



HTML



媒体特性与响应式设计

讲师：许井龙

ngsteel@qq.com

- 一. 响应式布局简介
- 二. meta标签及视口 (viewport)
- 三. 媒体类型
- 四. 媒体特性
- 五. IE兼容性 (支持CSS3 Media Queries)
- 六. rem
- 七. 渐进增强和优雅降级
- 八. 浏览器使用情况 (百度统计)



随着科技的进步，网页设备终端也越来越多样化。例如智能手机，平板电脑等。如果每个终端都设计一个网页显然不切实际。但是我们**至少要确保不同屏幕尺寸和不同设备上网页看起来是一模一样的。**

- 响应式布局不是新的技术

- Ethan Marcotte 在 A List Apart发表了一篇开创性的文章，将三种已有的开发技术（弹性网格布局、弹性图片、媒体和媒体查询）整合起来，并将其命名为RWD（Responsive Web Design，响应式设计）

- 响应式布局的特点

1. 网站必须灵活建立网格基础。
2. 引用到网站的图片必须是可伸缩性的。
3. 不同设备，需要在Media Query上设置不同的样式

响应式网站赏析

1. 国外 - <http://colly.com/>
2. 国内 - <https://www.wunderlist.com>
3. 国内 - <http://www.micourse.net/>

- 1、**流体网格**：网格大小可以随着屏幕尺寸大小做出相对应的比例缩放。
- 2、**弹性图片**：不给图片设定固定尺寸，根据流体网格自动缩放。一条代码即可搞定：`img{max-width: 100%}`
- 3、**媒体查询 (Media Query)**：响应式设计的灵魂，可以根据设备尺寸，匹配不同的样式。
- 4、**屏幕分辨率**：利用媒体查询对不同分辨率手机设置不同的样式。
- 5、**主要断点**：WEB开发中的新名词，在响应式设计中尤为重要。简单说就是设备的临界点。利用媒体查询针对不同尺寸终端设备设置不同样式。

- **响应式布局技巧：首先要让页面布局尽量简单，实现简单布局的技巧。**
 - 尽量少用无关紧要的div.
 - 不要使用内联元素（没法设置宽度）
 - 尽量少用JS或者flash
 - 丢弃没用的绝对定位或者相对定位
 - 摒弃任何冗余结构
- **哪些方法可以帮助我们更好的实现响应式设计**
 - 使用HTML5 Doctype和相关指南
 - 重置好样式（reset.css）
 - 一个简单的有语义的核心布局
 - 不要过分依赖现代技巧来实现，比如CSS3特效或者JS脚本

- 如何检查HTML结构是“简单干净”的？
 - 禁掉所有的CSS样式，如果内容排列有序，方便阅读，那么这个结构就不会差到哪里去。

1. **视觉视口**: **用户正在看到的网站的区域**，一般**视觉视口**和**布局视口**一致。并且它的CSS像素的数量会随着用户缩放而改变。
 - ◆ 可通过 `window.innerWidth/innerHeight` 获取;
2. **布局视口**: 不再与移动端浏览器相关联，完全是独立的。实际上**布局视口的宽度要比屏幕宽出很多**。了容纳为桌面浏览器设计的网站， 移动设备默认的布局视口宽度远大于屏幕的宽度， 设置为 **980px**或1024px（也可能是其它值，这个是由设备自己决定的）， 但带来的后果就是浏览器会出现横向滚动条， 因为浏览器可视区域的宽度是比这个默认的viewport的宽度要小的
 - ◆ 可通过 `document.documentElement.clientWidth/clientHeight`; 获取;
3. **理想视口**: 布局视口的默认宽度并不是一个理想的宽度。显然用户希望在进入页面时可以不需要缩放就可以有一个理想的浏览和阅读尺寸。理想视口仍是为移动端设备准备的。只有手动添加meta视口标签方才生效。如果没有meta视口标签，那么布局将会维持它的默认宽度。
 - ◆ **让网页的宽度自动适应手机屏幕的宽度。**

二、meta标签及视口 (viewport)

- `<meta name=" viewport" content=" " >` **告诉浏览器如何处理网页**。该标签主要作用让移动设备在浏览网页时进行优化，并且可以自定义界面可视区域的尺寸与缩放级别。
- viewport: 智能手机使用了一个比实际屏幕大的虚拟可视区域 (viewport)，主要的目的是让页面在智能手机阅读时不会因为实际可视区域而变形。

二、meta标签及视口 (viewport)

- `<meta name="viewport" content=" " >`
 - **width** : 可视区域宽度, 可以是一个具体的数字也可以是device-width。 **可以让网页的宽度自动适应手机屏幕的宽度**
 - **height** : 可视区域高度, 可以是一个具体的数字也可以是device-height。
 - **initial-scale**: 页面首次显示时, 可是区域的缩放比例, 取值1.0页面按实际比列显示, 无任何缩放。
 - **minimun-scale**: 可视区域最小缩放级别, 取值1.0禁止用户缩放至实际尺寸以下。
 - **maximun-sacle**: 可视区域最大缩放级别, 取值1.0禁止用户放大至实际尺寸以上。
 - **user-scalable**: 用户是否可以对页面进行缩放。yes可以缩放, no禁止缩放。

移动端的网页需要设置如下meta

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
```

媒体类型定义

值	描述
<i>all</i>	用于所有设备
<i>print</i>	用于打印机和打印预览
<i>screen</i>	用于电脑屏幕，平板电脑，智能手机等。
<i>speech</i>	应用于屏幕阅读器等发声设备
<i>aural</i>	已废弃。用于语音和声音合成器
<i>braille</i>	已废弃。 应用于盲文触摸式反馈设备
<i>embossed</i>	已废弃。 用于打印的盲人印刷设备
<i>handheld</i>	已废弃。 用于掌上设备或更小的装置，如PDA和小型电话
<i>projection</i>	已废弃。 用于投影设备
<i>tty</i>	已废弃。 用于固定的字符网格，如电报、终端设备和对字符有限制的便携设备
<i>tv</i>	已废弃。 用于电视和网络电视

媒体类型使用

在打印设备中，使用`print.css`里定义的样式

```
<link rel="stylesheet" type="text/css" href="print.css" media="print">
```

直接通过`@media`选择器（Media Query）定义在`<style>`元素中

```
@media print {  
    /*样式代码 */  
}
```

min-width: 768px

max-width: 1024px



断点
768px

断点
1024px

- 媒体属性 (**Media features**) : 大多数媒体属性带有“min-”和“max-”前缀, 用于表达“小于等于”和“大于等于”。这避免了使用与HTML和XML冲突的“<”和“>”字符。如果你未向媒体属性指定一个值, 并且该特性的实际值不为零, 则该表达式被解析为真。
 - **width**
 - **max-width**
 - **min-width**
 - **orientation**
 - **landscape** 横屏
 - **portrait** 竖屏

- Media Query 使用方法

- @media 媒体类型 and (媒体特性) {样式}

```
@media screen and (min-width: 768px) {  
    /** 执行代码 样式*/  
}
```

768px 是断点, 768px下屏幕尺寸

```
@media screen and (min-width: 640px) and (max-width: 768px) {  
    /** 执行代码 样式*/  
}
```


- rem (root em) , 与px , em 一样也是一个单位。不同的是它是基于html元素 (html = root element) 计算。
- 该单位常与媒体查询一起使用。
- 优点是：只要改变html的font-size大小，所有基于该元素计算的元素相关尺寸随之等比改变。

- rem 缺点
 - 1rem = 16px 需要进行换算，非常麻烦。
 - 解决办法： `html{ font-size: 62.5%}`, 此时可以使用1.2rem表示12px, 计算比较方便。但是致命的问题随之而来，Chrome浏览器最小字体是12px，如果设置了11px及一下，仍显示12px。
 - 目前普遍被接收的解决办法： `html{ font-size: 625%}`。但是这种方法使用不是很方便。

- rem 缺点
 - 1rem = 16px 需要进行换算，非常麻烦。
 - 解决办法： `html{ font-size: 62.5%}`, 此时可以使用1.2rem表示12px, 计算比较方便。但是致命的问题随之而来，Chrome浏览器最小字体是12px，如果设置了11px及一下，仍显示12px。
 - 目前普遍被接收的解决办法： `html{ font-size: 625%}`。但是这种方法使用不是很方便。

- px: 在PC端使用, 移动端谨慎使用。
- em: 避免嵌套使用, **基于自身字体的大小**。
- %: 避免嵌套使用, **基于父元素**。
- rem: **基于html元素字体大小**。注意Chrome不支持12px以下像素

.2em

.5em

2em

- border-box
- content-box

- **渐进增强**：针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。
- **优雅降级**：一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

- 第三方插件：
 - <https://github.com/scottjehl/Respond>
- 注意事项：
 - 1、需要在服务器上运行。
 - 2、包含媒体查询的 css文件需采用外链形式。
 - 3、**头部引用的respond.js 需置于css 文件之后。**