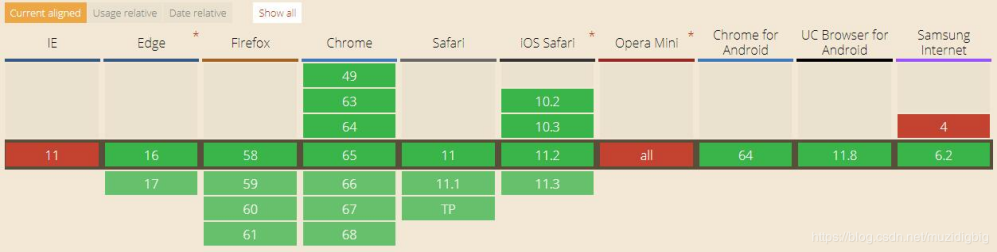
**风云再起：CSS变量**

一、兼容性

目前连微软都已宣布 Edge 浏览器将支持 CSS 变量。这个重要的 CSS 新功能，所有主要浏览器已经都支持了。



 二、用法

声明css变量的时候，变量名前面要加两根连词线（--）。

变量名大小写敏感，--header-color和--Header-Color是两个不同变量。

var()函数用于读取变量。

var()函数还可以使用第二个参数，表示变量的默认值。如果该变量不存在，就会使用这个默认值。

第二个参数不处理内部的逗号或空格，都视作参数的一部分。

1.全局/局部变量

1.1 :root伪类可以看做是一个全局作用域，在里面声明的变量，他下面的所有选择器都可以拿到

:root { --color: blue; }

.box{color: var(--color)}

1.2 局部变量

.box{

--color: red;

color: var(--color);

}

1.3 var()函数的第二个参数默认值

.box{

--color: red;

color: var(--bg,pink);

}

1.4 可以结合CSS3 calc()一起计算

p{

--size: 20;

font-size: calc(var(--size) \* 1px);//20px

}

1.5 CSS变量的继承(就近原则)

<div class="box" id="alert">muzidigbig</div>

:root { --color: blue; }

div { --color: green; }

#alert { --color: red; }

\* { color: var(--color); }

三、注意问题

1.当存在多个同样名称的变量时候，变量的覆盖规则由CSS选择器的权重决定的，但并无!important这种用法。

2.变量的取值采用就近原则。

3.如果变量值是数值，不能与数值单位直接连用。必须使用calc()函数，将它们连接。

.foo {

--gap: 20;

/\* 无效 \*/

margin-top: var(--gap)px;

}

.foo {

--gap: 20;

/\* 有效 \*/

margin-top: calc(var(--gap) \* 1px);

}

4.如果变量值带有单位，就不能写成字符串。

/\* 无效 \*/

.foo {

--foo: '20px';

font-size: var(--foo);

}

/\* 有效 \*/

.foo {

--foo: 20px;

font-size: var(--foo);

}

5. CSS属性名是不可以使用变量的。例如下面写法是错误的：

body {

--bc: background-color;

var(--bc): #369;

}

6. css变量是存在缺省值，只要定义正确，里面的值不对，结果以缺省值显示：如:

body {

--color: 20px;

background-color: #369;

background-color: var(--color, #cd0000);/\*背景色为：transparent\*/

}

7. css变量默认尾部是有空格的。例如：

p{

--size: 20;

font-size: var(--size)px;//等同于font-size:20 px;这里将使用元素默认的大小。这里可以结合上面例子calc()使用。

}

四、兼容性处理

检查是否兼容:说实话我是不想检测兼容的反正用的都是最新的东西，等团队决定要用了兼容问题也不会太严重

a {

color: #7F583F;

color: var(--primary);

}

/\* 用属性值得无效声明 \*/

@supports ( (--a: 0)) {

/\* supported \*/

}

@supports ( not (--a: 0)) {

/\* not supported \*/

}

/\* 也可以使用@support命令进行检测。 \*/

const isSupported =

window.CSS &&

window.CSS.supports &&

window.CSS.supports('--a', 0);

if (isSupported) {

/\* supported \*/

} else {

/\* not supported \*/

}

**与JS的联用：**

试想一下，在以前的项目中，你是如何写“雪花飘落”特效的？你是如何处理“在不同位置点击显示元素”特效的？你又是如何实现“tab切换”特效的？

现在有了CSS变量，你完全可以将上面这些情况的对应样式用变量表示，然后用JavaScript去操控变量：

比如：document.body.style.getPropertyValue(“变量名”).trim()、document.body.style.setProperty(“名”,”值”)

**补充知识：**

calc() 函数用于动态计算长度值。

需要注意的是，运算符前后都需要保留一个空格，例如：width: calc(100% - 10px)；

任何长度值都可以使用calc()函数进行计算；

calc()函数支持 "+", "-", "\*", "/" 运算；

calc()函数使用标准的数学运算优先级规则；

表达式中有“\*”和“/”时，其前后可以没有空格，但建议留有空格。

比如说“width:calc(50% + 2em)”，这样一来你就不用考虑元素DIV的宽度值到底是多少，而把这个烦人的任务交由浏览器去计算。

那么我们就可以将其和animation-delay结合以打造CSS中的“翻转特效”：

<head>

<meta charset="utf-8">

<title></title>

<style>

\*{margin: 0;padding: 0;font-family: "微软雅黑";}

body{display: flex;align-items: center;justify-content: center;min-height: 100vh;background-color: #000;}

.wavy{position: relative;-webkit-box-reflect: below -12px linear-gradient(transparent,rgba(0,0,0,.2));}

.wavy span{position: relative;display: inline-block;color: #fff;font-size: 2em;animation: All 1s ease-in-out infinite;animation-delay: calc(.1s\*var(--i));}

@keyframes All{

0%{transform: translateY(0px);}

20%{transform: translateY(-24px);}

40%,100%{transform: translateY(0px);}

}

</style>

</head>

<body>

<div class="wavy">

<span style="--i:1">欢</span>

<span style="--i:2">迎</span>

<span style="--i:3">来</span>

<span style="--i:4">到</span>

<span style="--i:5">CSS</span>

<span style="--i:6">世</span>

<span style="--i:7">界</span>

</div>

</body>

而以前，我们大多都是通过 :nth-child(n) 来获取某一元素，然后给其赋值transition-delay延迟过渡效果。

任务：你可以试着将下面链接文章（@csdn-云小梦）中实例用本文介绍方法改写：

<https://yunxiaomeng.blog.csdn.net/article/details/102734021>