模型

- 可缩放矢量图形（英語： Scalable Vector Graphics, SVG）
- SVG是D3.js主要操作的对象
 - `const svg = d3.select('svg');`
 - D3.js获取svg对象
- SVG作为矢量图，不会随着图片的缩放而发生失真；



Raster
.jpeg .gif .png



Vector
.svg

引入SVG

- Head
 - 设置标题
 - 导入d3.js库
- Body
 - 创建SVG
 - 设置修改SVG的脚本（使用D3）
- 制作可视化方案，关于HTML的部分仅需考虑svg即可！

```
<!DOCTYPE html>
<html>
  <head>
    <title>Data Visualization!</title>
    <script src="https://d3js.org/d3.v5.min.js"></script>
  </head>
  <body>
    <svg width="960" height="500" id="mainsvg" class="svgs"></svg>
    <script>
      let mainsvg = d3.select('.svgs');

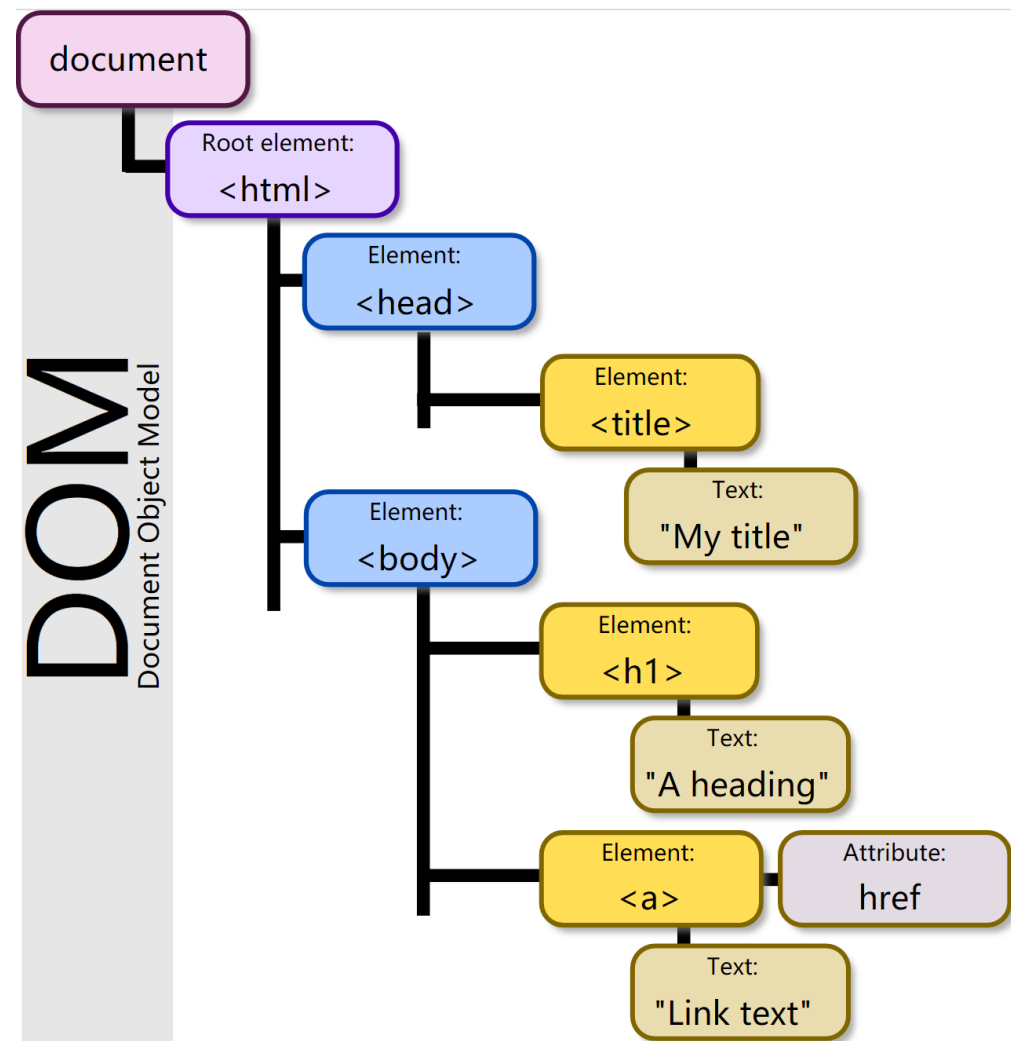
      let maingroup = mainsvg
        .append('g')
        .attr('transform', `translate(${100}, ${100})`);

      let circle = maingroup
        .append('circle')
        .attr('stroke', 'black')
        .attr('r', '66')
        .attr('fill', 'yellow');
    </script>
  </body>
</html>
```

- Code: <https://github.com/Shao-Kui/D3.js-Demos/blob/master/main.html>

HTML – 文档对象模型

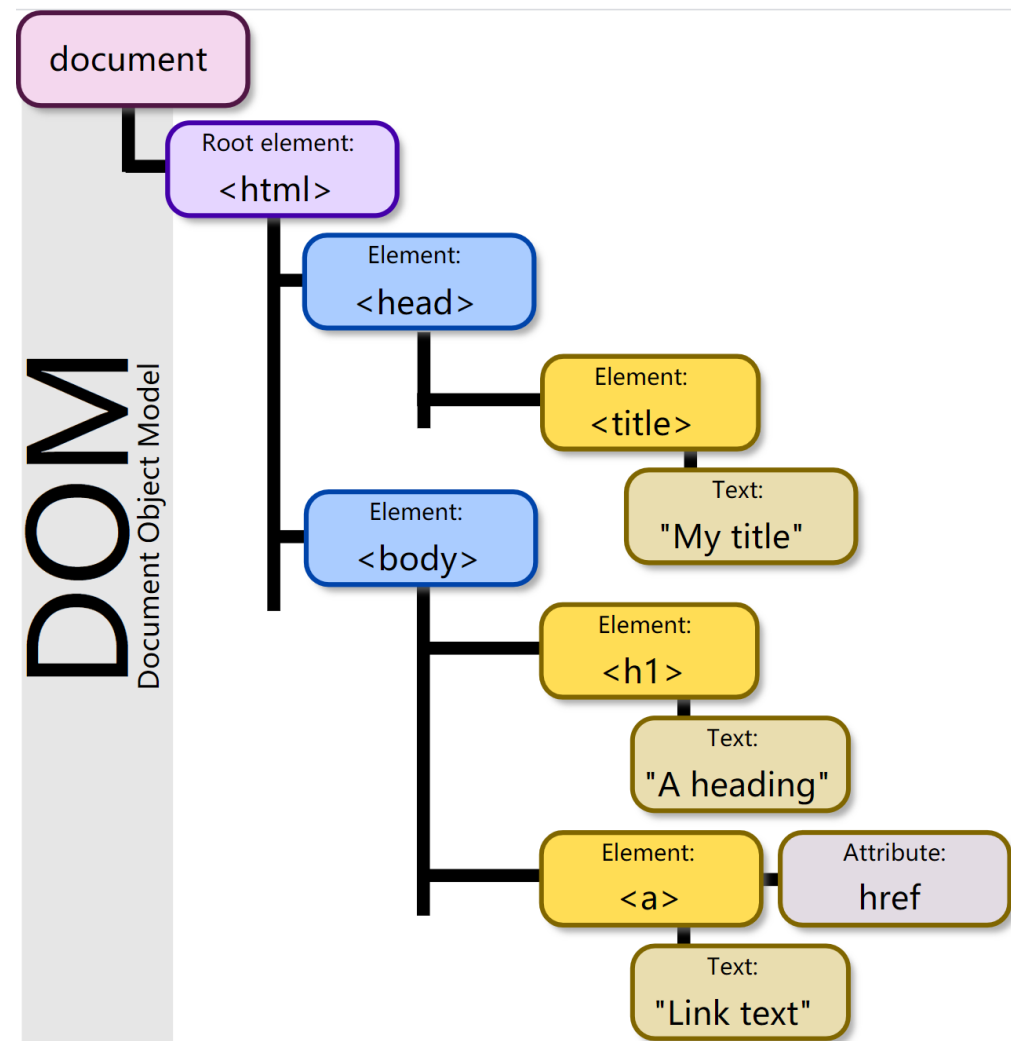
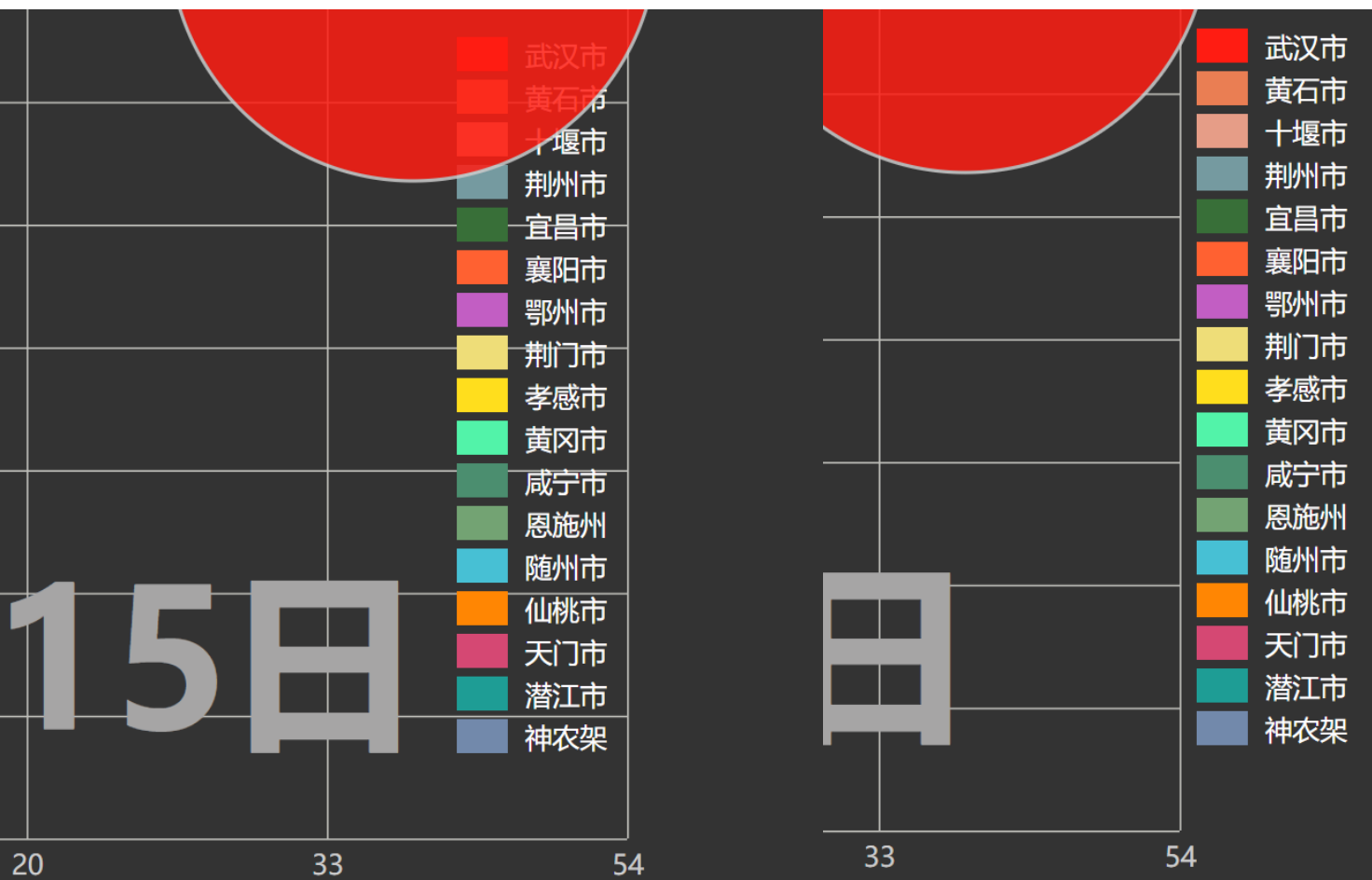
- HTML -> DOM
- DOM -> Document Object Model
- 对于根节点的操作会影响到子节点;
- 最常用的父节点<svg>中的<g>
 - Axis可封装成一个group
 - Legend (图例) 可封装成一个group
- Data-Driven **Document**
- **Document** Object Model



<https://en.wikipedia.org/wiki/File:DOM-model.svg>

HTML – 文档对象模型

- 对于根节点的操作会影响到子节点;



<https://en.wikipedia.org/wiki/File:DOM-model.svg>

JavaScript

- Web开发语言
 - 同时支持客户端和服务端，本课程主要集中在客户端
 - 前端通常由浏览器（如chrome）解析
- 解释型的编程语言
 - 不需要编译
 - E.g., C, C++需要经过编译才可执行
- JavaScript 是可插入 HTML 页面的编程代码
 - D3.js -> JavaScript
- JavaScript vs. Java
 - 两者没有直接关系！
- JavaScript vs. C(++)
 - 二者有很多类似，if条件语句、switch语句、while循环、do-while循环等

JavaScript – 主要语法特性

- 变量声明不需指定类型
 - let, var, const
- 运算操作基本等同于C、C++、JAVA等语言
 - +, -, *, /, %,
- 函数定义同样不需要指定类型
 - function abc(a){ return a + 5; }
 - datum => datum.value;
 - const p = function(a, b) { return a + b; }
 - let myFunction = () => a + b

JavaScript – 函数式编程

- 一个变量可以是一个函数
 - `const myFunction = function(a, b) { return a + b; }`
 - 类似与C/C++的函数指针
- 回调 (Callback) :
 - JavaScript脚本中常见把函数作为变量输入
 - 用于实现异步编程
 - `setTimeout(function () {`
 - `console.log('hello world! ');`
 - `}, 1000);`
- 在D3中有大量的函数式编程，请大家务必熟悉这种形式！
 - 如：为每个数据点指定不同的颜色

JavaScript – D3中的常用接口

- `console.log('hello world! ');`
- 数组 `a = [1, 2, 3]`
- 对象 `a = {name: 'Shao-Kui', age: 24.3, lab: 'cscg'}`
 - D3数据可视化中常见对象数组，如：
 - `a = [{name: 'Shao-Kui', age: 24.3, dept: 'cs'},`
 - `{name: 'Wen-Yang', age: 23, dept: 'cs'},`
 - `{name: 'Yun', age: 18, dept: 'art'}]`
- 数组的排序 `a.sort()`
 - 可加入回调函数来替代缺省的排序方案，如对日期排序
 - `a.sort(function(a,b){ return new Date(b.date) - new Date(a.date); })`
- 数组的查询 `a.find(d => d.name === 'Wen-Yang')`
- 把字符串转换成数值： `+('3.14')`
- D3.js经常读取CSV文件，届时会涉及大量的数组、对象的操作！

JavaScript in D3.js

- 在HTML中查找并获取SVG
- 在SVG中加入组 (Group)
<g>
 - 设置组的平移为向下、向右个100像素
- 在主要组中加入一个圆
 - 圆的边框为黑色
 - 圆的半径为66像素
 - 圆的填充颜色为黄色

```
<!DOCTYPE html>
<html>
  <head>
    <title>Data Visualization!</title>
    <script src="https://d3js.org/d3.v5.min.js"></script>
  </head>
  <body>
    <svg width="960" height="500" id="mainsvg" class="svgs"></svg>
    <script>
      let mainsvg = d3.select('.svgs');

      let maingroup = mainsvg
        .append('g')
        .attr('transform', `translate(${100}, ${100})`);

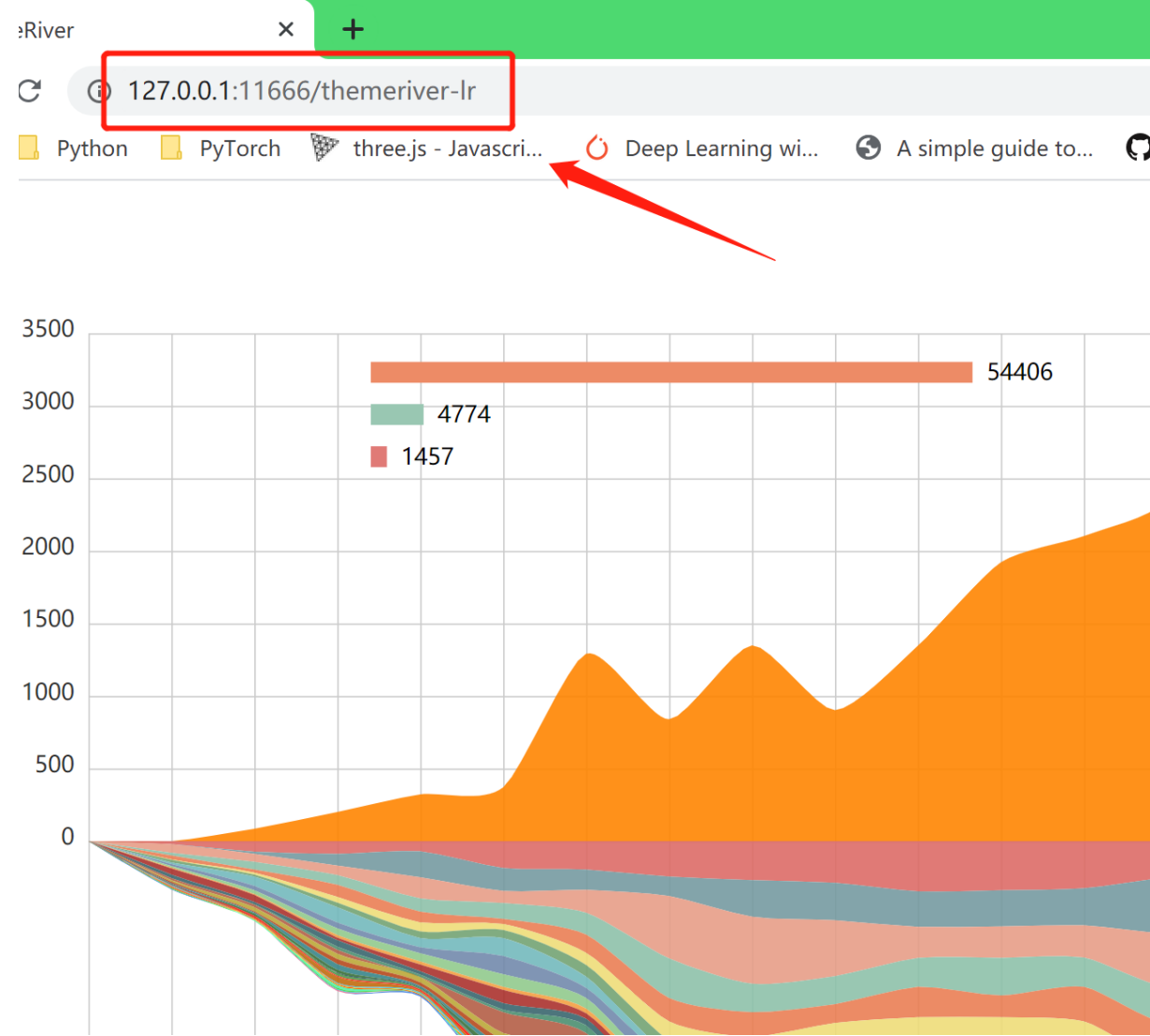
      let circle = maingroup
        .append('circle')
        .attr('stroke', 'black')
        .attr('r', '66')
        .attr('fill', 'yellow');
    </script>
  </body>
</html>
```

Flask – Why?

- 为什么需要一个后台（服务器）？
 - CORS (Cross Origin Resource Sharing)
- 可否用一个完整的.html文件编程
 - 完全可以，但数据处理、链接d3.js库、链接.js、链接.css文件会非常麻烦！
- 对于没有WEB开发经验的同学，本课程对D3.js和可视化以外的工程不做要求：
 - 提供默认调通的客户端-服务器框架
 - 仅需要了解后续的内容，对服务器进行配置并运行即可
 - 推荐有web经验的同学搭建自己熟悉的环境

Web编程 – 请求

- 一条请求
- `http://127.0.0.1:11666/scatter-animation`
- IP:
 - 127.0.0.1
- 端口:
 - 11666
- 路径 (route) :
 - `/scatter-animation`
- 每个请求可以返回服务器端请求对应的结果， 如一个.html页面



Flask – 安装与使用

- 安装Python（课上不介绍）
 - 请务必将python的环境变量配置好
 - <https://www.python.org/>
- 打开命令行输入：
 - `pip install flask flask_cors`
- 创建一个最基础的Flask程序或使用如下代码（见下页）：

d3demo > static

- ☐ 名称
- themeriver-lr.html
 - themeriver.html
 - scatternxt.html
 - Scatter-Animation.html
 - scatter.html
 - renderearth.html
 - pinkheart.ico
 - nCovnxt.html
 - ☒ nCov.html
 - loveheart.ico
 - index.html
 - color-gradient.html
 - barcscginteraction.html
 - barchart.html
 - baraxes.html
 - js
 - icon
 - html-tutorial
 - data
 - css

- d3demo >
- ☐ 名称
- .git
 - deprecated
 - static
 - LICENSE
 - main.html
 - main.py
 - README.md

```
import flask
from flask_cors import CORS
from flask import Flask

app = Flask(__name__)
app.config['TEMPLATES_AUTO_RELOAD'] = True
CORS(app)

@app.route('/')
def index():
    return flask.send_from_directory('static', 'index.html')

@app.route('/nCov')
def nCov():
    return flask.send_from_directory('static', 'nCov.html')

@app.route('/barchart')
def barchart():
    return flask.send_from_directory('static', 'barchart.html')
```

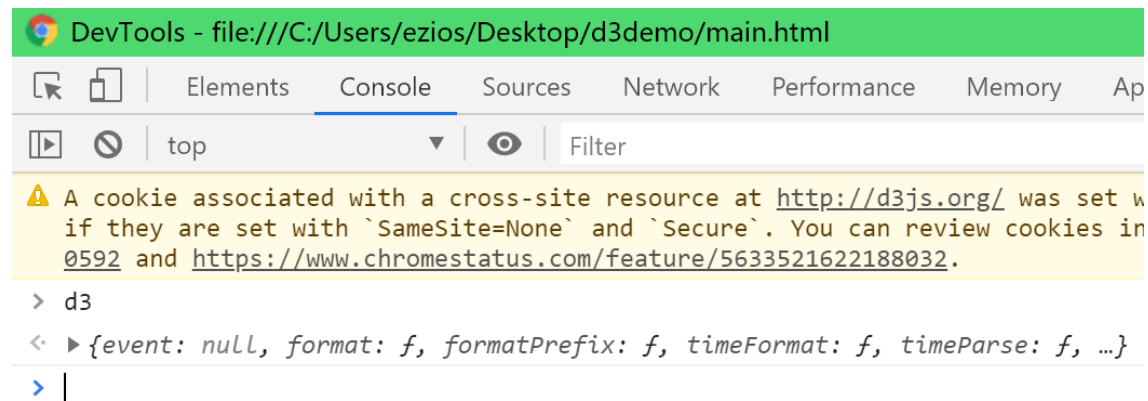
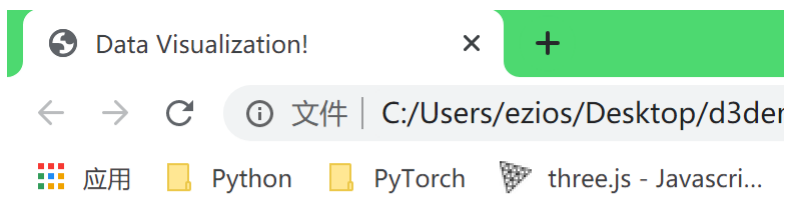
本课程对于D3的支持

- 已经配置好简单的服务器环境
- 下载（包括课上的一些可视化效果的可运行源代码）
 - <https://github.com/Shao-Kui/D3.js-Demos>
 - 网络学堂
- 尽可能让大家关注在可视化编程， e.g., D3.js
- 对于熟悉WEB开发的同学更推荐搭建自己的熟悉的开发环境！

Hello World - End

- 推荐使用VSCode作为代码编辑器
 - Sublime Text, Atom也可以但前者收费，后者启动略慢且功能略少
- 推荐使用Chrome浏览器
- 对于HTML、JavaScript仍较陌生的同学请利用课下时间学习，任何课程相关困难可给助教发邮件
- 如何确认是否成功导入了D3?（见下页）
 - 在Chrome中呼出开发者，直接在命令行键入“d3”并回车看是否报错
- 下次课会正式进入“使用D3.js”操作SVG中的元素
- 请大家务必下次课前搭建好D3的环境，由任何问题请联系助教!
 - zhangsk18@mails.tsinghua.edu.cn

Hello World - End



课程参考资源

- 教学参考书：数据可视化（第二版），陈为 沈则潜 陶煜波 等，电子工业出版社，
- 2019教学主要参考资料：IEEE VIS会议论文和IEEE TVCG期刊论文集
- D3编程参考资料：
 - <https://d3js.org/> （官方网站，包括文档、样例）
 - <https://developer.mozilla.org/zh-CN/docs/Web/SVG/Attribute> （SVG属性表）
 - <https://github.com/d3/d3/wiki/Gallery> （官方样例的仓库）
 - https://github.com/xswei/d3js_doc （一个d3.js的资源汇总 包括示例、书籍、API文档等）