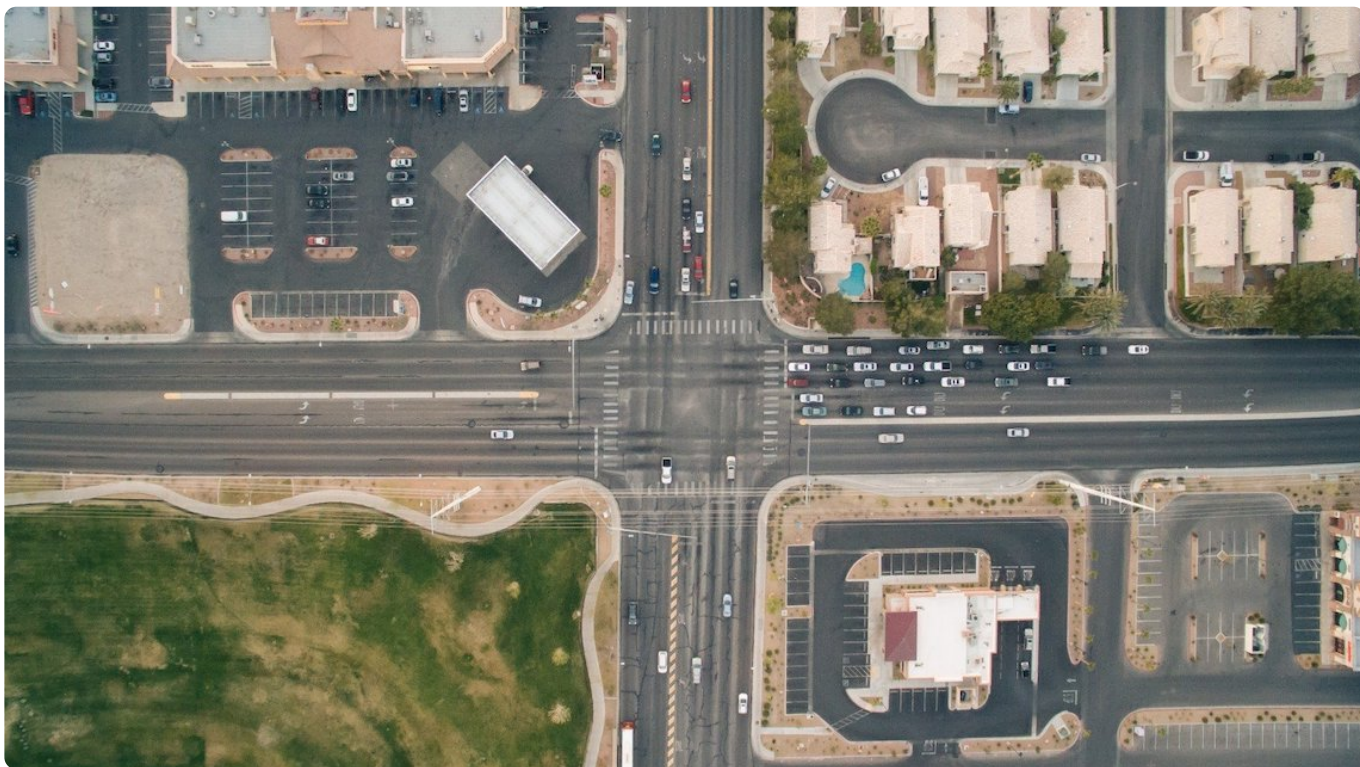


预习 | Web前端与可视化到底有什么区别？

2020-06-22 月影

跟月影学可视化

[进入课程 >](#)



讲述：月影

时长 11:27 大小 10.49M



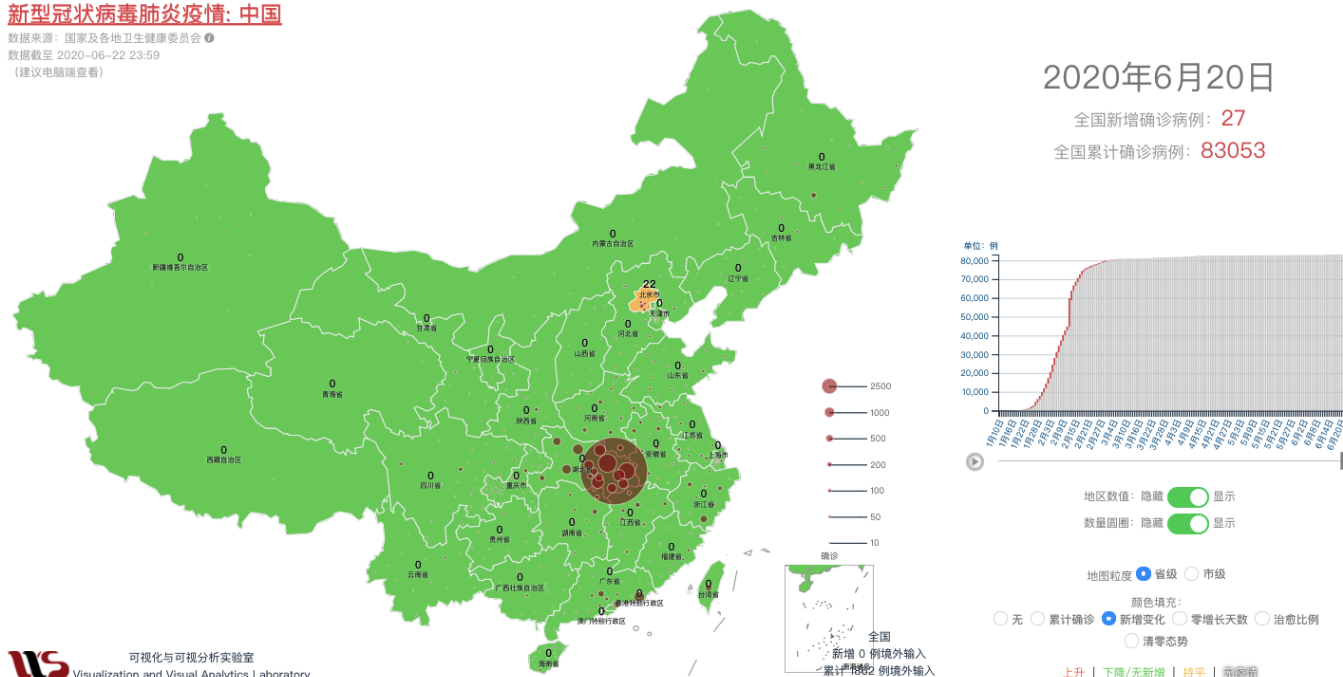
你好，我是月影。在课程正式开始之前，我想先和你聊聊可视化是什么，Web 前端和可视化的区别，以及可视化领域中非常重要的工具。了解了这些基本的东西，对我们的学习是非常有帮助的。

前段时间因为新型冠状病毒，我们每天都在关注疫情信息。不知道你有没有注意过这些疫情信息的展示方式。



新型冠状病毒肺炎疫情: 中国

数据来源: 国家及各地卫生健康委员会
数据截至 2020-06-22 23:59
(建议电脑端查看)



来源: 北京大学可视化与可视分析实验室

我们看到的疫情图大概都会通过上面这个信息图的样子展示出来。这种信息图与普通的网页看上去差别非常大，我们没办法用传统的 Web 开发技术实现这样的“网页”。没错，这是一个与传统 Web 开发完全不一样的领域，叫做**数据可视化**（Data Visualization）。

其实，除了“疫情地图”之外，我们平时见到的很多东西，都是通过数据可视化来实现的。比如，每年淘宝“双十一”的可视化数据大屏、各种平台的年度账单等等。

那你可能要问了，可视化到底是什么呢？

如果要给可视化下一个定义，我觉得应该是这样的：**可视化是将数据组织成易于为人所理解和认知的结构，然后用图形的方式形象地呈现出来的理论、方法和技术。**实现可视化有两个关键要素，一个是数据，另一个是图形。如果要考虑在计算机上呈现，那还要加上交互。

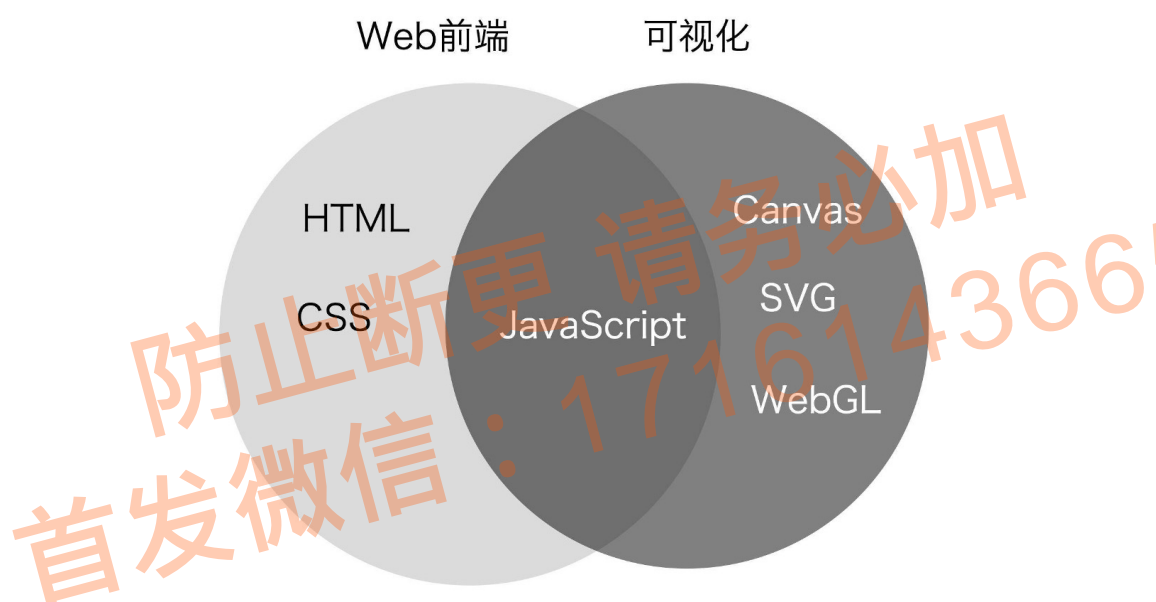
Web 前端与可视化有什么区别？

据我所知，很多同学在工作中因为产品需求要呈现图表，而不知不觉从 Web 开发进入了可视化领域。但因为不了解它们之间的核心区别，或多或少都会遇到一些棘手的问题。

那可视化和 Web 前端究竟有什么区别呢？

我们先从工作内容来看，前端工程师主要负责处理内容呈现和用户交互。可视化的数据呈现，尤其是在 Web 端的呈现，也属于前端工程师的范畴。但是，与传统的前端开发相比，可视化开发的工具、思路和方法，又和 Web 有着比较大的区别。

首先是技术栈的不同。Web 开发主要以 HTML 来描述结构，以 CSS 来描述表现，以 JavaScript 来描述行为。而可视化则较少涉及 HTML 和 CSS，它更多地要同浏览器的 Canvas、SVG、WebGL 等其他图形 API 打交道。这是因为，Web 开发以呈现块状内容为主，所以 HTML 是更合适的技术。而可视化开发因为需要呈现各种各样的形状、结构，所以，形状更丰富的 SVG 以及更底层的 Canvas2D 和 WebGL 就是更合适的技术了。



Web前端和可视化基础技术栈

其次，Web 开发着重于处理普通的文本和多媒体信息，渲染普通的、易于阅读的文本和多媒体内容，而可视化开发则着重于处理结构化数据，渲染各种相对复杂的图表和图形元素。两者针对的信息特征和对图形渲染的要求有所不同，所以，在**细节处理**上也有比较大的差异。

这么说比较抽象，我来举个例子。我们在使用 ECharts 这样现成的图标库开发图表的时候，往往会遇到一些产品需求不能很好被满足的情况。比如，如果想要把图表的默认布局方式从 absolute 改变成其他方式、把坐标轴指示线的样式换成虚线、把配色换成线性渐变，或者想要修改地图中某个点的点击范围。虽然有的问题，通过修改参数配置（ECharts 的每

种图表可能有数十种不同的配置项) 确实能解决, 但还是有一些问题必须要修改库的源码才能解决。

因此, 像 ECharts 等现成的图表库, 尽管功能强大、配置复杂, 但是在**样式细节控制**上, 仍然无法做到如 CSS 那般灵活。

除此之外, 可视化要处理更多**偏视觉方面的细节信息**, 尤其是在要呈现的数据细节比较丰富的时候, 可能要精确地呈现大小、距离、角度、高度、光线、阴影等等。这些细节的处理, 都需要我们对图形绘制有更加精确的控制。因此, 我们需要较深入地掌握图形学理论知识, 了解并熟悉专业的图形库和绘图的工具。简而言之, 就是 Web 开发的前端主要还是关注内容和样式, 图形的渲染和绘制是由浏览器底层来完成的, 而可视化前端则可能要深入底层渲染层, 去真正地控制图形的绘制和细节的呈现。

尽管 Web 前端与可视化在工具、思路和方法上有一些区别, 但前端开发的同学想要学习可视化, 门槛并不高, 主要原因有三点。

其一, **可视化与 Web 前端一样, 最终都是以图像呈现在浏览器上, 因此有许多通用的方法论**。比如, 两者都要涉及 DOM、都要处理浏览器事件、都采用同样的颜色表达方式和同样的资源请求方法等等。

其二, **二者都使用 JavaScript, 而且都是浏览器端的 JavaScript**。所以, 就 JavaScript 的应用而言, 这一块差别并不大。不过, 可视化应用面对的数据和渲染的图形元素都比传统的 Web 应用更复杂一些, 所以也就更加依赖 JavaScript 一些。

其三, Web 前端领域有许多成熟的工具, 包括响应式框架 (比如, 三大框架 Vue、React、Angular)、设计系统 (比如 Ant Design、ElementUI)、函数库 (比如, Underscore、Lodash) 等等。**与 Web 前端一样, 可视化领域也有丰富的 JavaScript 工具和活跃的生态**, 通常这些 “开箱即用” 的工具, 就能够帮助我们完成可视化开发。

因此, 只要善于使用这些成熟的可视化工具, 我们就能高效率地完成手头的工作。所以, 这些工具对可视化的学习和应用来说非常重要。那接下来, 我就带你了解一下, 可视化领域中都有哪些常用的工具。

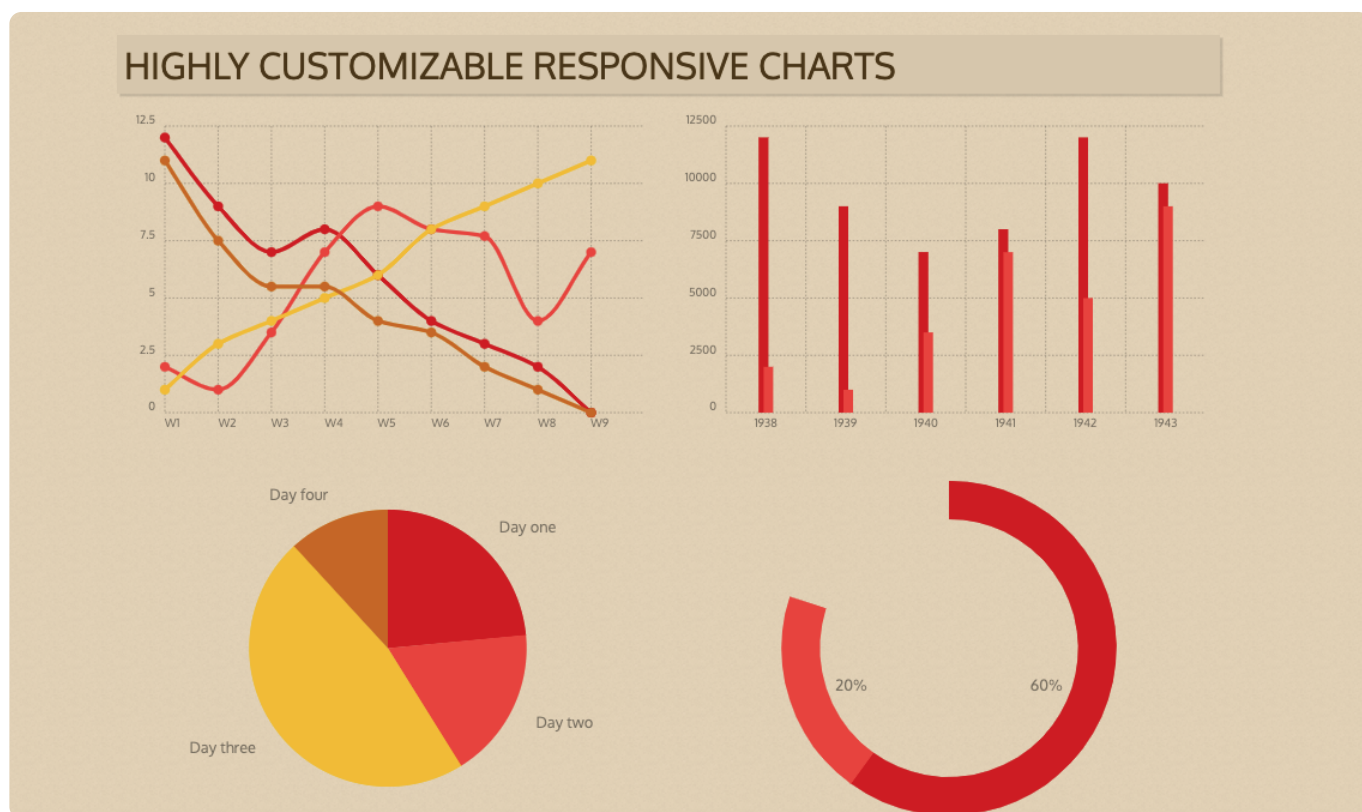
可视化领域的工具

可视化领域经过多年的发展，有非常多丰富的工具。我们可以把这些工具大体上分为四类，分别是：专业呈现各种类型图表的**图表库**；专业处理地图、地理位置的可视化**地理库**；专业处理视觉呈现的**渲染库**；以及处理数据的**数据驱动框架**。

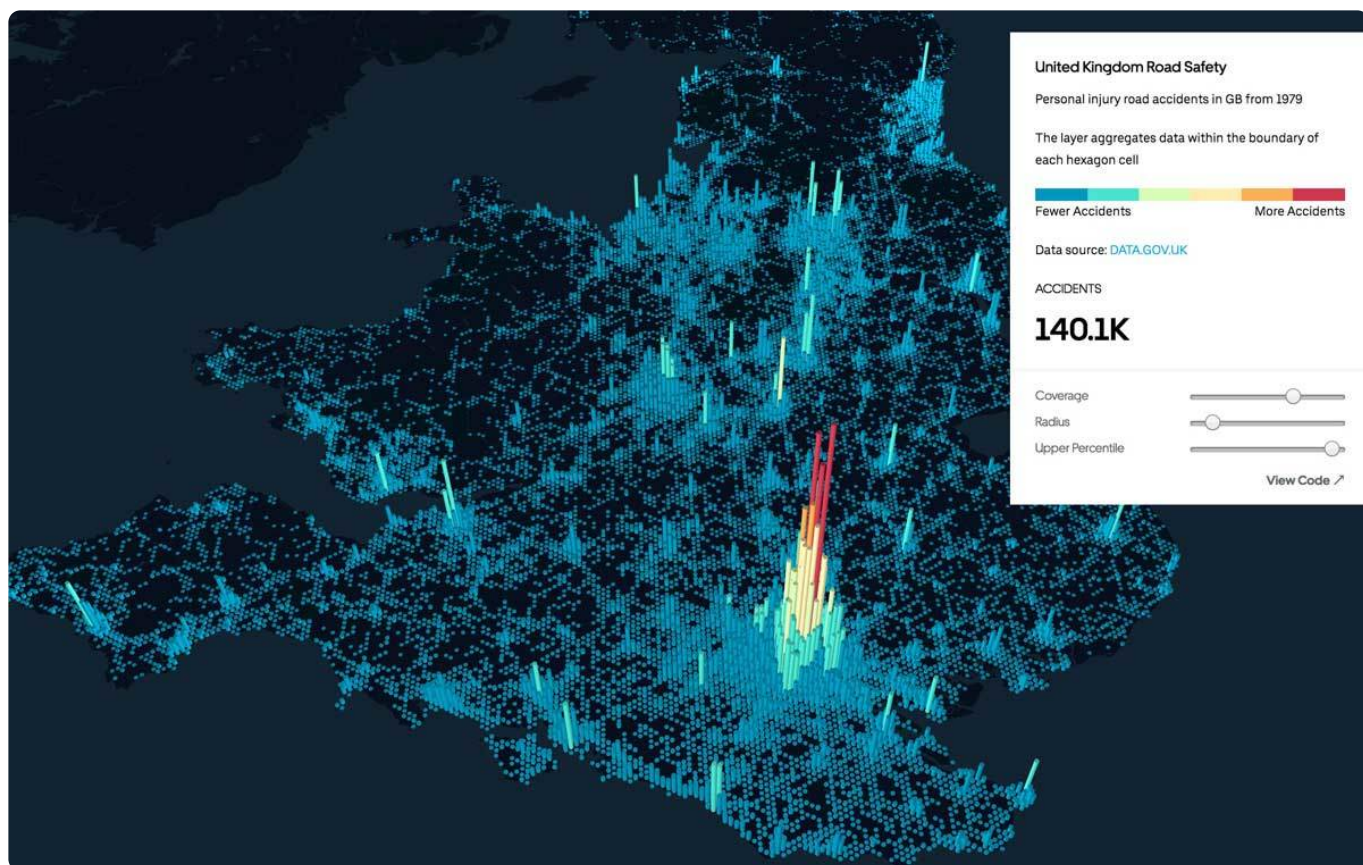
下面，我就分别来介绍一下这些重要的库。

首先是图表库。可视化应用通常需要绘制大量的图表，比如，柱状图、折线图、饼图，还有一些简单的平面地图等等。图表库能够帮助我们快速把它们绘制出来。

社区中有许多优秀的开源图表库，比如我们前面说的 ECharts，或者类似的 [chartlist](#)、[chart.js](#) 等等，它们都属于图表库。

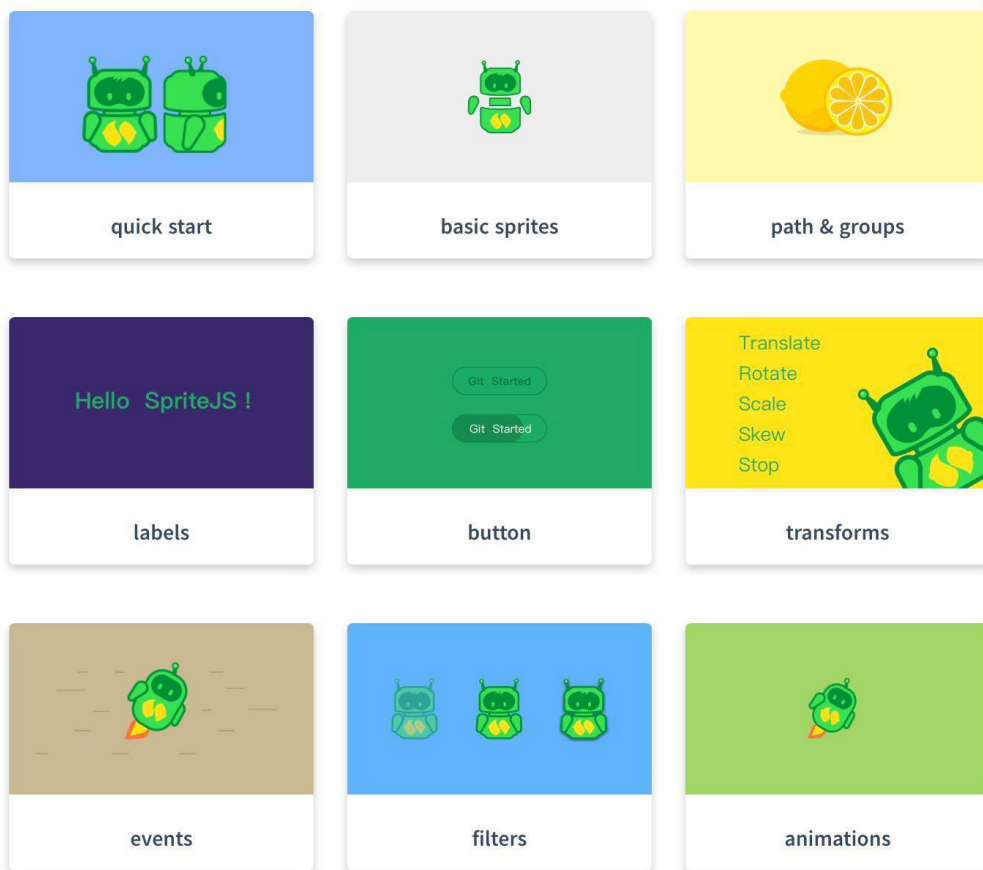


如果要绘制更加复杂的地图，比如，一座城市的交通线路和建筑物三维模型，或者一个园区的立体建筑模型等等，我们可能要依赖专业的 GIS 地图库。社区中比较成熟的 GIS 库也不少，比较常见的像 [Mapbox](#)、[Leaflet](#)、[Deck.gl](#)、[CesiumJS](#) 等等。

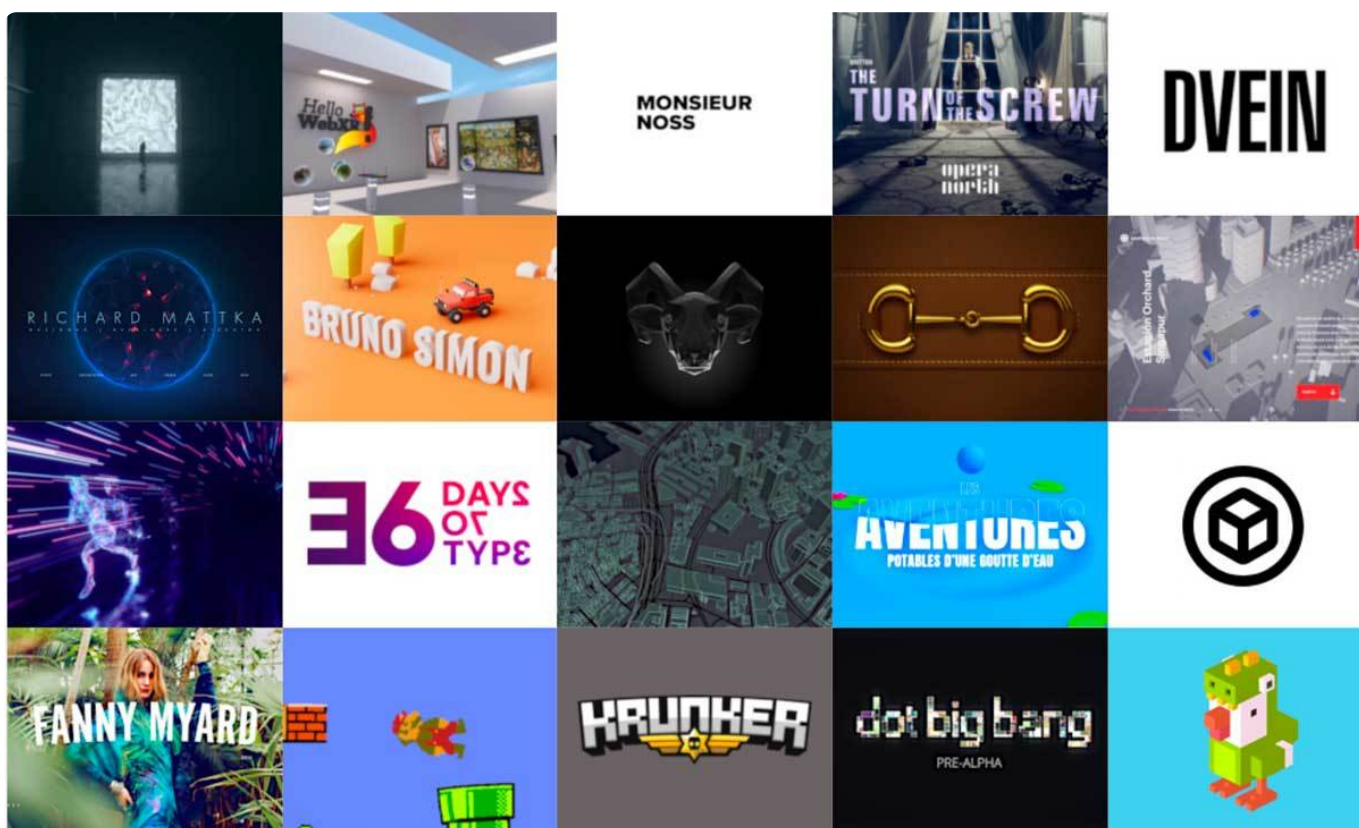


Deck.gl绘制的3D地图效果

如果要**绘制其他更灵活的图形、图像或者物理模型**，我们常用的一些图表库就不一定能完成了。这个时候，我们可以用 [ThreeJS](#)、[SpriteJS](#) 这样比较通用的渲染库（实际上图表库或 GIS 地图库本身底层渲染也基于这些渲染库）。我们可以选择通用的图形库，比如，2D 渲染可以选择 [SpriteJS](#)，3D 渲染可以选择 [ThreeJS](#)、[BabylonJS](#) 以及 [SpriteJS3D](#) 扩展等等。



SpriteJS



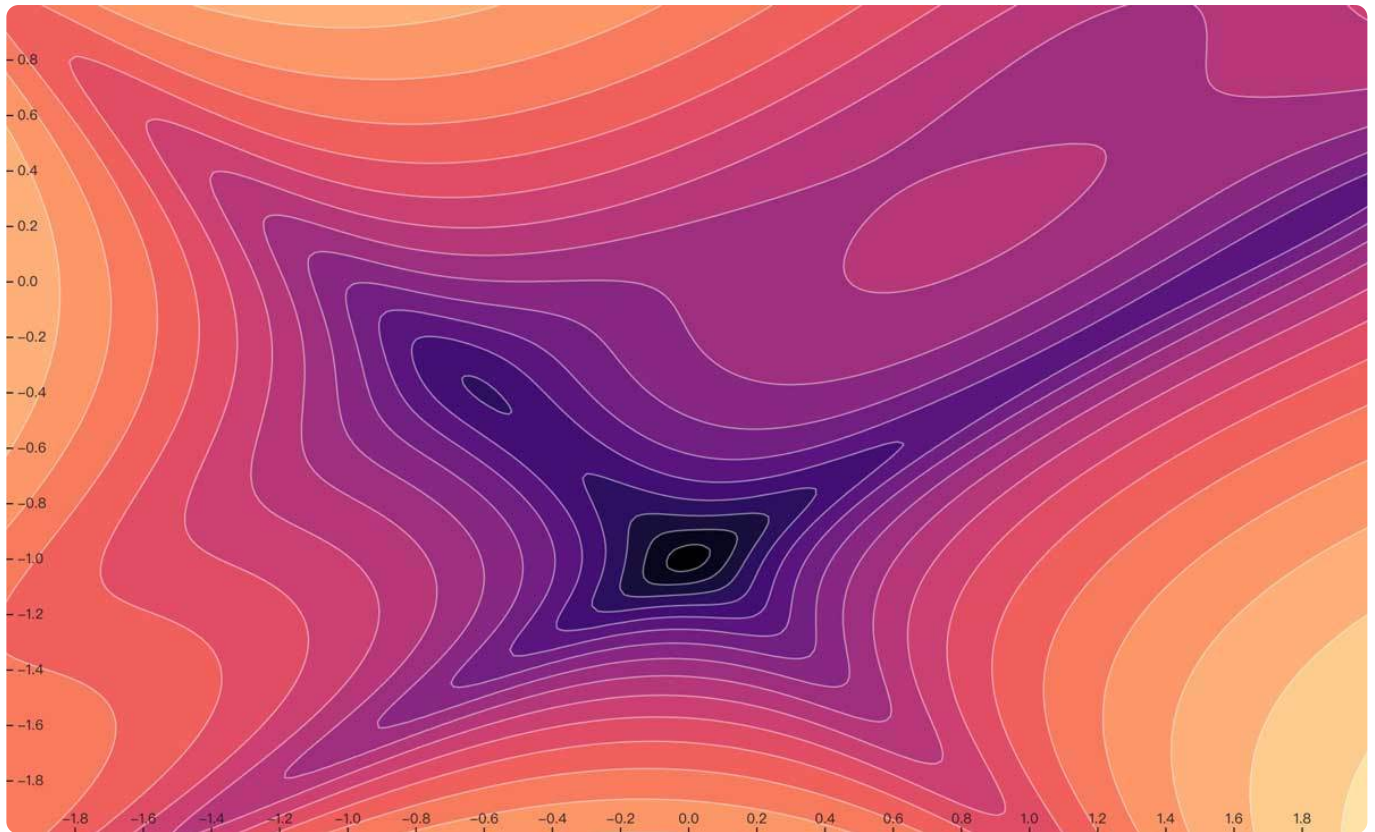
ThreeJS

除了这些库之外，还有一类比较特殊的库，比如 [D3.js](#)，它属于数据驱动框架。那什么是数据驱动框架呢？这是一种特殊的库，它们更**专注于处理数据的组织形式**，而将数据呈现交

给更底层的图形系统（DOM、SVG、Canvas）或通用图形库（SpriteJS、ThreeJS）去完成。

D3.js 与图表库一样，都能完成可视化项目中的各种图表展现，但是它们之间也有区别。下面，我来说说它们各自的优势和劣势。

ECharts 等大部分图表库会提供封装好的图表类型，我们只需要简单配置一下参数就可以使用。但正因为如此，图表能够表现的形式也会被预设的图表类型和封装好的参数所固定了。如果我们想做一些非常个性化的视觉呈现形式，用图表库来做，相对就比较困难。而 D3.js 因为只关注数据的组织形式，将具体的渲染交给底层去做，所以更加灵活，扩展起来也很方便。但相对地，就不像其他的图表库一样，拥有完整的封装了，使用的门槛也就相对高一些。



用d3绘制的等高线

D3.js 是可视化领域一个很重要的库，关于它的具体运作机制比较复杂，我们会在数据篇里详细来讲。

要点总结

这一节课，我们重点讲了 Web 前端和可视化的区别和共同之处。

区别主要有两方面，首先是技术栈的不同。Web 开发主要会用到 HTML 和 CSS，而可视化则较少涉及 HTML 和 CSS，它更多地要同浏览器的 Canvas、SVG、WebGL 等其他图形 API 打交道。其次，Web 开发着重于处理普通的文本和多媒体信息，渲染普通的、易于阅读的文本和多媒体内容，而可视化开发则着重于处理结构化数据，有时需要深入渲染引擎层，从而控制细节，让浏览器渲染出各种相对复杂的图表和图形元素。

不过，如果你原本负责前端开发，想要学习可视化门槛并不高，因为它们之间有很多共通的方法论，而且都依赖于 JavaScript。并且，可视化也有很多成熟的工具和库可以供我们使用。

其实，今天说了这么多，我就是想告诉你，可视化虽然涉及的内容非常多，其核心技术与 Web 开发的前端工程师的侧重点有所不同，但是其核心思想和方法论与 Web 前端是相通的，很多技术也是相辅相成、相互启发的。

因此，前端工程师学习可视化并没有太大的困难，甚至学习这些和可视化相关的图形学理论和技术，也能够提升 Web 开发的前端技术。

一起聊一聊

你为什么想要学习可视化呢？你觉得在学习的过程中有什么难点？

欢迎你在留言区把你的想法和疑惑分享出来，我会尽力为你解答。如果这一节课对你有帮助，也欢迎你把它分享给你的朋友们！

跟月影学可视化

系统掌握图形学与可视化核心原理

月影

奇虎 360 奇舞团团长

可视化 UI 框架 SpriteJS 核心开发者



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 开篇词 | 不写网页的前端工程师，还能干什么？

下一篇 01 | 浏览器中实现可视化的四种方式

精选留言 (24)

 写留言



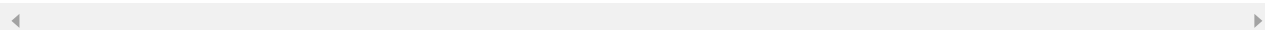
白酒小林

2020-06-22

最近做一个需求是将很多小的病理切片拼成一个大图，能放大缩小倍数还能在图片上做标记的需求。老师有没有推荐的现成的库可以使用？

展开 ▾

作者回复：这个似乎类似于瓦片地图，用leaflet这样的地图的库是不是可以，你可以研究下



 1

 3



Silence

2020-06-22

为啥要学可视化？

因为想要成为一名资深工程师，必须得掌握一个大的方向，只会使用框架和知道其原理，

早晚要被随便就能上手的新人给淘汰掉。目前看来，可视化是一个不错的方向。

展开 ▾



👍 2



新哥

2020-06-22

可以每节课都留点思考题或布置点作业，不然我太懒了

展开 ▾

作者回复: 是的，每节课都会有思考题或者作业的~



👍 2



秋谷

2020-06-22

之所以学习图形化，其实就被‘表象’给吸引到了，各种层出不穷的视觉效果，以及看到某个厉害的作品的时候，总是会感叹一下，牛批啊，这也是我们前端写的诶，所以自己也想体验一下这种感觉。另外学习图形化的难点，相信很多小伙伴应该都有这样的感受吧，我数学不行，会不会太复杂之类的疑问，希望月影大大能够解惑一下。

展开 ▾

作者回复: 其实数学并没有想象的那么难，的确图形学深入下去，可以研究到相当深，要求的数学能力也会很高，但是基础一些的东西并没有那么难，甚至大部分问题只需要有高中数学的基础就能解决。所以可以循序渐进来学习，由浅到深，你会发现这个过程很有趣，而且不知不觉就有很大的成长了。这些底层知识的收获，不仅让你可以在可视化领域游刃有余，也会让你的其他领域技能得到很大提升。



👍 2



筑梦师刘渊

2020-06-22

前端可视化把前端的能力边界扩展了一大截，让我可以再传统的网页开发上实现更高的追求，并且我也觉得这是技术发展的趋势，越来越多的功能都可以在网页端来实现，我也定好了自己的职业方向要深耕前端可视化领域，跟月影大大，加油！

展开 ▾



👍 1



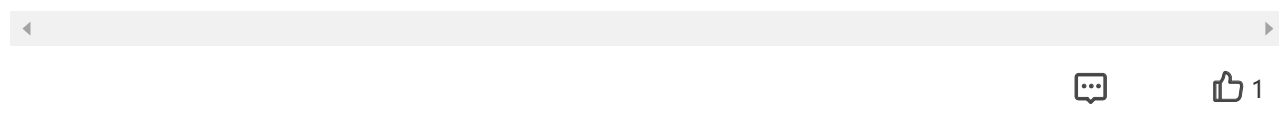
宁康

2020-06-22

学习图形化技术不仅是扩展自己在web前端方面的知识，同时我想图形化技术学好了，以后VR和AR方面也会很有用处。

展开 ▾

作者回复: 是的~



randyshenggo

2020-06-23

想试着做一些炫酷的效果，想知道其中的原理，想设计出一款可视化相关的产品，所以选择学习可视化。我觉得学习过程中的难点，还是如何坚持!!!

展开 ▾



浮生若梦

2020-06-23

echarts也有custom的接口支持客户定制一些比较特别的图形，在视觉效果上基本可以实现d3的效果，但是动画的定制比较弱。另外，老师说的各种库可以结合工作使用场景深入说下具体区别吗？是否开源（免费）？最炫能实现什么效果？性能瓶颈？还有就是工作中怎么结合多个技术之长实现某一效果（比如echarts结合d3甚至webgl，或者某地图结合canvas和webgl）？比如mapbox是国外地图的佼佼者，他跟国内开源的maptalks有什么...

展开 ▾



温同学

2020-06-23

平时最多用echart完成可视化，修改一下配置。难点:普通做多了，定制化3d建模，粒子效果 用的three 脑壳痛 不知道怎么学习

展开 ▾



zcdll

2020-06-23

感觉 前端 可以扩展边界到 任何和 显示 相关的领域，包括1-在什么设备上显示 2-用什么技术显示 3-显示什么图像。

这也是我转行到前端最重要的原因，图像，图形的魔力大大滴

展开 ▾





Warn

2020-06-23

视觉化是平时使用中接触较少的存在，有需求的时候可能就是echarts等拼凑一下，因此也想看看到底能不能触及这一领域。

另外，视觉化这块的兼容性需要考虑很多么，毕竟像Canvas、WebGL接触不多也有担心兼容性的问题

展开 ∨



变成海的话

2020-06-23

经常有些分不清web前端开发和可视化开发的界限，也思考过两者的价值区别，有些迷茫，很多时候同样的东西，用传统前端的html和css可以解决，用可视化的canvas和svg也可以实现，就有些迷，到底该选哪个了。学习这门课，其实主要有两个目的，一个是为了知识，一个是为了延伸自己对可视化的价值和方向的思考

展开 ∨

作者回复: 嗯嗯，其实这些问题是比较普遍的问题。web开发和可视化也并没有特别明显的界限。专栏里我在前面几节课介绍4种图形系统，就是让大家能够知道该如何选择合适的技术实现对应的可视化需求。相信能解答你的疑问。



没事撸代码

2020-06-23

为啥要学可视化？

认为可视化是未来很可行的一个方向，加上自己也比较有兴趣。

难点：

数学知识？图形学，应该会涉及到很多矩阵计算的数学知识把，自己忘的完完了

展开 ∨

作者回复: 可以把数学慢慢捡起来滴～其实我刚开始研究可视化的时候也发现把数学都还给老师了。后面做久了就慢慢重新掌握了，最后发现没有想象的那么难。



落叶归根

2020-06-23

以前用three.js做过室内导航，换工作之后就好几年没碰过了，突然看到这个课程，勾起了内心深处对于可视化的向往

展开 ▾



墨白™

2020-06-23

想要学习可是的目标是找出一个技术上的突破点。我将工程师能力分成3块来评价：

1. 技术能力
2. 项目管理能力
3. 团队管理能力

...

展开 ▾

作者回复: 赞~挺好的个人成长规划



雷霹雳的爸爸

2020-06-23

想学学这个可能跟个人经历有关，就好像，多年前认识的一个女孩儿，其实觉得挺好，就是那个时间点可能彼此之间不太合适...现在忽然又遇到了；学习难点方面，觉得数学、图形学可能是个坎儿，或者能不能个向上真的走到一个高度的必要的一块基石吧，就记得和前面说的那个“女孩儿”厮混的时候，那时候m\$的ie里有vml，firefox还没有canvas啥的，我们要地图上画导航路线，用几个角度带渐变的background图的div来模仿，真够喝一...

展开 ▾



趁你还年轻233

2020-06-23

掌握了可视化的前端，在项目中会更有话语权

展开 ▾



vkingnew

2020-06-23

因为大数据的前端的数据展示，需要不少图标 可以直观的展示数据。我的问题：

- 1.Web开发转数据可视化比较简单，反之呢？
- 2.有什么框架可以兼容PC端，Android，IOS，微信小程序做图形可视化？一次开发多端运行

展开 ▾

作者回复: 纯可视化转前端, 好像一般这样的同学比较少, 不过我想应该也不是很困难, 因为做可视化, 多少也要对浏览器有了解, 而且对JS也要掌握, 可能也就是HTML和CSS熟悉程度低一些。要兼容PC、Android、iOS和小程序, 得看兼容到什么程度, 目前我也不知道是否有能完全适配的框架。不过SpriteJS可以运行在PC端、移动端Web以及Node.js服务端渲染, 也可以支持微信小程序, 就是兼容性方面不能完全保证, 因为我们主要还是用在可视化大屏, 所以并没有测试其他设备的兼容性。



MwumLi

2020-06-23

我一直以为程序员不只是撸码的工具人, 特别是前端程序员, 把语言文字需求视觉化, 仿佛创造一个新世界, 你就是造物主, 学习可视化, 会让这个造物主的权力更大

作者回复: 赞同~



鲸勿落

2020-06-23

可以说随着大数据时代的发展, 数据可视化就会越来越重要了。

