# CSS变量(CSS Variables)

## 概述

如果突然有一天,当你页面的主题色发生变化时,有许多元素的颜色要一起变化,这是你会考虑怎么做呢?使用Less/Sass/Stylus来定义一个颜色变量?还是专门去定义多了个类名来控制元素?

使用第一种方法,有他的弊端,浏览器无法识别CSS预处理器的语法,所以我们都会将Less/Sass/Stylus的文件编译成CSS文件,这样的话就无法再去修改CSS预处理器中变量的值了,因为我们的页面中根本没有Less等CSS预处理器文件

使用第二种方法,也有弊端,如果一个元素是字体颜色为blue,另一个元素是背景颜色为blue,那这时候就要取多个类名才能实现,这样太不方便了

说完了上面这两种方法的弊端,那我们就要介绍一下今天的主角:CSS变量(英文CSS Variables,官方取名为自定义属性或者级联变量)

## 一.什么是CSS变量

**CSS 变量当前有两种形式**：

变量，就是拥有合法标识符和合法的值。可以被使用在任意的地方。可以使用var()函数使用变量。例如：var(--example-variable)会返回--example-variable所对应的值

自定义属性。这些属性使用--\*where\*的特殊格式作为名字。例如--example-variable: 20px;即使一个css声明语句。意思是将20px赋值给--example-varibale变量。

注意：

1.在之前的标准中，自定义属性以var-作为前缀，后来才改成--前缀。Firefox 31和以后的版本遵循新标准。

2.自定义属性和常规属性一样，作用在当前的层级，若没有定义，则从其父元素继承其值。

## 二.为什么要使用CSS变量

在构建大型站点时，作者通常会面对可维护性的挑战。在这些网页中， 所使用的 CSS 的数量是非常庞大的，并且在许多场合大量的信息会重复使用。例如，在网页中维护一个配色方案，意味着一些颜色在CSS文件中多次出现，并被重复使用。当你修改配色方案时，不论是调整某个颜色或完全修改整个配色，都会成为一个复杂的问题，不容出错，而单纯查找替换是远远不够的。

如果使用了 CSS 框架，这种情况会变得尤其糟糕，此时如果要修改颜色，则需要对框架本身进行修改。在这些场合使用 LESS 或 Sass 类似的预处理器是非常有帮助的，但是这种通过添加额外步骤的方式，可能会增加系统的复杂性。CSS变量为我们带来一些预处理器的便利，并且不需要额外的编译。

这些变量的第二个优势就是名称本身就包含了语义的信息。CSS 文件变得易读和理解。main-text-color比文档中的#00ff00更容易理解，特别是同样的颜色出现在不同的文件中的时候。

## 三.怎么使用CSS变量

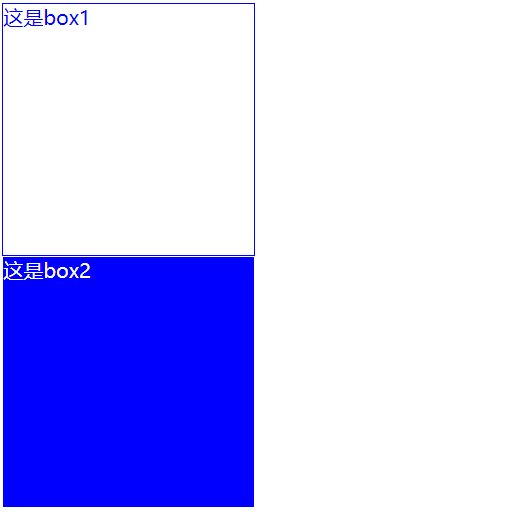
初始样式:

|  |
| --- |
| <style>  #box1{  width:200px;  height: 200px;  border: 1px solid;  color:blue;  }  #box2{  width:200px;  height: 200px;  border: 1px solid;  background-color:blue;  color:white;  }  </style> |

页面结构:

|  |
| --- |
| <body>  <div id="box1">  这是box1  </div>  <div id="box2">  这是box2  </div>  </body> |

页面效果:



### 1.定义CSS变量

这时候我们来定义一个CSS变量,变量值为blue

|  |
| --- |
| <style>  :root{  --global-color:blue;  }  </style> |

在style标签中首先声明一个全局伪元素:root,然后在内部声明变量名及其变量值

注意:变量名一定要--开头

到这里我们成功定义了CSS变量,接下来我们来看下如何使用CSS变量

### 2.使用CSS变量

我们使用上面定义的变量分别替换掉#box1元素的color属性的值和#box2元素的background-color属性的值

|  |
| --- |
| <style>  :root{  --global-color:blue;  }  #box1{  width:200px;  height: 200px;  border: 1px solid;  color:var(--global-color);  }  #box2{  width:200px;  height: 200px;  border: 1px solid;  background-color:var(--global-color);  color:white;  }  </style> |

使用起来虽然没有Less等CSS预处理器那么简单,但是还是比较方便的

语法:CSS属性名:var(CSS变量名)

注意:CSS变量名是包括了—的,千万别漏了

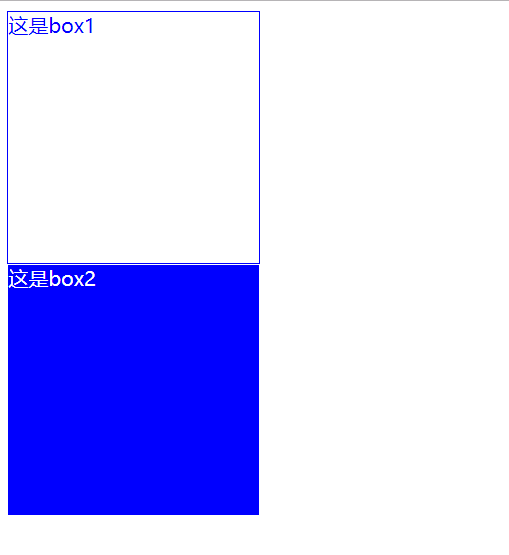
到这里我们已经看完了CSS变量的定义和使用,那接下来我们再来看看怎么修改CSS变量的值

### 3.修改CSS变量

在我们的页面中,如果想要动态修改CSS变量的值,我们需要借助到JavaScript来实现

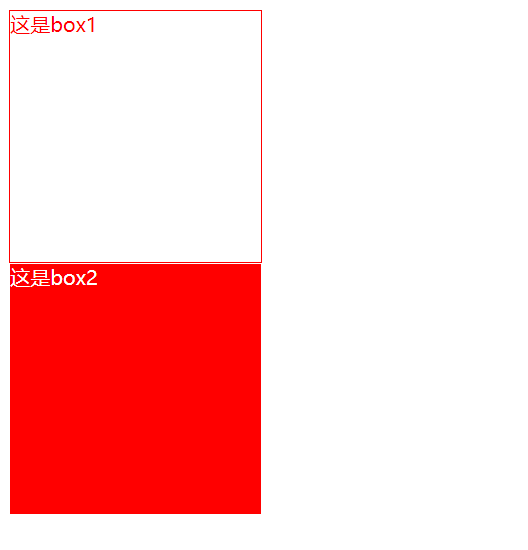
|  |
| --- |
| <script>  //找到html标签  var htmlNode=document.documentElement;  //将html标签(根标签)的CSS变量(--global-color)的值修改为red  htmlNode.style.setProperty("--global-color", "red");  </script> |

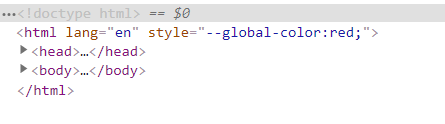
修改前:





修改后:





此时我们可以很明显的看到html标签上多了一个标签属性style,内部是我们修改的CSS变量及其变量值,到这里就修改成功了

语法:DOM对象.style.setProperty(CSS变量名,修改后的CSS变量值)

注意:修改时不能使用(DOM对象.style[CSS变量名]=变量值)的语法进行修改,例如:htmlNode.style["--global-color"]="red",此方法无效

### 4.读取CSS变量

在我们的JavaScript代码中,时常会需要去读取CSS变量的值,那我们来看下如何读取CSS变量的值

|  |
| --- |
| <script>  //找到html标签  var htmlNode=document.documentElement;  //将html标签(根标签)的CSS变量(--global-color)的值修改为red  htmlNode.style.setProperty("--global-color", "red");  //读取html标签(根标签)的CSS变量(--global-color)的值,并在控制台中打印  // var value=htmlNode.style.getPropertyValue('--global-color');  var value=getComputedStyle(htmlNode)['--global-color'];  console.log(value)  </script> |

### 5.CSS变量的其他操作