d3-scale-chromatic

这个模块提供了用来表示序列、发散以及分类的颜色方案，它的设计是用来和 d3-scale 的 d3.scaleOrdinal 和 d3.scaleSequential 结合使用。大多数的颜色方案都来自于 `Cynthia A. Brewer` 的 ColorBrewer。由于 `ColorBrewer` 只发布了离散的配色方案, 顺序和发散的颜色是通过 uniform B-splines 插值得到的。

例如，要使用 Accent 颜色方案创建分类颜色比例尺：

```js
var accent = d3.scaleOrdinal(d3.schemeAccent);
```

使用 Blues 颜色方案创建 9 色刻度尺：

```js
var blues = d3.scaleOrdinal(d3.schemeBlues[9]);
```

使用 PiYG 颜色方案撞见发散的、连续的颜色比例尺：

```js
var piyg = d3.scaleSequential(d3.interpolatePiYG);
```

Installing

使用 `NPM` 安装： `npm install d3-scale-chromatic` . 此外还可以下载 latest release 或直接从 d3js.org 以 standalone library 的方式引入. 支持 `AMD` , `CommonJS` , 和基本的标签使用方式，如果使用标签引用，则会暴露全局 `d3` 变量：

```html
<script src="https://d3js.org/d3-color.v1.min.js"></script>
<script src="https://d3js.org/d3-interpolate.v1.min.js"></script>
<script src="https://d3js.org/d3-scale-chromatic.v1.min.js"></script>
<script>

var yellow = d3.interpolateYlGn(0), // "rgb(255, 255, 229)"
    yellowGreen = d3.interpolateYlGn(0.5), // "rgb(120, 197, 120)"
    green = d3.interpolateYlGn(1); // "rgb(0, 69, 41)"
```

或者作为 D3 default bundle 的一部分:

```html
<script src="https://d3js.org/d3.v5.min.js"></script>
<script>

var yellow = d3.interpolateYlGn(0), // "rgb(255, 255, 229)"
    yellowGreen = d3.interpolateYlGn(0.5), // "rgb(120, 197, 120)"
    green = d3.interpolateYlGn(1); // "rgb(0, 69, 41)"

</script>
```

在浏览器中测试 d3-scale-chromatic.

API Reference

Categorical

d3.**schemeCategory10** <>

十个分类颜色的数组，表示为 RGB 十六进制字符串。

d3.**schemeAccent** <>

八个分类颜色的数组，表示为 RGB 十六进制字符串。

d3.**schemeDark2** <>

八个分类颜色的数组，表示为 RGB 十六进制字符串。

d3.**schemePaired** <>

十二种分类颜色的数组，表示为 `RGB` 十六进制字符串。

d3.**schemePastel1** <>

十二种分类颜色的数组，表示为 `RGB` 十六进制字符串。

d3.**schemePastel2** <>

八个分类颜色的数组，表示为 `RGB` 十六进制字符串。

d3.**schemeSet1** <>

九个分类颜色的数组，表示为 `RGB` 十六进制字符串。

d3.**schemeSet2** <>

八个分类颜色的数组，表示为 `RGB` 十六进制字符串。

d3.**schemeSet3** <>

十二种分类颜色的数组，表示为 `RGB` 十六进制字符串。

Diverging

发散的颜色方案可以用作连续型插值器（通常与d3.scaleSequential一起使用）和离散方案（通常与 d3.scaleOrdinal 一起使用）。每一个离散方案，比如 d3.schemeBrBG, 被表示为十六进制颜色字符串数组的数组。数组的第 *k* 个元素表示包含大小为 *k* 的颜色方案; 例如 `d3.schemeBrBG[9]` 包含了 9 个字符串表示的颜色，表示从 `brown` 到 `blue` 到 `green` 的颜色方案。离散颜色方案支持的 *k* 大小为从 3 到 11.

给定一个范围在 [0,1] 的值 *t*，返回一个经过 “BrBG” 颜色方案插值后对应的值，以 `RGB` 字符串表示。

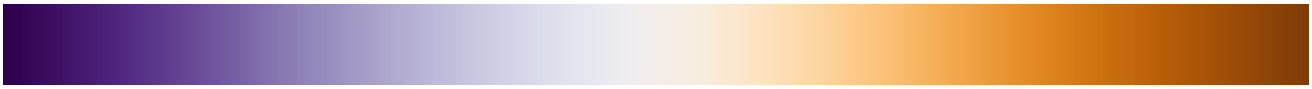
d3.**interpolatePRGn**(*t*) <>
d3.**schemePRGn**[*k*]

给定一个范围在 [0,1] 的值 *t*，返回一个经过 “PRGn” 颜色方案插值后对应的值，以 `RGB` 字符串表示。

d3.**interpolatePiYG**(*t*) <>
d3.**schemePiYG**[*k*]

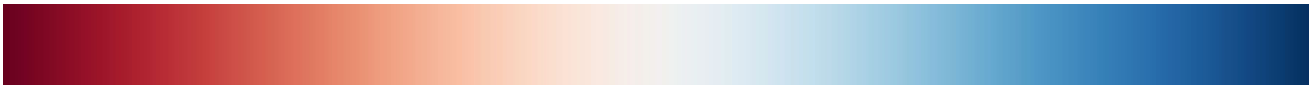
给定一个范围在 [0,1] 的值 *t*，返回一个经过 “PiYG” 颜色方案插值后对应的值，以 `RGB` 字符串表示。

d3.**interpolatePuOr**(*t*) <>
d3.**schemePuOr**[*k*]

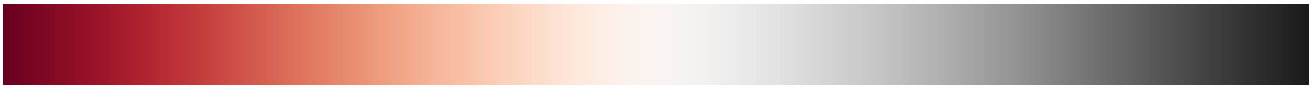


给定一个范围在 [0,1] 的值 *t*，返回一个经过 “PuOr” 颜色方案插值后对应的值，以 `RGB` 字符串表示。

d3.**interpolateRdBu**(*t*) <>
d3.**schemeRdBu**[*k*]



给定一个范围在 [0,1] 的值 *t*，返回一个经过 “RdBu” 颜色方案插值后对应的值，以 `RGB` 字符串表示。



给定一个范围在 [0,1] 的值 *t*，返回一个经过 “RdGy” 颜色方案插值后对应的值，以 `RGB` 字符串表示。

d3.**interpolateRdYlBu**(*t*) <>
d3.**schemeRdYlBu**[*k*]

给定一个范围在 [0,1] 的值 *t*，返回一个经过 “RdYlBu” 颜色方案插值后对应的值，以 `RGB` 字符串表示。

d3.**interpolateRdYlGn**(*t*) <>
d3.**schemeRdYlGn**[*k*]

给定一个范围在 [0,1] 的值 *t*，返回一个经过 “RdYlGn” 颜色方案插值后对应的值，以 `RGB` 字符串表示。

d3.**interpolateSpectral**(*t*) <>
d3.**schemeSpectral**[*k*]

给定一个范围在 [0,1] 的值 *t*，返回一个经过 “Spectral” 颜色方案插值后对应的值，以 `RGB` 字符串表示。

Sequential (Single Hue)

顺序，单色调的颜色方案适合作为连续型插值（通常与 d3.scaleSequential 结合使用）和离散方案 (通常与 d3.scaleOrdinal 结合使用)。每一个离散方案，比如 d3.schemeBlues，被表示为十六进制颜色字符串数组的数组。数组的第 *k* 个元素表示包含大小为 *k* 的颜色方案; 例如 `d3.schemeBlues[9]` 包含了 9 个字符串表示的 `blue` 系列颜色。顺序的多个色调的颜色方案支持的 *k* 值为从 3 到 9.

根据给定的位于 [0, 1] 的值 *t*，返回 “Blues” 色调对应的插值颜色，以 RGB 字符串表示。

d3.**interpolateGreens**(*t*) <>
d3.**schemeGreens**[*k*]

根据给定的位于 [0, 1] 的值 *t*，返回 “Greens” 色调对应的插值颜色，以 RGB 字符串表示。

d3.**interpolateGreys**(*t*) <>
d3.**schemeGreys**[*k*]

根据给定的位于 [0, 1] 的值 *t*，返回 “Greys” 色调对应的插值颜色，以 RGB 字符串表示。

d3.**interpolateOranges**(*t*) <>
d3.**schemeOranges**[*k*]

根据给定的位于 [0, 1] 的值 *t*，返回 “Oranges” 色调对应的插值颜色，以 RGB 字符串表示。

d3.**interpolatePurples**(*t*) <>
d3.**schemePurples**[*k*]

根据给定的位于 [0, 1] 的值 *t*，返回 “Purples” 色调对应的插值颜色，以 RGB 字符串表示。

d3.**interpolateReds**(*t*) <>
d3.**schemeReds**[*k*]

根据给定的位于 [0, 1] 的值 *t*，返回 “Reds” 色调对应的插值颜色，以 RGB 字符串表示。

Sequential (Multi-Hue)

顺序，多色调的颜色方案适合作为连续型插值（通常与 d3.scaleSequential 结合使用）和离散方案 (通常与 d3.scaleOrdinal 结合使用)。每一个离散方案，比如 d3.schemeBuGn，被表示为十六进制颜色字符串数组的数组。数组的第 *k* 个元素表示包含大小为 *k* 的颜色方案; 例如 `d3.schemeBuGn[9]` 包含了 9 个字符串表示的 `blue` 到 `green` 色调的颜色方案。顺序，多个色调的颜色方案支持的 *k* 值为从 3 到 9.

d3.**interpolateViridis**(*t*) <>

根据给定的位于 [0, 1] 之间的值 *t*，返回一个经 van der Walt, Smith and Firing 为 `matplotlib` 设计的 “viridis” 颜色方案插值后的颜色值，表示为 `RGB` 字符串。

d3.**interpolateInferno**(*t*) <>

根据给定的位于 [0, 1] 之间的值 *t*，返回一个经 van der Walt, Smith and Firing 为 `matplotlib` 设计的 “inferno” 颜色方案插值后的颜色值，表示为 `RGB` 字符串。

d3.**interpolateMagma**(*t*) <>

根据给定的位于 [0, 1] 之间的值 *t*，返回一个经 van der Walt, Smith and Firing 为 `matplotlib` 设计的 “magma” 颜色方案插值后的颜色值，表示为 `RGB` 字符串。

d3.**interpolatePlasma**(*t*) <>

根据给定的位于 [0, 1] 之间的值 *t*，返回一个经 van der Walt, Smith and Firing 为 `matplotlib` 设计的 “plasma” 颜色方案插值后的颜色值，表示为 `RGB` 字符串。

d3.**interpolateWarm**(*t*) <>

d3.**interpolateCool**(*t*) <>

根据给定的位于 [0, 1] 之间的值 *t*，返回 Niccoli's perceptual rainbow 颜色值，表示为 RGB 字符串。

d3.**interpolateCubehelixDefault**(*t*) <>

根据给定的位于 [0, 1] 之间的值 *t*，返回 Green's default Cubehelix 颜色值，表示为 RGB 字符串。

d3.**interpolateBuGn**(*t*) <>
d3.**schemeBuGn**[*k*]

根据给定的位于 [0, 1] 之间的值 *t*，返回 "BuGn" 颜色方案值，表示为 RGB 字符串。

d3.**interpolateBuPu**(*t*) <>
d3.**schemeBuPu**[*k*]

根据给定的位于 [0, 1] 之间的值 *t*，返回 "BuPu" 颜色方案值，表示为 RGB 字符串。

d3.**interpolateGnBu**(*t*) <>
d3.**schemeGnBu**[*k*]

根据给定的位于 [0, 1] 之间的值 *t*，返回 "GnBu" 颜色方案值，表示为 RGB 字符串。

d3.**interpolateOrRd**(*t*) <>
d3.**schemeOrRd**[*k*]

D3js: Data-Driven Documents

根据给定的位于 [0, 1] 之间的值 *t*，返回 “OrRd” 颜色方案值，表示为 RGB 字符串。

d3.**interpolatePuBuGn**(*t*) <>
d3.**schemePuBuGn**[*k*]

根据给定的位于 [0, 1] 之间的值 *t*，返回 “PuBuGn” 颜色方案值，表示为 RGB 字符串。

d3.**interpolatePuBu**(*t*) <>
d3.**schemePuBu**[*k*]

根据给定的位于 [0, 1] 之间的值 *t*，返回 “PuBu” 颜色方案值，表示为 RGB 字符串。

d3.**interpolatePuRd**(*t*) <>
d3.**schemePuRd**[*k*]

根据给定的位于 [0, 1] 之间的值 *t*，返回 “PuRd” 颜色方案值，表示为 RGB 字符串。

d3.**interpolateRdPu**(*t*) <>
d3.**schemeRdPu**[*k*]

根据给定的位于 [0, 1] 之间的值 *t*，返回 “RdPu” 颜色方案值，表示为 RGB 字符串。

d3.**interpolateYlGnBu**(*t*) <>
d3.**schemeYlGnBu**[*k*]

根据给定的位于 [0, 1] 之间的值 *t*，返回 “YlGnBu” 颜色方案值，表示为 RGB 字符串。

d3.**interpolateYlGn**(*t*) <>
d3.**schemeYlGn**[*k*]

根据给定的位于 [0, 1] 之间的值 *t*，返回 “YlGn” 颜色方案值，表示为 `RGB` 字符串。

d3.**interpolateYlOrBr**(*t*) <>

d3.**schemeYlOrBr**[*k*]

根据给定的位于 [0, 1] 之间的值 *t*，返回 “YlOrBr” 颜色方案值，表示为 `RGB` 字符串。

d3.**interpolateYlOrRd**(*t*) <>

d3.**schemeYlOrRd**[*k*]

根据给定的位于 [0, 1] 之间的值 *t*，返回 “YlOrRd” 颜色方案值，表示为 `RGB` 字符串。

Cyclical

d3.**interpolateRainbow**(*t*) <>

根据给定的位于 [0, 1] 之间的值 *t*，返回插值范围在 [0.0, 0.5] 的 **d3.interpolateWarm** 和 插值范围在 [0.5, 1.0] 的 **d3.interpolateCool** 组成的颜色方案插值后的颜色值。

d3.**interpolateSinebow**(*t*) <>

根据给定的位于 [0, 1] 之间的值 *t*，返回由 **Jim Bumgardner** 和 **Charle Loyd** 组成的 “sinebow” 颜色方案对应的颜色值。

最后更新: 2019/5/18 下午4:11:02

← d3-color d3-contour →