HTML 基础

目标

- 认识 HTML 的基本结构
- 学会使用常用的 HTML 标签

HTML 结构

认识 HTML 标签

HTML 代码是由 "标签" 构成的.

形如:

```
<body>hello</body>
```

- 标签名 (body) 放到 < > 中
- 大部分标签成对出现. `<body>` 为开始标签, `</body>` 为结束标签.
- 少数标签只有开始标签, 称为 "单标签".
- 开始标签和结束标签之间, 写的是标签的内容. (hello)
- 开始标签中可能会带有 "属性". id 属性相当于给这个标签设置了一个唯一的标识符(身份证号码).

```
<body id="myId">hello</body>
```

HTML 文件基本结构

```
<html>
    <head>
        <title>第一个页面</title>
    </head>
    <body>
        hello world
    </body>
</html>
```

- html 标签是整个 html 文件的根标签(最顶层标签)
- head 标签中写页面的属性.
- body 标签中写的是页面上显示的内容
- title 标签中写的是页面的标题.

标签层次结构

- 父子关系

- 兄弟关系

```
<html>
    <head>
        <title>第一个页面</title>
    </head>
    <body>
        hello world
    </body>
</html>
```

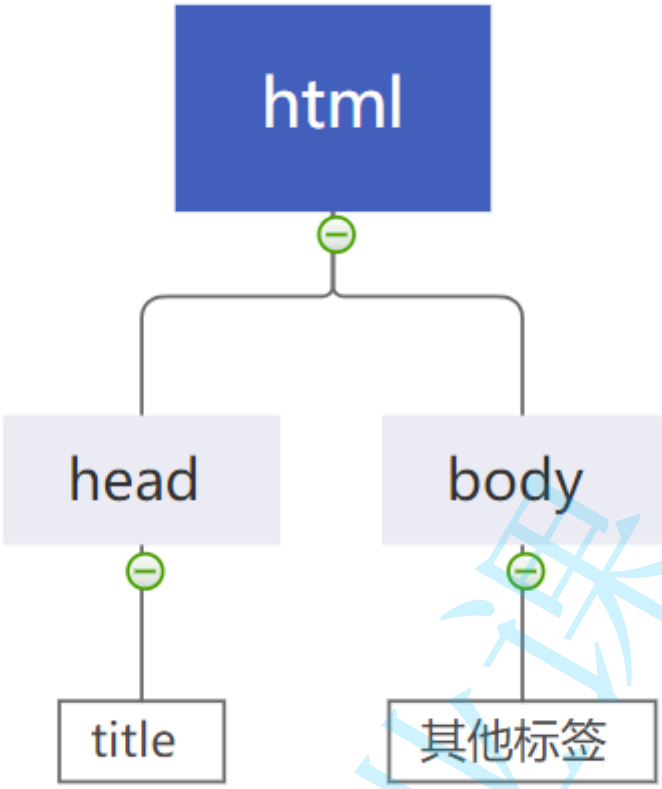
其中:

- head 和 body 是 html 的子标签(html 就是 head 和 body 的父标签)
- title 是 head 的子标签. head 是 title 的父标签.
- head 和 body 之间是兄弟关系.

> 可以使用 chrome 的开发者工具查看页面的结构.
>
> F12 或者右键审查元素, 开启开发者工具, 切换到 Elements 标签, 就可以看到页面结构细节.

标签之间的结构关系, 构成了一个 **DOM 树**

> DOM 是 Document Object Mode (文档对象模型) 的缩写.



快速生成代码框架

在 IDEA 中创建文件 `xxx.html`, 直接输入 !, 按 tab 键, 此时能自动生成代码的主体框架.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

细节解释: (了解即可, 不必深究)

- `<!DOCTYPE html>` 称为 DTD (文档类型定义), 描述当前的文件是一个 HTML5 的文件.
- `<html lang="en">` 其中 lang 属性表示当前页面是一个 "英语页面". 这里暂时不用管. (有些浏览器会根据此处的声明提示是否进行自动翻译).
- `<meta charset="UTF-8">` 描述页面的字符编码方式. 没有这一行可能会导致中文乱码.
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
  - `name="viewport"` 其中 viewport 指的是设备的屏幕上能用来显示我们的网页的那一块区域.
  - `content="width=device-width, initial-scale=1.0"` 在设置可视区和设备宽度等宽, 并设置初始缩放为不缩放. (这个属性对于移动端开发更重要一些).

HTML 常见标签

注释标签

注释不会显示在界面上. 目的是提高代码的可读性.

```
<!-- 我是注释 -->
```

> ctrl + / 快捷键可以快速进行注释/取消注释.

注释的原则

- 要和代码逻辑一致.
- 尽量使用中文.
- 不要传递负能量.

标题标签: h1-h6

有六个, 从 h1 - h6. 数字越大, 则字体越小.

```
<h1>hello</h1>
<h2>hello</h2>
<h3>hello</h3>
<h4>hello</h4>
<h5>hello</h5>
<h6>hello</h6>
```

段落标签: p

把一段比较长的文本粘贴到 html 中, 会发现并没有分成段落.

例如以下文本:

```
    css中的1px并不等于设备的1px

    在css中我们一般使用px作为单位，在桌面浏览器中css的1个像素往往都是对应着电脑屏幕的1个物理像素，这可能会造成我们的一个错觉，那就是css中的像素就是设备的物理像素。但实际情况却并非如此，css中的像素只是一个抽象的单位，在不同的设备或不同的环境中，css中的1px所代表的设备物理像素是不同的。在为桌面浏览器设计的网页中，我们无需对这个津津计较，但在移动设备上，必须弄明白这点。在早先的移动设备中，屏幕像素密度都比较低，如iphone3，它的分辨率为320x480，在iphone3上，一个css像素确实是等于一个屏幕物理像素的。后来随着技术的发展，移动设备的屏幕像素密度越来越高，从iphone4开始，苹果公司便推出了所谓的Retina屏，分辨率提高了一倍，变成640x960，但屏幕尺寸却没变化，这就意味着同样大小的屏幕上，像素却多了一倍，这时，一个css像素是等于两个物理像素的。其他品牌的移动设备也是这个道理。例如安卓设备根据屏幕像素密度可分为ldpi、mdpi、hdpi、xhdpi等不同的等级，分辨率也是五花八门，安卓设备上的一个css像素相当于多少个屏幕物理像素，也因设备的不同而不同，没有一个定论。

    还有一个因素也会引起css中px的变化，那就是用户缩放。例如，当用户把页面放大一倍，那么css中1px所代表的物理像素也会增加一倍；反之把页面缩小一倍，css中1px所代表的物理像素也会减少一倍。关于这点，在文章后面的部分还会讲到。

    在移动端浏览器中以及某些桌面浏览器中，window对象有一个devicePixelRatio属性，它的官方的定义为：设备物理像素和设备独立像素的比例，也就是 devicePixelRatio = 物理像素 /
    独立像素。css中的px就可以看做是设备的独立像素，所以通过devicePixelRatio，我们可以知道该设备上一个css像素代表多少个物理像素。例如，在Retina屏的iphone上，devicePixelRatio的值为2，也就是说1个css像素相当于2个物理像素。但是要注意的是，devicePixelRatio在不同的浏览器中还存在些许的兼容性问题，所以我们现在还并不能完全信赖这个东西，具体的情况可以看下这篇文章。
```

展示结果:

css中的1px并不等于设备的1px 在css中我们一般使用px作为单位，在桌面浏览器中css的1个像素往往都是对应着电脑屏幕的1个物理像素，这可能会造成我们的一个错觉，那就是css中的像素就是设备的物理像素。但实际情况却并非如此，css中的像素只是一个抽象的单位，在不同的设备或不同的环境中，css中的1px所代表的设备物理像素是不同的。在为桌面浏览器设计的网页中，我们无需对这个津津计较，但在移动设备上，必须弄明白这点。在早先的移动设备中，屏幕像素密度都比较低，如iphone3，它的分辨率为320x480，在iphone3上，一个css像素确实是等于一个屏幕物理像素的。后来随着技术的发展，移动设备的屏幕像素密度越来越高，从iphone4开始，苹果公司便推出了所谓的Retina屏，分辨率提高了一倍，变成640x960，但屏幕尺寸却没变化，这就意味着同样大小的屏幕上，像素却多了一倍，这时，一个css像素是等于两个物理像素的。其他品牌的移动设备也是这个道理。例如安卓设备根据屏幕像素密度可分为ldpi、mdpi、hdpi、xhdpi等不同的等级，分辨率也是五花八门，安卓设备上的一个css像素相当于多少个屏幕物理像素，也因设备的不同而不同，没有一个定论。 还有一个因素也会引起css中px的变化，那就是用户缩放。例如，当用户把页面放大一倍，那么css中1px所代表的物理像素也会增加一倍；反之把页面缩小一倍，css中1px所代表的物理像素也会减少一倍。关于这点，在文章后面的部分还会讲到。 在移动端浏览器中以及某些桌面浏览器中，window对象有一个devicePixelRatio属性，它的官方的定义为：设备物理像素和设备独立像素的比例，也就是 devicePixelRatio = 物理像素 / 独立像素。css中的px就可以看做是设备的独立像素，所以通过devicePixelRatio，我们可以知道该设备上一个css像素代表多少个物理像素。例如，在Retina屏的iphone上，devicePixelRatio的值为2，也就是说1个css像素相当于2个物理像素。但是要注意的是，devicePixelRatio在不同的浏览器中还存在些许的兼容性问题，所以我们现在还并不能完全信赖这个东西，具体的情况可以看下这篇文章。

- p 标签表示一个段落.

```
<p>这是一个段落</p>
```

通过 p 标签改进上述代码, 每个段落放到 p 标签中.

```
<p>css中的1px并不等于设备的1px</p>

<p>
```

```
    在css中我们一般使用px作为单位，在桌面浏览器中css的1个像素往往都是对应着电脑屏幕的1个物理像素，这可能会造成我们的一个错觉，那就是css中的像素就是设备的物理像素。但实际情况却并非如此，css中的像素只是一个抽象的单位，在不同的设备或不同的环境中，css中的1px所代表的设备物理像素是不同的。在为桌面浏览器设计的网页中，我们无需对这个津津计较，但在移动设备上，必须弄明白这点。在早先的移动设备中，屏幕像素密度都比较低，如iphone3，它的分辨率为320x480，在iphone3上，一个css像素确实是等于一个屏幕物理像素的。后来随着技术的发展，移动设备的屏幕像素密度越来越高，从iphone4开始，苹果公司便推出了所谓的Retina屏，分辨率提高了一倍，变成640x960，但屏幕尺寸却没变化，这就意味着同样大小的屏幕上，像素却多了一倍，这时，一个css像素是等于两个物理像素的。其他品牌的移动设备也是这个道理。例如安卓设备根据屏幕像素密度可分为ldpi、mdpi、hdpi、xhdpi等不同的等级，分辨率也是五花八门，安卓设备上的一个css像素相当于多少个屏幕物理像素，也因设备的不同而不同，没有一个定论。
</p>

<p>
    还有一个因素也会引起css中px的变化，那就是用户缩放。例如，当用户把页面放大一倍，那么css中1px所代表的物理像素也会增加一倍；反之把页面缩小一倍，css中1px所代表的物理像素也会减少一倍。关于这点，在文章后面的部分还会讲到。
</p>

<p>
    在移动端浏览器中以及某些桌面浏览器中，window对象有一个devicePixelRatio属性，它的官方的定义为：设备物理像素和设备独立像素的比例，也就是 devicePixelRatio = 物理像素 /
    独立像素。css中的px就可以看做是设备的独立像素，所以通过devicePixelRatio，我们可以知道该设备上一个css像素代表多少个物理像素。例如，在Retina屏的iphone上，devicePixelRatio的值为2，也就是说1个css像素相当于2个物理像素。但是要注意的是，devicePixelRatio在不同的浏览器中还存在些许的兼容性问题，所以我们现在还并不能完全信赖这个东西，具体的情况可以看下这篇文章。
</p>
```

展示结果:

css中的1px并不等于设备的1px

在css中我们一般使用px作为单位，在桌面浏览器中css的1个像素往往都是对应着电脑屏幕的1个物理像素，这可能会造成我们的一个错觉，那就是css中的像素就是设备的物理像素。但实际情况却并非如此，css中的像素只是一个抽象的单位，在不同的设备或不同的环境中，css中的1px所代表的设备物理像素是不同的。在为桌面浏览器设计的网页中，我们无需对这个津津计较，但在移动设备上，必须弄明白这点。在早先的移动设备中，屏幕像素密度都比较低，如iphone3，它的分辨率为320x480，在iphone3上，一个css像素确实是等于一个屏幕物理像素的。后来随着技术的发展，移动设备的屏幕像素密度越来越高，从iphone4开始，苹果公司便推出了所谓的Retina屏，分辨率提高了一倍，变成640x960，但屏幕尺寸却没变化，这就意味着同样大小的屏幕上，像素却多了一倍，这时，一个css像素是等于两个物理像素的。其他品牌的移动设备也是这个道理。例如安卓设备根据屏幕像素密度可分为ldpi、mdpi、hdpi、xhdpi等不同的等级，分辨率也是五花八门，安卓设备上的一个css像素相当于多少个屏幕物理像素，也因设备的不同而不同，没有一个定论。

还有一个因素也会引起css中px的变化，那就是用户缩放。例如，当用户把页面放大一倍，那么css中1px所代表的物理像素也会增加一倍；反之把页面缩小一倍，css中1px所代表的物理像素也会减少一倍。关于这点，在文章后面的部分还会讲到。

在移动端浏览器中以及某些桌面浏览器中，window对象有一个devicePixelRatio属性，它的官方的定义为：设备物理像素和设备独立像素的比例，也就是 devicePixelRatio = 物理像素 / 独立像素。css中的px就可以看做是设备的独立像素，所以通过devicePixelRatio，我们可以知道该设备上一个css像素代表多少个物理像素。例如，在Retina屏的iphone上，devicePixelRatio的值为2，也就是说1个css像素相当于2个物理像素。但是要注意的是，devicePixelRatio在不同的浏览器中还存在些许的兼容性问题，所以我们现在还并不能完全信赖这个东西，具体的情况可以看下这篇文章。

注意:

- p 标签之间存在一个空隙
- 当前的 p 标签描述的段落, 前面还没有缩进. (未来 CSS 会学)
- 自动根据浏览器宽度来决定排版.
- html 内容首尾处的换行, 空格均无效.
- 在 html 中文字之间输入的多个空格只相当于一个空格.
- html 中直接输入换行不会真的换行, 而是相当于一个空格.

换行标签: br

br 是 break 的缩写. 表示换行.

- br 是一个单标签(不需要结束标签)
- br 标签不像 p 标签那样带有一个很大的空隙.
- `<br/>` 是规范写法. 不建议写成 `<br>`

```
<p>
    在css中我们一般使用px作为单位，<br/>
    在桌面浏览器中css的1个像素往往都是对应着电脑屏幕的1个物理像素，这可能会造成我们的一个错觉，那就是css中的像素就是设备的物理像素。但实际情况却并非如此，css中的像素只是一个抽象的单位，在不同的设备或不同的环境中，css中的1px所代表的设备物理像素是不同的。在为桌面浏览器设计的网页中，我们无需对这个津津计较，但在移动设备上，必须弄明白这点。在早先的移动设备中，屏幕像素密度都比较低，如iphone3，它的分辨率为320x480，在iphone3上，一个css像素确实是等于一个屏幕物理像素的。后来随着技术的发展，移动设备的屏幕像素密度越来越高，从iphone4开始，苹果公司便推出了所谓的Retina屏，分辨率提高了一倍，变成640x960，但屏幕尺寸却没变化，这就意味着同样大小的屏幕上，像素却多了一倍，这时，一个css像素是等于两个物理像素的。其他品牌的移动设备也是这个道理。例如安卓设备根据屏幕像素密度可分为ldpi、mdpi、hdpi、xhdpi等不同的等级，分辨率也是五花八门，安卓设备上的一个css像素相当于多少个屏幕物理像素，也因设备的不同而不同，没有一个定论。
</p>
```

展示结果:

在css中我们一般使用px作为单位，
在桌面浏览器中css的1个像素往往都是对应着电脑屏幕的1个物理像素，这可能会造成我们的一个错觉，那就是css中的像素就是设备的物理像素。但实际情况却并非如此，css中的像素只是一个抽象的单位，在不同的设备或不同的环境中，css中的1px所代表的设备物理像素是不同的。在为桌面浏览器设计的网页中，我们无需对这个津津计较，但在移动设备上，必须弄明白这点。在早先的移动设备中，屏幕像素密度都比较低，如iphone3，它的分辨率为320x480，在iphone3上，一个css像素确实是等于一个屏幕物理像素的。后来随着技术的发展，移动设备的屏幕像素密度越来越高，从iphone4开始，苹果公司便推出了所谓的Retina屏，分辨率提高了一倍，变成640x960，但屏幕尺寸却没变化，这就意味着同样大小的屏幕上，像素却多了一倍，这时，一个css像素是等于两个物理像素的。其他品牌的移动设备也是这个道理。例如安卓设备根据屏幕像素密度可分为ldpi、mdpi、hdpi、xhdpi等不同的等级，分辨率也是五花八门，安卓设备上的一个css像素相当于多少个屏幕物理像素，也因设备的不同而不同，没有一个定论。

综合案例: 展示博客

案例的文本内容出自 https://www.cnblogs.com/yelongsan/p/7975580.html

预期效果:

meta name="viewport" content="width=device-width,initial-scale=1.0" 解释

蒂其之死

简单来说 meta name="viewport" content="width=device-width,initial-scale=1.0" 解释

content属性值：
width:可视区域的宽度，值可为数字或关键词device-width
height:同width
intial-scale:页面首次被显示是可视区域的缩放级别，取值1.0则页面按实际尺寸显示，无任何缩放
maximum-scale=1.0, minimum-scale=1.0,可视区域的缩放级别，
maximum-scale用户可将页面放大的程序，1.0将禁止用户放大到实际尺寸之上。
user-scalable:是否可对页面进行缩放，no 禁止缩放

复杂的说：
移动前端开发之viewport的深入理解
在移动设备上进行网页的重构或开发，首先得搞明白的就是移动设备上的viewport了，只有明白了viewport的概念以及弄清楚了跟viewport有关的meta标签的使用，才能更好地让我们的网页适配或响应各种不同分辨率的移动设备。

一、viewport的概念

通俗的讲，移动设备上的viewport就是设备的屏幕上能用来显示我们的网页的那一块区域，在具体一点，就是浏览器上(也可能是一个app中的webview)用来显示网页的那部分区域，但viewport又不局限于浏览器可视区域的大小，它可能比浏览器的可视区域要大，也可能比浏览器的可视区域要小。在默认情况下，一般来讲，移动设备上的viewport都是要大于浏览器可视区域的，这是因为考虑到移动设备的分辨率相对于桌面电脑来说都比较小，所以为了能在移动设备上正常显示那些传统的为桌面浏览器设计的网站，移动设备上的浏览器都会把自己默认的viewport设为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。下图列出了一些设备上浏览器的默认viewport的宽度。

二、css中的1px并不等于设备的1px

在css中我们一般使用px作为单位，在桌面浏览器中css的1个像素往往都是对应着电脑屏幕的1个物理像素，这可能会造成我们的一个错觉，那就是css中的像素就是设备的物理像素。但实际情况却并非如此，css中的像素只是一个抽象的单位，在不同的设备或不同的环境中，css中的1px所代表的设备物理像素是不同的。在为桌面浏览器设计的网页中，我们无需对这个津津计较，但在移动设备上，必须弄明白这点。在早先的移动设备中，屏幕像素密度都比较低，如iphone3，它的分辨率为320x480，在iphone3上，一个css像素确实是等于一个屏幕物理像素的。后来随着技术的发展，移动设备的屏幕像素密度越来越高，从iphone4开始，苹果公司便推出了所谓的Retina屏，分辨率提高了一倍，变成640x960，但屏幕尺寸却没变化，这就意味着同样大小的屏幕上，像素却多了一倍，这时，一个css像素是等于两个物理像素的。其他品牌的移动设备也是这个道理。例如安卓设备根据屏幕像素密度可分为ldpi、mdpi、hdpi、xhdpi等不同的等级，分辨率也是五花八门，安卓设备上的一个css像素相当于多少个屏幕物理像素，也因设备的不同而不同，没有一个

提示:

- 给标题, 作者, 小标题部分加上合适的标题.
- 给每个段落加上合适的段落.

- 给需要换行的地方加上 br 标签

格式化标签

- 加粗: strong 标签 和 b 标签
- 倾斜: em 标签 和 i 标签
- 删除线: del 标签 和 s 标签
- 下划线: ins 标签 和 u 标签

```
<strong>strong 加粗</strong>
<b>b 加粗</b>

<em>倾斜</em>
<i>倾斜</i>

<del>删除线</del>
<s>删除线</s>

<ins>下划线</ins>
<u>下划线</u>
```

---

strong 加粗 b 加粗 *倾斜 倾斜* ~~删除线 删除线~~ 下划线 下划线

> 使用 CSS 也可以完成类似的效果. 实际开发中以 CSS 方式为主.

图片标签: img

img 标签必须带有 src 属性. 表示图片的路径.

```
<img src="rose.jpg">
```

此时要把 rose.jpg 这个图片文件放到和 html 中的同级目录中.

img 标签的其他属性

- alt: 替换文本. 当文本不能正确显示的时候, 会显示一个替换的文字.
- title: 提示文本. 鼠标放到图片上, 就会有提示.
- width/height: 控制宽度高度. 高度和宽度一般改一个就行, 另外一个会等比例缩放. 否则就会图片失衡.
- border: 边框, 参数是宽度的像素. 但是一般使用 CSS 来设定.

```
<img src="rose.jpg" alt="鲜花" title="这是一朵鲜花" width="500px" height="800px"
border="5px">
```

> **注意:**
>
> 1. 属性可以有多个, 不能写到标签之前
> 2. 属性之间用空格分割, 可以是多个空格, 也可以是换行.

> 3. 属性之间不分先后顺序
> 4. 属性使用 "键值对" 的格式来表示.

关于目录结构:

对于一个复杂的网站, 页面资源很多, 这种情况可以使用目录把这些文件整理好.

1) **相对路径:** 以 html 所在位置为基准, 找到图片的位置.

- 同级路径: 直接写文件名即可 (或者 ./)
- 下一级路径: image/1.jpg
- 上一级路径: ../image/1.jpg

2) **绝对路径:** 一个完整的磁盘路径, 或者网络路径. 例如

- 磁盘路径 `D:\rose.jpg`
- 网络路径 `https://images0.cnblogs.com/blog/130623/201407/300958470402077.png`

代码示例

1) 使用相对路径: 创建一个 image 目录和 html 同级, 并放入一个 rose2.jpg

```
<img src="image/rose2.jpg" alt="">
```

2) 使用相对路径2: 在 image 目录中创建一个 html, 并访问上级目录的 rose.jpg

```
<img src="../rose.jpg" alt="">
```

3) 使用绝对路径1: 最好使用 / , 不要使用 \

```
<img src="D:/rose.jpg" alt="">
```

4) 使用绝对路径2: 使用网络路径

```
<img src="https://images0.cnblogs.com/blog/130623/201407/300958475557219.png"
alt="">
```

超链接标签: a

- href: 必须具备, 表示点击后会跳转到哪个页面.
- target: 打开方式. 默认是 _self. 如果是 _blank 则用新的标签页打开.

```
<a href="http://www.baidu.com">百度</a>
```

链接的几种形式:

- 外部链接: href 引用其他网站的地址

```
<a href="http://www.baidu.com">百度</a>
```

- 内部链接: 网站内部页面之间的链接. 写相对路径即可.

> 在一个目录中, 先创建一个 1.html, 再创建一个 2.html

```
<!-- 1.html -->
我是 1.html
<a href="2.html">点我跳转到 2.html</a>

<!-- 2.html -->
我是 2.html
<a href="1.html">点我跳转到 1.html</a>
```

- 空链接: 使用 # 在 href 中占位.

```
<a href="#">空链接</a>
```

- 下载链接: href 对应的路径是一个文件. (可以使用 zip 文件)

```
<a href="test.zip">下载文件</a>
```

- 网页元素链接: 可以给图片等任何元素添加链接(把元素放到 a 标签中)

```
<a href="http://www.sogou.com">
    <img src="rose.jpg" alt="">
</a>
```

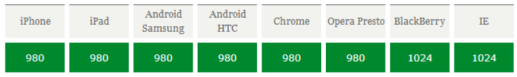
- 锚点链接: 可以快速定位到页面中的某个位置.

```
<a href="#one">第一集</a>
<a href="#two">第二集</a>
<a href="#three">第三集</a>
<p id="one">
    第一集剧情 <br>
    第一集剧情 <br>
    ...
</p>
<p id="two">
    第二集剧情 <br>
    第二集剧情 <br>
    ...
</p>
<p id="three">
    第三集剧情 <br>
    第三集剧情 <br>
    ...
</p>
```

> 禁止 a 标签跳转: `<a href="javascript:void(0);">` 或者`<a href="javascript:;">`

综合案例: 展示博客2

预期效果: 在上个博客页面的案例基础上, 加上图片和超链接



二、css中的1px并不等于设备的1px

在css中我们一般使用px作为单位，在桌面浏览器中css的1个像素往往都是对应着电脑屏幕的1个物理像素，这可能会造成我们的一个错觉，那就是css中的像素就是设备的物理像素。但实际情况却并非如此，css中的像素只是一个抽象的单位，在不同的设备或不同的环境中，css中的1px所代表的设备物理像素是不同的。在为桌面浏览器设计的网页中，我们无需对这个津津计较，但在移动设备上，必须弄明白这点。在早先的移动设备中，屏幕像素密度都比较低，如iphone3，它的分辨率为320x480，在iphone3上，一个css像素确实是等于一个屏幕物理像素的。后来随着技术的发展，移动设备的屏幕像素密度越来越高，从iphone4开始，苹果公司便推出了所谓的Retina屏，分辨率提高了一倍，变成640x960，但屏幕尺寸却没变化，这就意味着同样大小的屏幕上，像素却多了一倍，这时，一个css像素是等于两个物理像素的。其他品牌的移动设备也是这个道理。例如安卓设备根据屏幕像素密度可分为ldpi、mdpi、hdpi、xhdpi等不同的等级，分辨率也是五花八门，安卓设备上的一个css像素相当于多少个屏幕物理像素，也因设备的不同而不同，没有一个定论。

还有一个因素也会引起css中px的变化，那就是用户缩放。例如，当用户把页面放大一倍，那么css中1px所代表的物理像素也会增加一倍；反之把页面缩小一倍，css中1px所代表的物理像素也会减少一倍。关于这点，在文章后面的部分还会讲到。

在移动端浏览器中以及某些桌面浏览器中，window对象有一个devicePixelRatio属性，它的官方的定义为：设备物理像素和设备独立像素的比例，也就是 devicePixelRatio = 物理像素 / 独立像素。css中的px就可以看做是设备的独立像素，所以通过devicePixelRatio，我们可以知道该设备上一个css像素代表多少个物理像素。例如，在Retina屏的iphone上，devicePixelRatio的值为2，也就是说1个css像素相当于2个物理像素。但是要注意的是，devicePixelRatio在不同的浏览器中还存在些许的兼容性问题，所以我们现在还并不能完全信赖这个东西，具体的情况可以看下 这篇文章。

提示:

- 可以从页面上右键图片, 下载图片到本地.
- 可以从页面上右键链接, 复制链接, 粘贴到代码中.

参考代码:

```
<h1>meta name="viewport" content="width=device-width,initial-scale=1.0" 解释</h1>
<h3>蒂其之死</h3>

<p>
    简单来说 meta name="viewport" content="width=device-width,initial-scale=1.0"
解释
</p>

<p>
    content属性值 : <br>
    width:可视区域的宽度，值可为数字或关键词device-width<br>
    height:同width<br>
    intial-scale:页面首次被显示是可视区域的缩放级别，取值1.0则页面按实际尺寸显示，无任何缩放
<br>
    maximum-scale=1.0, minimum-scale=1.0;可视区域的缩放级别，<br>
    maximum-scale用户可将页面放大的程序，1.0将禁止用户放大到实际尺寸之上。<br>
    user-scalable:是否可对页面进行缩放，no 禁止缩放<br>
</p>

<p>
    复杂的说：<br>
    移动前端开发之viewport的深入理解<br>
    在移动设备上进行网页的重构或开发，首先得搞明白的就是移动设备上的viewport了，只有明白了
viewport的概念以及弄清楚了跟viewport有关的meta标签的使用，才能更好地让我们的网页适配或响应各
种不同分辨率的移动设备。<br>
</p>

<h2>
    一、viewport的概念
```

```
</h2>

<p>
    通俗的讲，移动设备上的viewport就是设备的屏幕上能用来显示我们的网页的那一块区域，在具体一点，就是浏览器上(也可能是一个app中的webview)用来显示网页的那部分区域，但viewport又不局限于浏览器可视区域的大小，它可能比浏览器的可视区域要大，也可能比浏览器的可视区域要小。在默认情况下，一般来讲，移动设备上的viewport都是要大于浏览器可视区域的，这是因为考虑到移动设备的分辨率相对于桌面电脑来说都比较小，所以为了能在移动设备上正常显示那些传统的为桌面浏览器设计的网站，移动设备上的浏览器都会把自己默认的viewport设为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。下图列出了一些设备上浏览器的默认viewport的宽度。
</p>
<img src="1.png" alt="">

<h2>
    二、css中的1px并不等于设备的1px
</h2>

<p>
    在css中我们一般使用px作为单位，在桌面浏览器中css的1个像素往往都是对应着电脑屏幕的1个物理像素，这可能会造成我们的一个错觉，那就是css中的像素就是设备的物理像素。但实际情况却并非如此，css中的像素只是一个抽象的单位，在不同的设备或不同的环境中，css中的1px所代表的设备物理像素是不同的。在为桌面浏览器设计的网页中，我们无需对这个津津计较，但在移动设备上，必须弄明白这点。在早先的移动设备中，屏幕像素密度都比较低，如iphone3，它的分辨率为320x480，在iphone3上，一个css像素确实是等于一个屏幕物理像素的。后来随着技术的发展，移动设备的屏幕像素密度越来越高，从iphone4开始，苹果公司便推出了所谓的Retina屏，分辨率提高了一倍，变成640x960，但屏幕尺寸却没变化，这就意味着同样大小的屏幕上，像素却多了一倍，这时，一个css像素是等于两个物理像素的。其他品牌的移动设备也是这个道理。例如安卓设备根据屏幕像素密度可分为ldpi、mdpi、hdpi、xhdpi等不同的等级，分辨率也是五花八门，安卓设备上的一个css像素相当于多少个屏幕物理像素，也因设备的不同而不同，没有一个定论。
</p>

<p>
    还有一个因素也会引起css中px的变化，那就是用户缩放。例如，当用户把页面放大一倍，那么css中1px所代表的物理像素也会增加一倍；反之把页面缩小一倍，css中1px所代表的物理像素也会减少一倍。关于这点，在文章后面的部分还会讲到。
</p>

<p>
    在移动端浏览器中以及某些桌面浏览器中，window对象有一个devicePixelRatio属性，它的官方的定义为：设备物理像素和设备独立像素的比例，也就是 devicePixelRatio = 物理像素 /
    独立像素。css中的px就可以看做是设备的独立像素，所以通过devicePixelRatio，我们可以知道该设备上一个css像素代表多少个物理像素。例如，在Retina屏的iphone上，devicePixelRatio的值为2，也就是说1个css像素相当于2个物理像素。但是要注意的是，devicePixelRatio在不同的浏览器中还存在些许的兼容性问题，所以我们现在还并不能完全信赖这个东西，具体的情况可以看下
    <a
href="http://www.quirksmode.org/blog/archives/2012/06/devicepixelrati.html">这篇文章</a>。
</p>
<img src="2.png" alt="">
```

表格标签

基本使用

- table 标签: 表示整个表格
- tr: 表示表格的一行
- td: 表示一个单元格
- th: 表示表头单元格. 会居中加粗
- thead: 表格的头部区域(注意和 th 区分, 范围是比 th 要大的)
- tbody: 表格得到主体区域.

table 包含 tr , tr 包含 td 或者 th.

> 表格标签有一些属性, 可以用于设置大小边框等. 但是**一般使用 CSS 方式**来设置.
>
> 这些属性都要放到 table 标签中.
>
> - align 是表格相对于周围元素的对齐方式. `align="center"` (不是内部元素的对齐方式)
> - border 表示边框. 1 表示有边框(数字越大, 边框越粗), "" 表示没边框.
> - cellpadding: 内容距离边框的距离, 默认 1 像素
> - cellspacing: 单元格之间的距离. 默认为 2 像素
> - width / height: 设置尺寸.
>
> 注意, 这几个属性, vscode 都提示不出来.

```
<table align="center" border="1" cellpadding="20" cellspacing="0" width="500"
height="500">
    <tr>
        <td>姓名</td>
        <td>性别</td>
        <td>年龄</td>
    </tr>
    <tr>
        <td>张三</td>
        <td>男</td>
        <td>10</td>
    </tr>
    <tr>
        <td>李四</td>
        <td>女</td>
        <td>11</td>
    </tr>
</table>
```

合并单元格

- 跨行合并: `rowspan="n"`
- 跨列合并: `colspan="n"`

步骤

1. 先确定跨行还是跨列
2. 找好目标单元格(跨列合并, 左侧是目标单元格; 跨行合并, 上方是目标单元格)
3. 删除的多余的单元格

```
<table align="center" border="10" cellpadding="20" cellspacing="0" width="500"
height="500">
    <tr>
```

```
        <td>姓名</td>
        <td>性别</td>
        <td>年龄</td>
    </tr>
    <tr>
        <td>张三</td>
        <td colspan="2">男</td>
    </tr>
    <tr>
        <td>李四</td>
        <td>女</td>
        <td>11</td>
    </tr>
</table>
```

列表标签

主要使用来布局的. 整齐好看.

- 无序列表[重要] `ul` `li`, .
- 有序列表[用的不多] `ol` `li`
- 自定义列表[重要] `dl`(总标签) `dt`(小标题) `dd`(围绕标题来说明) 上面有个小标题, 下面有几个围绕着标题来展开的.

自定义列表(参考小米官网下方)

> **注意**
>
> - 元素之间是并列关系
> - ul/ol 中只能放 li 不能放其他标签, dl 中只能放 dt 和 dd
> - li 中可以放其他标签.
> - 列表带有自己的样式, 可以使用 CSS 来修改. (例如前面的小圆点都会去掉)

```
<h3>无序列表</h3>
<ul>
    <li>咬人猫</li>
    <li>兔总裁</li>
    <li>阿叶君</li>
</ul>

<h3>有序列表</h3>
<ol>
    <li>咬人猫</li>
    <li>兔总裁</li>
    <li>阿叶君</li>
</ol>

<h3>自定义列表</h3>
<dl>
    <dt>我的老婆们</dt>
    <dd>咬人猫</dd>
    <dd>兔总裁</dd>
    <dd>阿叶君</dd>
</dl>
```

无序列表

- 咬人猫
- 兔总裁
- 阿叶君

有序列表

1. 咬人猫
2. 兔总裁
3. 阿叶君

自定义列表

我的老婆们
    咬人猫
    兔总裁
    阿叶君

表单标签

表单是让用户输入信息的重要途径.

分成两个部分:

- 表单域: 包含表单元素的区域. 重点是 form 标签.
- 表单控件: 输入框, 提交按钮等. 重点是 input 标签.

form 标签

```
<form action="test.html">
    ... [form 的内容]
</form>
```

描述了要把数据按照什么方式, 提交到哪个页面中.

> 关于 form 需要结合 服务器 & 网络编程 来进一步理解. 后面再详细研究.

input 标签

各种输入控件, 单行文本框, 按钮, 单选框, 复选框.

- type(必须有), 取值种类很多多, button, checkbox, text, file, image, password, radio 等.
- name: 给 input 起了个名字. 尤其是对于 单选按钮, 具有相同的 name 才能多选一.
- value: input 中的默认值.

- checked: 默认被选中. (用于单选按钮和多选按钮)
- maxlength: 设定最大长度.

1) 文本框

```
<input type="text">
```

2) 密码框

```
<input type="password">
```

3) 单选框

```
性别:
<input type="radio" name="sex">男
<input type="radio" name="sex" checked="checked">女
```

> 注意: 单选框之间必须具备相同的 name 属性, 才能实现 **多选一** 效果.

4) 复选框

```
爱好:
<input type="checkbox"> 吃饭 <input type="checkbox"> 睡觉 <input type="checkbox">
打游戏
```

5) 普通按钮

```
<input type="button" value="我是个按钮">
```

> 当前点击了没有反应. 需要搭配 JS 使用(后面会重点研究).
>
> ```
> <input type="button" value="我是个按钮" onclick="alert('hello')">
> ```

6) 提交按钮

```
<form action="test.html">
    <input type="text" name="username">
    <input type="submit" value="提交">
</form>
```

> 提交按钮必须放到 form 标签内. 点击后就会尝试给服务器发送

'demo_表单标签/test.html?username=张三

7) 清空按钮

```
<form action="test.html">
    <input type="text" name="username">
    <input type="submit" value="提交">
    <input type="reset" value="清空">
</form>
```

> 清空按钮必须放在 form 中. 点击后会将 form 内所有的用户输入内容重置.

8) 选择文件

```
<input type="file">
```

> 点击选择文件, 会弹出对话框, 选择文件.

选择文件 未选择任何文件

label 标签

搭配 input 使用. 点击 label 也能选中对应的单选/复选框, 能够提升用户体验.

- for 属性: 指定当前 label 和哪个相同 id 的 input 标签对应. (此时点击才是有用的)

```
<label for="male">男</label> <input id="male" type="radio" name="sex">
```

select 标签

下拉菜单

- option 中定义 selected="selected" 表示默认选中.

```
<select>
    <option>北京</option>
    <option selected="selected">上海</option>
</select>
```

注意! 可以给的第一个选项, 作为默认选项

```
<select>
    <option>--请选择年份--</option>
    <option>1991</option>
    <option>1992</option>
    <option>1993</option>
    <option>1994</option>
    <option>1995</option>
</select>
```

textarea 标签

```
<textarea rows="3" cols="50">

</textarea>
```

> 文本域中的内容, 就是默认内容, 注意, 空格也会有影响.

> rows 和 cols 也都不会直接使用, 都是用 css 来改的.

无语义标签: div & span

div 标签, division 的缩写, 含义是 分割

span 标签, 含义是跨度

就是两个盒子. 用于**网页布局**

- div 是独占一行的, 是一个大盒子.
- span 不独占一行, 是一个小盒子.

```
<div>
    <span>咬人猫</span>
    <span>咬人猫</span>
    <span>咬人猫</span>
</div>
<div>
    <span>兔总裁</span>
    <span>兔总裁</span>
    <span>兔总裁</span>
</div>
<div>
    <span>阿叶君</span>
    <span>阿叶君</span>
    <span>阿叶君</span>
</div>
```

综合案例: 展示简历信息

某某某

基本信息

求职意向：Java 开发工程师

联系电话：XXX-XXX-XXXX

邮箱：xxx@foxmail.com

我的 github

我的 博客

教育背景

1. 1990 - 1996 小葵花幼儿园 幼儿园
2. 1996 - 2002 小葵花小学 小学
3. 2002 - 2005 小葵花中学 初中
4. 2005 - 2008 小葵花中学 高中
5. 2008 - 2012 小葵花大学 计算机专业 本科

专业技能

- Java 基础语法扎实，已经刷了 800 道 Leetcode 题；
- 常见数据结构都可以独立实现并熟练应用；
- 熟知计算机网络理论，并且可以独立排查常见问题；
- 掌握 Web 开发能力，并且独立开发了学校的留言墙功能。

我的项目

1. **留言墙**

   开发时间：2008年9月 到 2008年12月

   功能介绍：

功能介绍：

- 支持留言发布
- 支持匿名留言

2. **学习小助手**

开发时间：2008年9月 到 2008年12月

功能介绍：

- 支持错题检索
- 支持同学探讨

个人评价

在校期间，学习成绩优良，多次获得奖学金。

```
<h1>某某某</h1>
<!-- 基本信息 -->
<div>
    <h2>基本信息</h2>
    <img src="image/profile.jpg">
    <p><span>求职意向：</span>Java 开发工程师</p>
    <p><span>联系电话：</span>XXX-XXX-XXXX</p>
    <p><span>邮箱：</span>xxx@foxmail.com</p>
    <p><a href="https://github.com">我的 github</a></p>
    <p><a href="https://csdn.com">我的 博客</a></p>
</div>

<!-- 教育背景 -->
<div>
    <h2>教育背景</h2>
    <ol>
        <li>1990 - 1996 小葵花幼儿园 幼儿园</li>
        <li>1996 - 2002 小葵花小学 小学</li>
        <li>2002 - 2005 小葵花中学 初中</li>
        <li>2005 - 2008 小葵花中学 高中</li>
        <li>2008 - 2012 小葵花大学 计算机专业 本科</li>
    </ol>
</div>

<!-- 专业技能 -->
<div>
    <h2>专业技能</h2>
    <ul>
        <li>Java 基础语法扎实，已经刷了 800 道 Leetcode 题；</li>
        <li>常见数据结构都可以独立实现并熟练应用；</li>
        <li>熟知计算机网络理论，并且可以独立排查常见问题；</li>
        <li>掌握 Web 开发能力，并且独立开发了学校的留言墙功能。</li>
    </ul>
</div>

<!-- 项目 -->
<div>
    <h2>我的项目</h2>
```

```
    <ol>
        <li>
            <h3>留言墙</h3>
            <p>开发时间：2008年9月 到 2008年12月</p>
            <p>功能介绍：
            <ul>
                <li>支持留言发布</li>
                <li>支持匿名留言</li>
            </ul>
        </p>
    </li>
<li>
    <h3>学习小助手</h3>
    <p>开发时间：2008年9月 到 2008年12月</p>
    <p>功能介绍：
    <ul>
        <li>支持错题检索</li>
        <li>支持同学探讨</li>
    </ul>
</p>
</li>
</ol>
</div>

<!-- 其他信息 -->
<div>
    <h2>个人评价</h2>
    <p>在校期间，学习成绩优良，多次获得奖学金。</p>
</div>
```

综合案例: 填写简历信息

请填写简历信息

姓名
性别 ♂男 ♀女
出生日期 --请选择年份-- --请选择月份-- --请选择日期--
就读学校
应聘岗位 前端开发 后端开发 测试开发 运维开发
掌握的技能
项目经历
我已仔细阅读过公司的招聘要求
查看我的状态

请应聘者确认:

- 以上信息真实有效
- 能够尽早去公司实习
- 能接受公司的加班文化

> 提示:
>
> - 使用表格进行整体布局
> - 使用各种 input 标签 + textarea 实现页面中的输入控件
> - input 标签搭配合适的 label 提升用户体验.
> - 针对下拉框这种选项较多的, 使用 Emmet 快捷键提高输入效率.
> - 图标图片可以去 https://www.iconfont.cn/ 找

```
<table width="500px" cellspacing="0">
    <thead>
        <h3>请填写简历信息</h3>
    </thead>
    <tbody>
        <tr>
            <td>
                <label for="name">姓名</label>
            </td>
            <td>
                <input type="text" id="name">
            </td>
        </tr>
```

```
        <tr>
            <td>
                性别
            </td>
            <td>
                <input type="radio" name="sex" id="male" checked="checked">
                <!-- img 要放到 label 内部, 保证点击图标也能选中单选框 -->
                <!-- 还需要把 width 加上, 否则图片太大了 -->
                <label for="male"><img src="./image/男.png" alt="" width="20px">
男</label>
                <input type="radio" name="sex" id="female">
                <label for="female"><img src="./image/女.png" alt=""
width="20px">女</label>
            </td>
        </tr>
        <tr>
            <td>
                出生日期
            </td>
            <td>
                <select>
                    <option>--请选择年份--</option>
                    <option>1998</option>
                    <option>1999</option>
                    <option>2000</option>
                    <option>2001</option>
                </select>
                <select>
                    <option>--请选择月份--</option>
                    <option value="">1</option>
                    <option value="">2</option>
                    <option value="">3</option>
                    <option value="">4</option>
                    <option value="">5</option>
                    <option value="">6</option>
                    <option value="">7</option>
                    <option value="">8</option>
                    <option value="">9</option>
                    <option value="">10</option>
                    <option value="">11</option>
                    <option value="">12</option>
                </select>
                <select>
                    <option>--请选择日期--</option>
                    <option value="">1</option>
                    <option value="">2</option>
                    <option value="">3</option>
                    <option value="">4</option>
                    <option value="">5</option>
                    <option value="">6</option>
                    <option value="">7</option>
                    <option value="">8</option>
                    <option value="">9</option>
                    <option value="">10</option>
                    <option value="">11</option>
                    <option value="">12</option>
                    <option value="">13</option>
                    <option value="">14</option>
```

```
                <option value="">15</option>
                <option value="">16</option>
                <option value="">17</option>
                <option value="">18</option>
                <option value="">19</option>
                <option value="">20</option>
                <option value="">21</option>
                <option value="">22</option>
                <option value="">23</option>
                <option value="">24</option>
                <option value="">25</option>
                <option value="">26</option>
                <option value="">27</option>
                <option value="">28</option>
                <option value="">29</option>
                <option value="">30</option>
                <option value="">31</option>
            </select>
        </td>
    </tr>
    <tr>
        <td>
            就读学校
        </td>
        <td>
            <input type="text">
        </td>
    </tr>
    <tr>
        <td>
            应聘岗位
        </td>
        <td>
            <input type="checkbox" id="frontend">
            <label for="frontend">前端开发</label>
            <input type="checkbox" id="backend">
            <label for="backend">后端开发</label>
            <input type="checkbox" id="qa">
            <label for="qa">测试开发</label>
            <input type="checkbox" id="op">
            <label for="op">运维开发</label>
        </td>
    </tr>
    <tr>
        <td>
            掌握的技能
        </td>
        <td>
            <textarea name="" id="" cols="30" rows="10"></textarea>
        </td>
    </tr>
    <tr>
        <td>
            项目经历
        </td>
        <td>
            <textarea name="" id="" cols="30" rows="10"></textarea>
        </td>
```

```
        </tr>
        <tr>
            <td></td>
            <td>
                <input type="checkbox" id="lisence">
                <label for="lisence">我已仔细阅读过公司的招聘要求</label>
            </td>
        </tr>
        <tr>
            <td></td>
            <td>
                <a href="#">查看我的状态</a>
            </td>
        </tr>
        <tr>
            <td></td>
            <td>
                <h3>请应聘者确认：</h3>
                <ul>
                    <li>以上信息真实有效</li>
                    <li>能够尽早去公司实习</li>
                    <li>能接受公司的加班文化</li>
                </ul>
            </td>
        </tr>
    </tbody>
</table>
```

Emmet 快捷键

- 快速输入标签

```
input[tab]
```

- 快速输入多个标签

```
div*3[tab]
```

- 标签带id

```
div#sex[tab]
```

- 标签带类名

```
div.sex[tab]
```

- 标签带子元素

```
ul>li*3[tab]
```

- 标签带兄弟元素

```
span+span
```

- 标签带内容

```
div{hello}
```

- 标签带内容(带编号)

```
div{$.hello}
```

> 除此之外还有很多, 大家可以在使用中自己积累.

HTML 特殊字符

有些特殊的字符在 html 文件中是不能直接表示的, 例如:

空格: ` `

小于号: `<`

大于号: `>`

按位与: `&`

> html 标签就是用 < > 表示的. 因此内容里如果存在 < > , 就会发生混淆.

参考内容:

https://www.jb51.net/onlineread/htmlchar.htm

参考文档

MDN HTML 介绍

MDN HTML 文档

小结

HTML 只是描述了页面的骨架结构.

使用 CSS 可以针对页面进行进一步美化.