

# 利用clamp()超快速实现响应式布局！

---

## 前言

响应式布局已经被提出很多年了，因为我们的电子设备多种多样，比如手机、平板、电脑等等，这些设备的屏幕大小都是不一样的。如果我们的网页在每个设备上显示一样的话，那么用户体验无疑是很差的，所以为了解决这个问题，我们通常有两种方案：响应式布局和编写多套网页代码。编写多套网页代码工作量很大，所以很多公司都直接采用响应式布局，响应式布局的原理就是根据设备屏幕的不同大小加载不同的css，我们经常采取的方案就是媒体查询。

不过今天我给大家介绍一个更为简单的方法实现响应式：**利用clamp()函数！**

## 1.基本介绍

我们知道不仅仅JS里面有函数，其实CSS里面也有函数的，比如calc()、max()等等都是CSS函数，今天我们讲的**clamp()**也是CSS的一个函数。

学习一个新的知识我们都先就要去官网看看，毕竟这才是权威，我们先来看看官网是怎么解释的。

**官网解释：**

clamp() 函数的作用是把一个值限制在一个上限和下限之间，当这个值超过最小值和最大值的范围时，在最小值和最大值之间选择一个值使用。它接收三个参数：最小值、首选值、最大值。

估计很多小伙伴都会被上面官网的解释绕晕吧！不过我们应该还是有大概的理解，我们从上面的话中抽出一些关键词，方便大家理解。

**关键词：**

- 限制值范围
- 最大值
- 最小值
- 三个参数

上面的关键词中我们先来学习三个参数，了解了三个参数之后我们就会很快理解该函数做什么了。

### 接收参数：

- MIN：最小值，通常就是我们CSS表示大小的单位，比如rem，px，vw等等
- VAL：首选值，通常就是我们CSS表示大小的单位，比如rem，px，vw等等
- MAX：最大值，通常就是我们CSS表示大小的单位，比如rem，px，vw等等

知道这三个参数后，我们在将这三个参数代入官方的解释中去看，应该就能理解了，我们这里用自己的话讲解一遍。

### 通俗解释：

既然clamp()是一个函数，那么该函数的返回值是什么呢？首先该函数接收三个参数，形如clamp(MIN, VAL, MAX)，三个参数分别代表最小值、首选值、最大值。当VAL < MIN时，函数返回MIN，当VAL > MAX时，函数返回MAX，当MIN < VAL < MAX时，函数返回VAL。

到这里我们应该就能真正理解该函数在做什么了，无非就是三个参数之间的比较，然后返回不同的值罢了。

那和我们的响应式有什么关系呢？

响应式的页面原理就是根据不同设备大小渲染不同的CSS，而我们这里的函数也是根据VAL的大小，返回不同的值，是不是和我们的响应式有相似之处了，所以我们可以利用clamp函数来实现响应式布局。

## 2.简单使用

既然知道了clamp函数的语法以及作用，那么我们可以写一个简单的小案例加深一下印象，也为小伙伴们思考它的使用场景提供一点思路。

### 实现需求：

屏幕大小发生变化，字体大小也会发生变化。

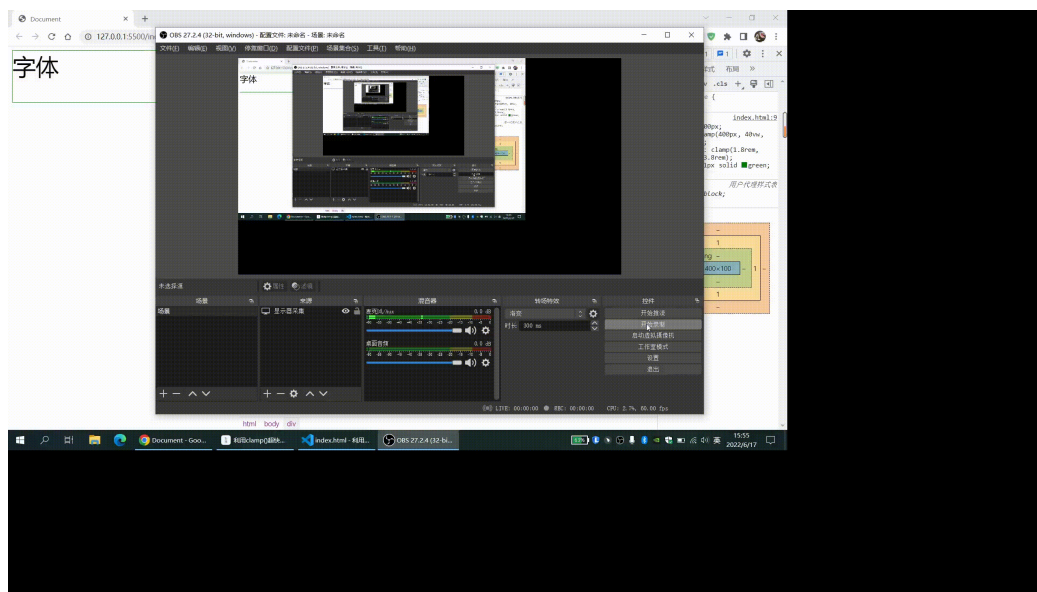
这是一个很常见的需求，我们需要在不同的屏幕大小之下显示不同的字体大小。

### 代码如下：

```
1 <head>
2   <style>
3     div {
4       height: 100px;
5       width: clamp(400px, 40vw, 500px);
6       font-size: clamp(1.8rem, 10vw, 3.8rem);
7       border: 1px solid green;
8     }
9   </style>
10 </head>
11 <body>
```

```
12 <div>字体</div>
13 </body>
```

实现效果：



上段代码中给div设置宽度时利用clamp函数，其中MIN=400px，VAL=40vw，MAX=500px。当我们拖动控制台时，40vw的实际宽度也是在跟着变化的，当40vw<400px时，div的宽度为400px，当40vw>500px时，div的宽度为500px，当400px<40vw<500px，div的宽度为40vw。

同理，代码中的字体大小的具体值我们也利用了clamp函数，字体最小为1.8rem，最大为3.8rem。具体结果都是VAL与MIN和MAX比较得出的。

### 3.响应式布局

前面我们的简单示例也算实现了响应式，不过没有那么正式，可能很多小伙伴还体会不到它的魅力。我们接下来就稍微正式一点，利用clamp实现一个简单的响应式页面。

代码如下：

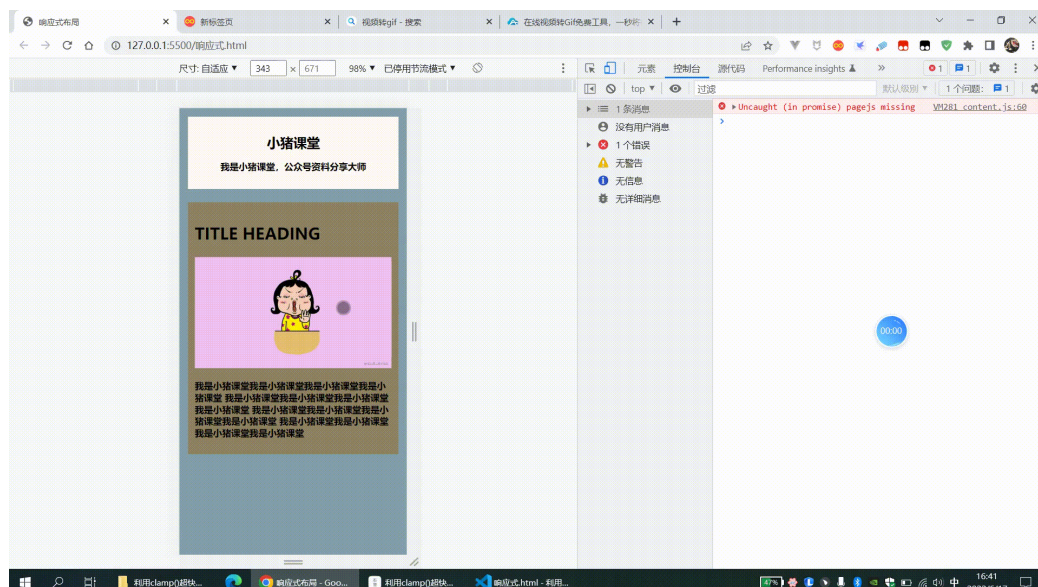
```
1 <head>
2 <style>
3   * {
4     box-sizing: border-box;
5   }
6   body {
7     padding: 5px;
8     background: #84a0ab;
9   }
10
11   .header {
12     padding: 12px;
13     text-align: center;
14     background: #fffaf6;
15     font-weight: bold;
16   }
17 }
```

```

18     .card {
19         background-color: #8f8261;
20         padding: 10px;
21         margin-top: 20px;
22         font-weight: bold;
23     }
24
25     img {
26         width: clamp(15vw, 800%, 100%); // 图片宽度使用为100%
27     }
28
29     h1 {
30         font-size: clamp(20px, 5vw, 35px);
31     }
32
33     p {
34         font-size: clamp(10px, 4vw, 20px);
35     }
36 }
37 </style>
38 </head>
39
40 <body>
41     <div class="header">
42         <h1>小猪课堂</h1>
43         <p>我是小猪课堂，公众号资料分享大师</p>
44     </div>
45     <div class="card">
46         <h2>TITLE HEADING</h2>
47         
48         <p>
49             我是小猪课堂我是小猪课堂我是小猪课堂我是小猪课堂
50             我是小猪课堂我是小猪课堂我是小猪课堂我是小猪课堂
51             我是小猪课堂我是小猪课堂我是小猪课堂我是小猪课堂
52             我是小猪课堂我是小猪课堂我是小猪课堂我是小猪课堂
53         </p>
54     </div>
55 </body>
56

```

实现效果：



从上图中我们可以看到当屏幕大小不断变化时，我们的字体大小以及标题都是在跟着变化的。在上段代码中我们分别给p标签、h1标签以及img标签都使用了clamp函数，以此来动态设置它们的大小。

我们常见的响应式布局其实就和上图类似了，我们只需要最大最小临界值即可了。

但是我们仔细想一想会发现，我们的设备有很多，比如平板、电脑、手机等等，这个时候就会出现多个首选值VAL，虽然我们可以利用clamp函数实现基本的响应式，但是如果想要更加细分每个设备显示为什么样，可能还需要借助max()、min()函数了。

## 总结

clamp()函数自身是不复杂的，我们学会使用它非常快，但是难点在于我们如何判断一个场景是否可以使用它，这就是需要我们日积月累的经验来决定了。比如说我们需要做手机和电脑端的响应式，那么完全可以利用clamp函数，当元素宽度小于某个值时，我们的字体、宽度、等等都可以设置为某个值，这就简单的实现了响应式布局。当然实现响应式布局的方式还有很多种，需要大家下来了解了。