

HTML5 相应式布局

九、HTML5响应式布局

什么是响应式网页设计（布局）

响应式布局的实现

1. Meta标签定义

2.使用Media Queries适配对应样式

设备类型

设备特性

实现响应式布局

响应式布局缺点和优点

这里有一个很严重的缺点

解决方案：

九、HTML5响应式布局

什么是响应式网页设计（布局）

响应式网页设计 (Responsive Web design, 通常缩写为RWD)是指：页面的设计与开发应当根据用户行为以及设备环境(系统平台、屏幕尺寸、屏幕定向等)进行相应的响应和调整。可以说是一种网页设计的技术做法，该设计可使网站在多种浏览设备（从桌面电脑显示器到移动电话或其他移动产品设备）上阅读和导航，同时减少缩放、平移和滚动。

响应式网页设计就是一个网站能够兼容多个终端——而不是为每个终端做一个特定的版本。

响应式布局的实现

1. Meta标签定义

使用 **viewport meta** 标签在手机上控制布局

```
1. <meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1" />
```

隐藏状态栏

```
1. <meta name="apple-mobile-web-app-status-bar-style" content="blank" />
```

iPhone会将看起来像是电话号码的数字添加链接，需要关闭

```
1. <meta name="format-detection content="telephone=no" />
```

2.使用Media Queries适配对应样式

常用于布局的CSS Media Queries

设备类型

- all 所有设备；
- screen 电脑显示器；
- int 打印用纸或打印预览视图；
- ndheld 便携设备；
- 电视机类型的设备；
- eech 语意和音频盒成器；
- aille 盲人用点字法触觉回馈设备；
- bossed 盲文打印机；
- ojection 各种投影设备；
- tty 使用固定密度字母栅格的媒介，比如电传打字机和终端。

设备特性

- width 浏览器宽度；
- height 浏览器高度；
- device-width 设备屏幕分辨率的宽度值；
- device-height 设备屏幕分辨率的高度值；
- orientation 浏览器窗口的方向纵向还是横向，当窗口的高度值大于等于宽度时该特性值为portrait，否则为landscape；
- aspect-ratio 比例值，浏览器的纵横比；
- device-aspect-ratio 比例值，屏幕的纵横比。

实现响应式布局

通过媒体查询

```
1. @media screen and (max-width:320px) { }
2. @media screen and (min-width:320px) and (max-width:640px) { }
3. 样式表也可以引入外部的
4. @import url("css/demo.css") screen and (min-width:320px) and (max-width:640px);
5. @media screen and (min-width:640px) { }
```

当然，工作中是使用的外部样式表

1. `<link rel="stylesheet" type="text/css" href="a.css" media="screen and (min-width: 1000px)">`
- 2.
3. `<link rel="stylesheet" type="text/css" href="b.css" media="screen and (min-width: 600px) and (max-width: 1000px)">`
- 4.
5. `<link rel="stylesheet" type="text/css" href="c.css" media="screen and (max-width: 600px)">`

也许，你需要让手机的屏幕横着，比如你在玩 [王者荣耀](#)

1. 竖屏- 屏幕宽度小于高度
2. `<link rel="stylesheet" media="all and (orientation:portrait)" href="portrait.css">`
- 3.
4. 横屏-屏幕宽度大于高度
5. `<link rel="stylesheet" media="all and (orientation:landscape)" href="portrait.css">`

响应式布局缺点和优点

- 优点：面对不同分辨率设备，灵活性强，能够快捷地解决设备显示适应问题。
- 缺点：兼容各种设备时所需工作量大、效率低下、代码累赘，会隐藏无用的元素，加载时间延长，其实这是一种折中性质的设计解决方案，由于多方面元素影响而达不到最佳效果，在一定程度上改变了网站原有的布局结构，会出现用户混淆的情况。

这里有一个很严重的缺点

由于PC端和移动终端访问的是同一个网站，PC端可以不计较流量限制，但是移动端不可能不计较。

这里主要是针对于 [图片](#) 的使用，为适配不同终端机型的屏幕宽度和像素密度，我们一般会使用如下方法设置图片的CSS样式：

1. `<style>`
2. `img{`
3. `max-width:100%;`
4. `height:auto;`
5. `}`
6. `</style>`

将图片的最大宽度设置为100%，以确保图像不会超出其父级元素的宽度，如果父级元素的宽度发生改变，图片的宽度也随之改变，height : auto 可以确保图片的宽度发生改变时，图片的高度会依据自身的宽高比例进行缩放。

这样当我们在移动设备上访问响应式网页里的图片时，只是把图片的分辨率做了缩放，下载的还是PC端的那张大图，这样不仅浪费流量，而且浪费带宽，而且会拖慢网页的打开速度，严重影响用户的使用体验。

解决方案：<picture>

如下栗子中针对不同屏幕宽度加载不同的图片；当页面宽度 在320px到640px之间时加载 minpic.png；当页面宽度大于640px时加载middle.png

```
1. <picture>
2.     <source media="(min-width: 320px) and (max-width: 640px)" srcset="img/minpic.png">
3.     <source media="(min-width: 640px)" srcset="img/middle.png">
4.     
5. </picture>
```

如下例子中添加了屏幕的方向作为条件；当屏幕方向为横屏方向时加载_landscape.png结尾的图片；当屏幕方向为竖屏方向时加载 _portrait.png结尾的图片；

```
1. <picture>
2.     <source media="(min-width: 320px) and (max-width: 640px) and (orientation: landscape)" srcset="img/minpic_landscape.png">
3.     <source media="(min-width: 320px) and (max-width: 640px) and (orientation: portrait)" srcset="img/minpic_portrait.png">
4.     <source media="(min-width: 640px) and (orientation: landscape)" srcset="img/middlepic_landscape.png">
5.     <source media="(min-width: 640px) and (orientation: portrait)" srcset="img/middlepic_portrait.png">
6.     
7. </picture>
```

