

问题一：如何适配移动端？【经典】

不要仅简单的回答用 媒体查询 或者 用js单位适配函数，回答用js尤其是最不好的。

参考回答：

"结果我们发现，想让网页在一堆不同的设备上合理展示，只需要在最终产品上添加一点 CSS 媒体查询就可以了。这件事情之所以这么简单，关键在于我们的布局原本就是弹性可伸缩的。因此，优化网页在小屏幕上的表现，其实只意味着把一些外边距收拢到最小程度，然后把因为屏幕太窄而无法显示成双列的侧栏调整为单列布局而已。"

在《css揭秘》一书中关于响应式布局推荐有一些布局建议：

- 使用百分比长度来取代固定长度。如果做不到这一点，也应该尝试使用与视口相关的单位（vw、vh、vmin 和 vmax），它们的值解析为视口宽度或者高度的百分比。
- 当你需要在较大分辨率下得到固定宽度时，使用max-width而不是width，因为它可以适应较小的分辨率，而无需使用媒体查询。
- 不要忘记为替换元素（比如img、object、video、iframe等）设置一个max-width，值为100%
- 假如背景图片需要完整地铺满一个容器，不管容器的尺寸如何变化，background-size:cover; 这个属性都可以做到。但是，我们也要时刻牢记--带宽并不是无限的，因此在移动网页中通过css把一张大图缩小显示往往是不明智的。
- 当图片（或其他元素）以行列式进行布局时，让视口的宽度来决定列的数量。弹性盒布局（即Flexbox）或者display:inline-block; 加上常规的文本折行行为，都可以实现这一点。
- 在使用多行文本时，指定column-width（列宽）而不是指定column-count（列数），这样它就可以在较小的屏幕上自动显示为单例布局。

```
// 背景图片
```

```
bg-image($url)
```

```
background-image: url($url + "@2x.png")
```

```
@media (-webkit-min-device-pixel-ratio: 3), (min-device-pixel-ratio: 3)
```

```
background-image: url($url + "@3x.png")
```

```
// 不换行
```

```
no-wrap()
```

```
text-overflow: ellipsis
```

```
overflow: hidden
```

```
white-space: nowrap
```

```
// 扩展点击区域
```

```
extend-click()
```

```
position: relative
```

```
&:before
```

```
content: ''
```

```
position: absolute
```

```
top: -10px
```

```
left: -10px
```

```
right: -10px
```

```
bottom: -10px
```