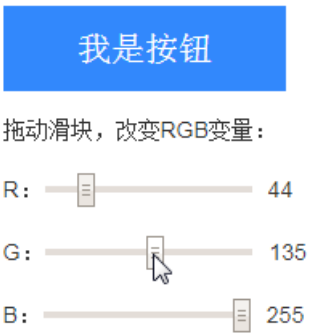
CSS前景背景自动配色技术简介

这篇文章发布于 2018年11月18日，星期日，22:02，归类于 CSS相关。 阅读 5867 次, 今日 17 次 7 条评论

一、颜色匹配效果预览

如下GIF示意，当我们按钮背景色逐渐变淡的时候，文字颜色也从原来的白色变成黑色了，同时边框也显示出来了，其中的配色转换是纯CSS实现的。



您可以狠狠地点击这里：按钮文字及边框颜色随着背景色自动变化demo

拖动R，G，B对应滑块，可以看到按钮文字颜色以及边框颜色，会自动根据背景色不同而发生变化。具体表现为：

- 深色背景下，文字白色，无边框。
- 浅色背景下，文字黑色，有加深边框，便于和周围环境区分开。

这种智能匹配是纯CSS实现的，主要是使用CSS3 calc()计算，以及CSS3原生var变量。

二、配色代码展示

HTML代码很简单，如下：

```
<button class="btn">我是按钮</button>
```

重点和难点在CSS部分：

```
:root {
  /* 定义RGB变量 */
  --red: 44;
  --green: 135;
  --blue: 255;
  /* 文字颜色变色的临界值，建议0.5~0.6 */
  --threshold: 0.5;
  /* 深色边框出现的临界值，范围0~1，推荐0.8+*/
  --border-threshold: 0.8;
}
```

```
.btn {
  /* 按钮背景色就是基本背景色 */
  background: rgb(var(--red), var(--green), var(--blue));

  /**
   * 使用sRGB Luma方法计算灰度（可以看成亮度）
   * 算法为：
   * lightness = (red * 0.2126 + green * 0.7152 + blue * 0.0722) / 255
  */
  --r: calc(var(--red) * 0.2126);
  --g: calc(var(--green) * 0.7152);
  --b: calc(var(--blue) * 0.0722);
  --sum: calc(var(--r) + var(--g) + var(--b));
  --lightness: calc(var(--sum) / 255);

  /* 设置颜色 */
  color: hsl(0, 0%, calc((var(--lightness) - var(--threshold)) * -999999%));

  /* 确定边框透明度 */
  --border-alpha: calc((var(--lightness) - var(--border-threshold)) * 100);
  /* 设置边框相关样式 */
  border: .2em solid;
  border-color: rgba(calc(var(--red) - 50), calc(var(--green) - 50), calc(var(--blue) - 50
), var(--border-alpha));
}
```

乍一看，犹如鸭子听雷——不知所云，其实不复杂，且容我剖析下实现原理。

三、前景背景自动配色原理

1. CSS属性值范围溢出边界渲染特性

CSS这门语言有个很有意思的特性，就是CSS属性值超过正常的范围的时候，只要格式正确，也会渲染，而渲染的值就是合法边界值。

举两个板栗：

1. `opacity`透明度属性值合法范围是0-1，但是，你设置负数，或者极大值，浏览器也能解析，只是要么是0，要么是1而已，如下：

```
.example {
  opacity: -2;    /* 解析为 0，完全透明 */
  opacity: -1;    /* 解析为 0，完全透明 */
  opacity: 2;     /* 解析为 1，完全不透明 */
  opacity: 100;   /* 解析为 1，完全不透明 */
}
```

2. 色值，如HSL，S和L的范围都是0%-100%，但是，你设置负数，或者极大值，浏览器也能解析，只是要么是0%，要么是100%而已，如下：

```
.example {
  color: hsl(0, 0%, -100%);    /* 解析为 hsl(0, 0%, 0%)，黑色 */
  color: hsl(0, 0%, 200%);     /* 解析为 hsl(0, 0%, 100%)，白色 */
}
```

本文的配色技术就活用了这种边界渲染特性。

2. var变量传递给calc实现复杂计算

我们对CSS代码从上往下逐个剖析下。

首先，在:root根选择器上定义几个全局CSS变量（语义上的全局）：

```
:root {
  --red: 44;
  --green: 135;
  --blue: 255;

  --threshold: 0.5;
  --border-threshold: 0.8;
}
```

其中：

–threshold

这个是color变色的临界值，用来和当前RGB颜色值的亮度对比。

–border-threshold

这个是边框颜色透明度的临界值，同样也是和当前RGB颜色值的亮度对比。

然后是`.btn{}`内部的CSS代码：

1. ```
   background: rgb(var(--red), var(--green), var(--blue));
   ```

   这个很好理解，就是基本的RGB色值作为背景色。

2. ```
   --r: calc(var(--red) * 0.2126);
   --g: calc(var(--green) * 0.7152);
   --b: calc(var(--blue) * 0.0722);
   --sum: calc(var(--r) + var(--g) + var(--b));
   --lightness: calc(var(--sum) / 255);
   ```

   这里5行5个CSS变量，需要的其实是最后一个`--lightness`，就是计算当前背景色的亮度。用的是使用sRGB Luma灰度算法①，为什么需要5行呢？因为计算公式就是如此：

   ```
   lightness = (red * 0.2126 + green * 0.7152 + blue * 0.0722) / 255
   ```

   其中，R，G，B色值相乘的系数就是固定的，不同灰度算法系数还不一样。`--lightness`表示亮度，范围是0-1，此时就可以和`--threshold`和`--border-threshold`这两个临界值比对，来确定按钮的文字颜色，边框透明度。

   > ① 这里的灰度可以看成是亮度，实际上HSL的亮度计算方法应该是，R，G，B中的色值最大值和最小值之和的二分之一。

3. ```
   color: hsl(0, 0%, calc((var(--lightness) - var(--threshold)) * -999999%))
   ```

   设置颜色的CSS代码。

   此处`calc`计算翻译成中文就是：（亮度值 – 临界值）* 比例系数。

   其中亮度值在0-1之间游走，临界值是固定的0.5，于是：
   ```

- 如果亮度值小于0.5，亮度值减去临界值为负，由于我们的比例系数是很大很大的负数，负负得正，于是，会得到一个巨大的正数，按照边界渲染原理，会按照100%渲染，于是颜色是白色；
- 如果亮度值大于0.5，亮度值减去临界值为正，由于我们的比例系数是很大很大的负数，于是，会得到一个巨大的负数，按照边界渲染原理，会按照0%渲染，于是颜色是黑色；

以上就是按钮文字颜色变色背景下黑色，深色背景下白色的原理。

4.
```
--border-alpha: calc((var(--lightness) - var(--border-threshold)) * 100);
```

边框透明度是类似的原理。此处`calc`计算翻译成中文就是：（亮度值 – 边框临界值）* 100。

其中亮度值在0-1之间游走，边框临界值是固定的0.8，于是：

- 如果亮度值小于0.8，亮度值减去边框临界值为负，在CSS中，负数透明度会按照边界0渲染，此时边框完全透明；
- 如果亮度值大于0.8，亮度值减去边框临界值为正，此时的透明度计算值会在0~20之间游走，当然，数值大于1的透明度值都会按照1渲染，此时，边框基本处于完全不透明状态，加深的边框显现；

5.
```
border: .2em solid;
border-color: rgba(calc(var(--red) - 50), calc(var(--green) - 50), calc(var(--blue) - 50), var(--border-alpha));
```

设置边框样式，边框颜色比背景色深50个单位值（负数为按照0渲染），然后透明度就是基于亮度动态计算的。深色背景下，按钮边框透明度为0，不显示；浅色背景下，透明度在0~1之间游走，出现，北京颜色越浅，边框透明度越大，边框颜色越深，符合配色预期。

相信经过上面的一番剖析，大家就会明白其实现的原理了。

改变按钮的背景色

.btn类名下的CSS代码是固定的，让我们需要改变按钮的颜色的时候，不是改.btn下的CSS，而是修改:root中的下面3个变量值：

```
--red: 44;
--green: 135;
--blue: 255;
```

CSS设置直接改数值，如果是JS设置，借助`setProperty()`方法，若不了解，可以参考这篇文章：“如何HTML标签和JS中设置CSS3 var变量”。

四、最后结束语

由于var的兼容性限制，这个非常有意思的CSS技巧还不太适合在大型对外项目中使用。

小程序可以一试，因为内核环境相对固定。内部系统，实验项目可以玩一玩，会很有意思。

这种配色技巧其实不仅可以用在按钮上，一些大区域的布局也是可以用的，因为这些布局的背景色可能是动态的，此时，文字颜色的配色也可以借助CSS实现自动化。

另外，本文demo中按钮文字就黑白两色，实际上，我们的相乘系数小一点的话，可以有更多的色值出现，配色会更加精致。

参考文章

- Switch font color … CSS

以上就是本文全部内容，感谢阅读！

😃

> 《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

（本篇完） // 想要打赏？点击这里。有话要说？点击这里。

« CSS filter:hue-rotate色调旋转滤镜实现按钮批量生产

纯CSS实现任意格式图标变色的研究 »

猜你喜欢

- 小tips:了解CSS/CSS3原生变量var
- Canvas中颜色过渡动画效果的实现
- Chrome opacity非1时border-radius圆角边框剪裁问题
- CSS届的绘图板CSS Paint API简介
- 小tips: 如何HTML标签和JS中设置CSS3 var变量
- CSS3下的147个颜色名称及对应颜色值
- JS HEX十六进制与RGB, HSL颜色的相互转换
- 小tip: 了解LinearRGB和sRGB以及使用JS相互转换
- 小tip:CSS vw让overflow:auto页面滚动条出现时不跳动
- 翻译 - 解释JavaScript的“预解析(置顶解析)”
- CSS float浮动的深入研究、详解及拓展(二)

标签： border, calc, css3, css相关, hsla, opacity, RGB, var, 变量, 按钮, 颜色

发表评论（目前7条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1. **dotbin说道：**
2018年12月18日 15:38

膜拜张大大

回复

2. **????说道：**
2018年12月13日 09:17

谢谢

回复

3. **Caldis说道：**
2018年11月26日 14:08

很有意思的分享，之前就在想，iPhone 首頁的應用標題背景色會隨着背景顏色而變成黑色或白色，後來發現是根據對比度計算得來的，只要文字顏色與背景色疊加低於某個閾值，就更換成白色/黑色來保持可讀性。

另外，現在 Chrome Dev Tools 在編輯 CSS 面板的時候，點擊文字顏色的可視化調色盤也可以在底部看到 Contrast ratio 的實時計算值，來保證文字的辨識度

回复

4. **优惠券说道：**
2018年11月22日 15:53

楼主写的真是专业。我这种小白只能表示膜拜。。。

回复

5. **z.z.说道：**
2018年11月19日 17:18

先回去钻研了var的那篇，再回来看，还是一脸蒙蔽，看到“只要格式正确，也会渲染，而渲染的值就是合法边界值。”豁然开朗

回复

6. **animaK说道：**
2018年11月19日 11:49

都还没使用过css原生变量，学习了

另外，三、前景背景自动配色原理，第五点有个背景颜色打错成北京颜色了

回复

7. **aaaa说道：**
2018年11月19日 09:58

这篇难度也太大了,学不动了

回复

最新文章

» 常见的CSS图形绘制合集
» 粉丝群第1期CSS小测点评与答疑
» 分享三个纯CSS实现26个英文字母的案例
» 小tips: 纯CSS实现打字动画效果
» CSS/CSS3 box-decoration-break属性简介
» CSS :placeholder-shown伪类实现Material Design占位符交互效果
» 从天猫某活动视频不必要的3次请求说起
» CSS vector-effect与SVG stroke描边缩放
» CSS ::backdrop伪元素是干嘛用的?
» 周知：CSS -webkit-伪元素选择器不再导致整行无效

今日热门

» 常见的CSS图形绘制合集 (179)
» 粉丝群第1期CSS小测点评与答疑 (113)
» 未来必热：SVG Sprite技术介绍 (111)
» HTML5终极备忘大全（图片版+文字版） (85)
» 让所有浏览器支持HTML5 video视频标签 (83)
» Selectivizr-让IE6~8支持CSS3伪类和属性选择器 (80)
» CSS3下的147个颜色名称及对应颜色值 (78)
» 小tips: 纯CSS实现打字动画效果 (72)
» 写给自己看的display: flex布局教程 (70)
» 分享三个纯CSS实现26个英文字母的案例 (69)

今年热议

» 《CSS世界》女主角诚寻靠谱一起奋斗之人 (76)
» 不借助Echarts等图形框架原生JS快速实现折线图效果 (64)
» 看，for..in和for..of在那里吵架！ (60)
» 是时候好好安利下LuLu UI框架了！ (47)
» 原来浏览器原生支持JS Base64编码解码 (35)
» 妙法攻略：渐变虚框及边框滚动动画的纯CSS实现 (33)
» 炫酷H5中序列图片视频化播放的高性能实现 (31)
» CSS scroll-behavior和JS scrollIntoView让页面滚动平滑 (30)
» windows系统下批量删除OS X系统.DS_Store文件 (26)
» 写给自己看的display: flex布局教程 (26)

猜你喜欢

- 小tips:了解CSS/CSS3原生变量var
- Canvas中颜色过渡动画效果的实现
- Chrome opacity非1时border-radius圆角边框剪裁问题
- CSS届的绘图板CSS Paint API简介
- 小tips: 如何HTML标签和JS中设置CSS3 var变量
- CSS3下的147个颜色名称及对应颜色值
- JS HEX十六进制与RGB, HSL颜色的相互转换
- 小tip: 了解LinearRGB和sRGB以及使用JS相互转换
- 小tip:CSS vw让overflow:auto页面滚动条出现时不跳动
- 翻译 - 解释JavaScript的“预解析(置顶解析)”
- CSS float浮动的深入研究、详解及拓展(二)