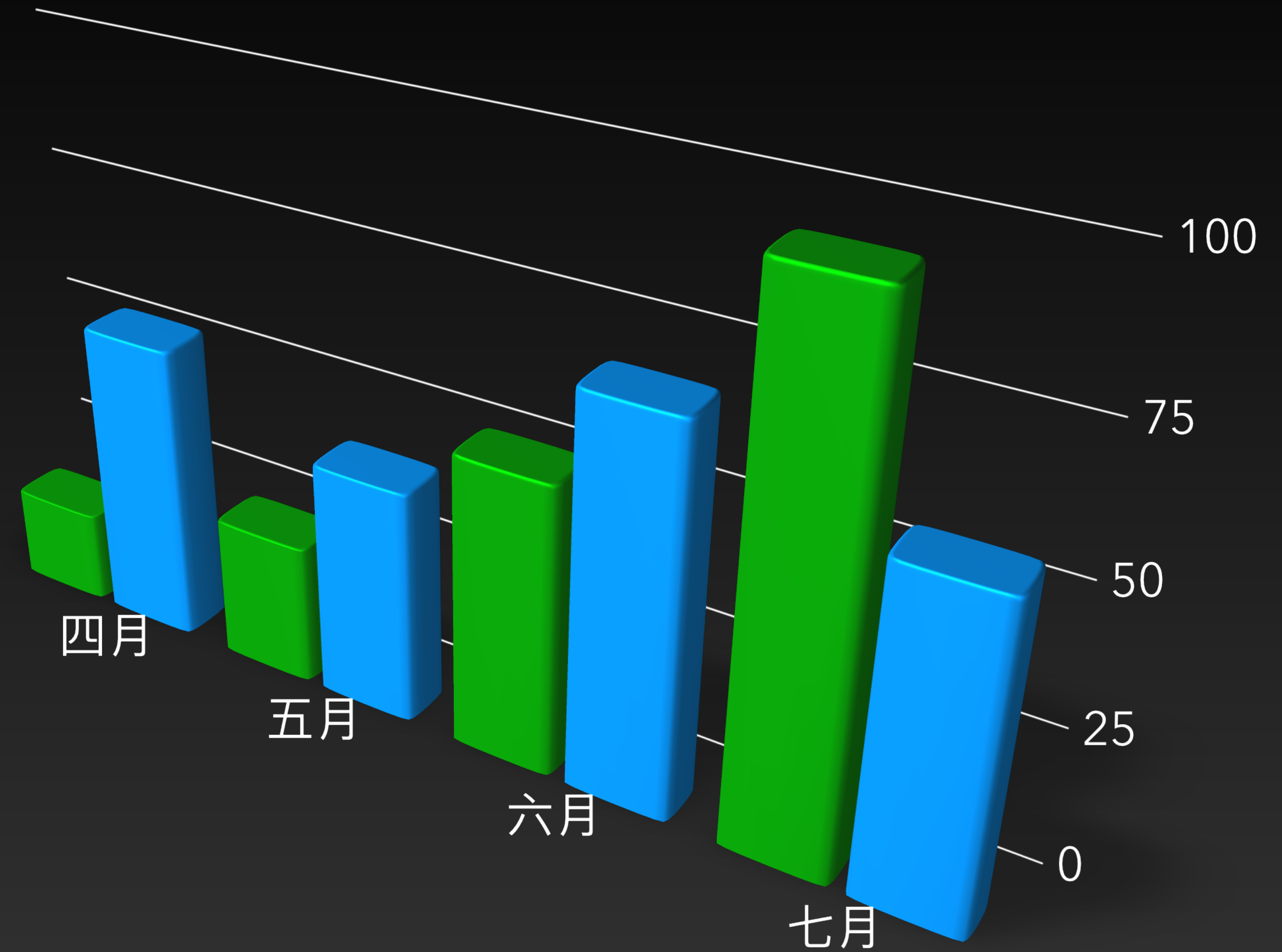


# 可视化入门

初识可视化体系

# 主题内容

- 什么是可视化
- 为什么要学习可视化
- 如何学习可视化
- 可视化的未来



# 什么是可视化

可视化（Visualization）是利用计算机图形学和图像处理技术，将数据转换成图形或图像在屏幕上显示出来，再进行交互处理的理论、方法和技术。

可视化的概念是相对数据来说的，也叫数据可视化。通过对数据的收集和分析，过滤出具有客观特征的数据，以一系列的理论和技術，映射成人类视觉感官上容易理解的各种图形和图像，传达出数据背后的意义，就是数据可视化。

## 可视化的目的

数据可视化是为了清晰有效地传达与沟通信息。因为人类的大脑在对视觉信息的处理上，图形或者图像的形式比纯文字或数字要更容易理解。所以，在进行可视化设计的时候，要在视觉和易理解上寻找一个平衡点，切忌单纯为了视觉效果而设计。

为什么要学习可视化

1. 使用在线脑图，关系图或流程图的时候，有没有好奇是如何实现的？
2. 业务中碰到简单的图可视化开发需求时，却手足无措，不知从何下手？
3. 稍微了解过一点GIS开发，偶然碰到酷炫的GIS效果，却不清楚了解过的几款GIS库能否完成类似的效果？



## 学习可视化的目的

- 了解应用场景，助力业务开展

可视化作为前端的旁系技术，拥有广泛的应用场景。比如，电商大屏，指挥调度大屏，视频监控，全球疫情态势，Web GIS。

- 拓展技术认知，打破技术边界

学习可视化背后的图形学和视觉呈现原理，对于理解CSS的原理和实现其他技术领域的需求有很大的帮助。比如AR试衣，VR看房，3D游戏，数字孪生。

## 培训目标

- 基本了解可视化
- 遇到可视化需求，可以做合理的技术选型
- 能实现简单的可视化需求

# 如何学习可视化

## 前端与可视化

传统的Web前端，是用HTML，CSS，JS为用户做内容呈现和动作交互的。而可视化偏结构化数据和视觉细节，注重于将结构化的数据，映射成各式各样的图形。

- 共同点

1. 都是做视觉呈现和动作交互。所以，有些理论和方法是相通的。
2. 在各自的领域，都有对应的一系列技术和工具，但是都离不开JS

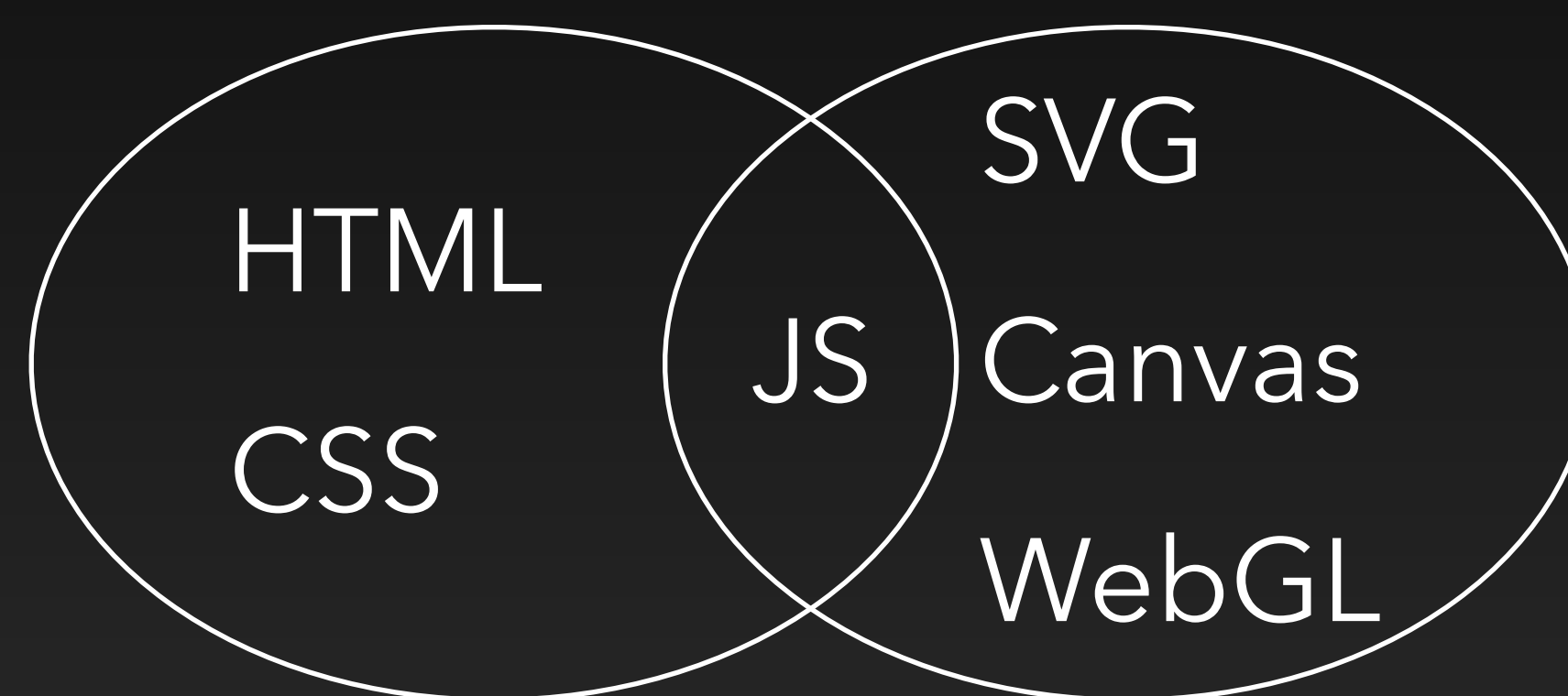
- 不同点

1. Web前端偏向文本内容和媒体信息；可视化偏向于结构化数据和图形渲染
2. 各自领域的技术栈不同；前者主要以HTML, CSS, JS为基础及其上层的技术组成，后者是以SVG, Canvas, WebGL为基础及其上层的技术组成。



可视化不一定局限于使用Web技术实现，但是相对来说，全球的Web应用可能最多，后面所述皆属于Web可视化的范畴，但是可视化的理论，方法和技术都是相通的。

## 前端基础   可视化基础



# 可视化

SVG

Canvas

WebGL

SVG.js

Fabric.js

ThreeJS

BabylonJS

D3.js

ECharts

AntV

Highcharts

Chart.js

Mapbox

Leaflet

Deck.gl

CesiumJS



## 可视化原始的实现方式

- HTML+CSS+JS
- SVG
- Canvas
- WebGL

## HTML+CSS+JS

- 使用场景：可视化需求简单，图形简单
- 优点：不必引入第三方库
- 缺点：
  1. 画不规则图形曲折，麻烦
  2. 实现过程曲折，麻烦
  3. 元素数量过多时，有性能问题

# SVG

- 使用场景：图形元素数量少的时候使用
- 优点：
  1. 画规则或不规则的图形有现成的标签，比较直观
  2. 矢量，不失真
  3. 图形需要的SVG个数少的情况下，性能较好
  4. 交互事件的处理非常容易
- 缺点：元素数量过多时，有性能问题

# Canvas

- 使用场景：画布尺寸不是非常大的时候使用
- 优点：不需要像前两种方式一样经过浏览器的解析，构建DOM树，构建渲染树，计算布局等步骤，单从绘制速度来说，比前两者都快
- 缺点：
  1. 不好使用，不比SVG有现成的图形标签
  2. 不好控制内部图形元素
  3. 不好做用户交互

# WebGL

- 使用场景：
  1. 遇到严重的性能瓶颈，需要使用GPU并行渲染的能力
  2. 需要绘制3D图形
- 优点：更底层的绘制系统，拥有高性能的绘制能力，能方便绘制3D图形
- 缺点：比前几种使用起来都要复杂，有很高的门槛

## 渲染库

- SVG.js
- Fabric.js
- Three.js
- D3.js (数据驱动库)

## 深入学习可视化

其实可视化涵盖的内容，不止上面提到的基本的技术。它还包括，数据部分（数据收集，数据分析，数据治理，数据管理，数据挖掘），计算机图形学，图像处理，计算机视觉，视觉设计，计算机辅助设计。但是作为一名技术人员，最主要的是要掌握以下几项：

- 四种基本的可视化实现方式
- 可视化性能优化
- 计算机图形学
- 开源图表库和D3.js的设计思想

# 可视化的未来



## 数字孪生

可视化归根究底，还是计算机视觉呈现的范畴，无论未来如何发展，还是依赖于计算机图形学，计算机图像处理等核心技术的支撑，从大的角度来看，或许可视化并没有那么狭隘的定义，其不过是视觉呈现和用户交互的领域，与时下比较热门的数字孪生的概念殊途同归。

数字孪生（IoT+AI）更像是科技社会的未来。

# 致谢

vviz.