

# 目录

1. Html.....	1
1.1 基本结构介绍.....	1
1.2 CSS 介绍.....	1
1.3 H5 页面布局特点.....	1
2. 头文件.....	1
3. 标签.....	2
4. 快捷键.....	3
1) Visual studio code 快捷键.....	3
2) Visual studio code 瞬间敲出一片代码设置.....	4
5. 样式.....	6
5.1 关于字体的样式.....	6
5.2 关于间距的样式.....	7
5.3 选择器.....	7
5.4 关于边框线.....	8
5.5 关于 position.....	8
5.6 关于居中.....	9
5.7 样式继承.....	11
5.8 其他.....	11
6. 实体符号（&开头的就是实体符号）.....	12
7. 文字效果.....	12
7.1 加粗效果、粗体：.....	12
7.2 字体倾斜.....	12
7.3 下划线.....	12
7.4 删除线.....	12
8. 关于表格.....	13
9. 关于列表.....	13
1) 有序列表.....	14
2) 无序列表.....	14
3) 定义列表/详细列表.....	14
10. 前缀后缀.....	14
11. 自动编号.....	15
12. 行元素 与 块元素.....	15
13. 并排.....	17
14. 单位.....	17
15. 浮动.....	17
16. 关于鼠标效果.....	18
17. 定宽居中.....	18
18. 浮动分栏.....	19
19. 内容分栏.....	19
20. 图片横幅.....	19
21. box 垂直居中.....	20
22. flex 弹性布局.....	20

23. 空白与换行.....	22
24. 2D 变换 3D 变换.....	22
25. 动画.....	25
26. CSS 过渡.....	26
27. Viewport.....	27
28. 表单.....	29
1) 单选框.....	29
2) 多选框.....	30
3) 下拉选择菜单.....	30
4) 多行文本框.....	30
5) 文本框.....	30
6) 密码框.....	31
7) 按钮.....	31
8) 媒体查询技术.....	31
29. Less.....	31
1) 导入头文件.....	31
2) 样式.....	31
3) 基本格式.....	32
4) 基本 DOS 命令.....	32
5) 颜色操作函数.....	32
30. 关于 PS.....	32
31. Bootstrap.....	33

# 1. Html

超文本标记语言，使用各种标签描述内容

## 1.1 基本结构介绍

```
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
  </body>
</html>
```

## 1.2 CSS 介绍

Cascading Style Sheet 层叠样式表，一组可以选择性应用的页面布局和美化的样式列表

```
Article.alert-text{
  Color:red;———>颜色      }
  Font-size:13px;——>字号      }样式列表
  Font-family:'楷体' ;——>字体  }
  _____
  样式名      样式值
}
```

1.3 H5 页面布局特点

（流式布局）

方向上：从左往右，从上往下  
空间上：尽可能少的占用空间，但块元素会在横向上占满

2. 头文件

```
<meta charset="utf-8">  
// (8bit Unicode Transformation Format 统一码转换格式)，使中文不会乱码
```

UTF-8 和 GBK 的区别	
UTF-8	GBK
英文用 8 位表示，中文用 16 位表示	全部用 16 位表示
可以表示世界上所有的语言	只能存储英文和中文
前 128 个 ASCII 码相同	
UTF-8 是 Unicode 语言	GBK 不是 Unicode 语言

3. 标签

```
(1)<a href="http://www.baidu.com" target="_blank">百度搜索</a>  
//<标签名 属性名="属性值" >内容</结束标签>== <开始标签></结束标签>
```

href 内是链接地址，  
href="#"：跳转到页面顶部  
href="id 名"：跳转到 id 所在位置  
target="\_blank"：在新的窗口打开链接(target 目标、blank 空白)

```
(2)
```

//id: identity 身份  
Src: source 图片源地址  
Title: 鼠标放在图片上会显示这句话  
Alt: alternative 可供替代的、两者挑一的，如果图片不显示，就会显示 alt 的内容

(3)

<span></span>	<div></div>
span 跨度、范围	Div 分区，可以让他内部的内容独占一行

(4)<cite>西游记</cite>

//定义引用，对参考文献的引用进行定义，比如书籍或杂志的标题。

(5)<figure><figcaption></figcaption></figure>

//figure 标签用于对元素进行组合，figcaption 标签用于为元素组添加标题

```
<figure>
```

```
  
```

```
  <figcaption lang="zh-CN">著名韩国演员</figcaption>
```

```
  //lang 语言，zh-CN 中文简体，TW 中文繁体，jp 日语，en 英语
```

```
  <figcaption lang="en">Famous Korean Actor Li Zhong Shuo</figcaption>
```

```
</figure>
```

(6)<abbr></abbr>//缩写

(7)<var></var>//变量

(8)<pre></pre>//使用 pre 标签可以保留其中内容的空白和换行

(9)<code></code>//里边包裹的是代码

(10)<q></q>//quote 引号，标签内内容自动被“”起来

```
Q{
```

```
  Quote:'『』''「」';//引号可自定义
```

```
}
```

(11)<aside></aside>//定义页面内容之外的内容

(12)<br>//换行

(13)<hr>//横线

(14)<ul></ul>//无序列表

(15)<ol></ol>//有序列表

(16)<li></li>//列表项

(17)<p></p>//段落

(18)<nav></nav>//菜单

(19)<h1></h1>//标题

(20)<h7></h7>//

(21)<section></section>//部分

(22)<a></a>//链接

(23)<optgroup></optgroup>//定义选项组，用于组合选项

(24)<fieldset></fieldset>//出现一个框把里边的内容包裹起来，可将表单内的相关元素分组，当一组表单元素放到 <fieldset> 标签内时，浏览器会以特殊方式来显示它们，它们可能有特殊的边界、3D 效果，或者甚至可创建一个子表单来处理这些元素。

(25)<legend></legend>//legend 元素为 fieldset 元素定义标题，字出现在大框线上

(26)<button></button>//按钮

(27)<dl></dl>//定义列表

(28)<dt></dt>//定义了定义列表中的项目（即术语部分）

(29)<dd></dd>//在定义列表中定义条目的定义部分

## 4. 快捷键

**Tab 键**: 对齐快捷键, 增加缩进

**Shift+tab**: 减少缩进

**Ctrl+[**: 减少缩进

**Ctrl+]**: 增加缩进

**Shift+Tab**: 对齐格式往前

**Ctrl+shift+m**: 火狐浏览器响应式设计视图切换

### 1) Visual studio code 快捷键

1. 英文输入模式下 Ctrl+k, m 选择 html 语言
2. 瞬间敲完一片代码, 代码预置:  
F1---->sni----->html---->在 body 里写默认格式代码, 里边有 “ ” 的前边加\即可正常显示; \$1 是默认光标位置, \t 是缩进, 换行\t\t\t 是空出一行, \$2 是 enter 后光标移到此处  
新建--->ctrl+K--->M--->html--->Go--->Enter
3. 对齐大招: ctrl+a 全选, shift+alt+f 对齐
4. 新建: ctrl+shift+n 在新窗口新建  
Ctrl+n 在已打开窗口新建
5. 替换和选区替换: 选中--->ctrl+h

### 2) Visual studio code 瞬间敲出一片代码设置

```
{
/*
    // Place your snippets for HTML here. Each snippet is defined under a snippet name and has
    a prefix, body and
    // description. The prefix is what is used to trigger the snippet and the body will be
    expanded and inserted. Possible variables are:
    // $1, $2 for tab stops, ${id} and ${id:label} and ${1:label} for variables. Variables with the
    same id are connected.
    // Example:
    "Print to console": {
        "prefix": "log",
        "body": [
            "console.log('$1');",
            "$2"
        ],
        "description": "Log output to console"
    }
}
```

\*/

"HTML5 文档基本标签": {

  "prefix": "go",

  "body": [

    "<!DOCTYPE html>",

    "<html>",

    "\t<head>",

    "\t\t<meta charset=\"utf-8\">",

    "\t\t<meta name=\"author\" content=\"夏梦君\">",

    "\t\t<link rel=\"shortcut icon\" href=\"favicon.png\" type=\"image/png\">",

    "\t\t<link rel=\"icon\" href=\"favicon.png\" type=\"image/png\">",

    "\t\t<title>\$1</title>",

    "\t\t<style>",

    "\t\t\t",

    "\t\t</style>",

    "</head>",

    "<body>",

    "\t\t\$2",

    "</body>",

    "</html>"

  ],

  "description": "瞬间敲完一片代码"

},

"HTML5 App 文档基本标签": {

  "prefix": "app",

  "body": [

    "<!DOCTYPE html>",

    "<html>",

    "\t<head>",

    "\t\t<meta name=\"viewport\" content=\"initial-scale=1, maximum-scale=1, user-scalable=no\">",

    "\t\t<meta charset=\"utf-8\">",

    "\t\t<meta name=\"author\" content=\"夏梦君\">",

    "\t\t<link rel=\"shortcut icon\" href=\"favicon.png\" type=\"image/png\">",

    "\t\t<link rel=\"icon\" href=\"favicon.png\" type=\"image/png\">",

    "\t\t<title>\$1</title>",

    "\t\t<style>",

    "\t\t\t",

    "\t\t</style>",

    "</head>",

    "<body>",

    "\t\t\$2",

```

        "\t</body>",
        "</html>"
    ],
    "description": "好好学习"
},

"Bootstrap 文档基本结构": {
    "prefix": "boot",
    "body": [
        "<!DOCTYPE html>",
        "<html lang=\"zh-CN\">",
        "\t<head>",
        "\t\t<meta charset=\"utf-8\">",
        "\t\t<meta http-equiv=\"X-UA-Compatible\" content=\"IE=edge\">",
        "\t\t<meta name=\"viewport\" content=\"width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no\">",
        "\t\t<meta name=\"author\" content=\"夏梦君\">",
        "\t\t<link rel=\"shortcut icon\" href=\"favicon.png\" type=\"image/png\">",
        "\t\t<link rel=\"icon\" href=\"favicon.png\" type=\"image/png\">",
        "\t\t<title>$1</title>",
        "\t\t<link href=\"css/bootstrap.min.css\" rel=\"stylesheet\">",
        "\t\t<style>",
        "\t\t\t",
        "\t\t</style>",
        "\t</head>",
        "\t<body>",
        "\t\t$2",
        "\t\t<script src=\"js/jquery.min.js\"></script>",
        "\t\t<script src=\"js/bootstrap.min.js\"></script>",
        "\t</body>",
        "</html>"
    ],
    "description": "Bootstrap 文档基本结构"
}
}

```

## 5. 样式

**<header class="fs40px white absolute">**//样式组合，使用多个 class，用空格隔开，上边样式表写很多种



## 5.1 关于字体的样式

Font-weight:400;//调整字体的粗细，400 正常（范围：100-700）  
Text-decoration:none;//取消文字下划线  
Text-decoration:underline;//给文字添加下划线  
Font-family:'宋体','隶书';//改变字体，首选第一个字体，没有的话往后选  
Color:red;//调整文字颜色  
Font-style:normal;//文字风格  
font-variant: small-caps;//font-variant 字体变形，small-caps: 小型大写字母(capital 大写字母)  
text-indent: 2em;// text-indent 文本缩，em 是相对单位，相对于当前文字的大小  
text-shadow: 5px 5px 10px red;//文本阴影：上下 左右 大小 颜色  
text-transform: capitalize;//文本转换：uppercase 全部大写，lowercase 全部小写，capitalize 只把首字母大写

## 5.2 关于间距的样式

外边距 margin 会重叠，重叠后取最大的外边距为外边距  
内边距 padding 不会重叠  
Line-height:40px;//调整行高  
Padding: 40px 20px;//设置内间距，可设置 1 个值，2 个值（上下、左右），4 个值（上、右、下、左）  
Margin:20px 40px;//设置外边距，可设置 1 个值，2 个值（上下、左右），4 个值（上、右、下、左）

## 5.3 选择器

*	作用范围最广的样式选择器，对所有标签都起效
标签名称	所有同名的标签
#id	对指定的 id 标签
.class	对含有此 class 的标签
A b	嵌套关系
A>b	父子关系
:nth-child( )	第 X 个标签，所有标签都计数
:nth-of-type( )	第 X 个同类标签
:first-child	第 1 个标签

:last-child	最后 1 个标签
:link	选择所有未被访问的链接。
:visited	选择所有已被访问的链接。
:active	选择活动链接。（鼠标点住的效果）
:hover	选择鼠标指针位于其上的链接。
:focus	选择获得焦点的 input 元素。
header>*	Header 内所有标签（如 header>*:nth-of-type(2n+1)）
section:not(:nth-of-type(n+5)) :first-letter{ Font-size: 2em; }	----section:not(:nth-of-type(n+5))把第 5 个以及往后的排除掉 ----first-letter 第一个字母/字(执行后首字变大) ----first-line 第一行
: empty	选择每个空标签 :empty:before{ content: '额外加入的文字，相加什么加什么'; content: url(logo.png);//在下方空的 span 标签 内插入了这张 图片}
:only-child	某个标签只有一个子标签就会被选中
:only-of-type	某个标签里这个类型的标签只有一个，就会被选中
属性选择器	
a[ ]表示属性	----a[title] a[title=购物、分享] ----a[title^=购物]//只要 a 标签里 title 以购物开头，以下样式就会起作用 ----a[title\$=新闻]//只要 a 标签里 title 以新闻结尾，以下样式就会起作用 ----a[zhiyou100*='9']//[属性名*=属性内含有的任意字] 数字要加 '';汉字不用

#### 5.4 关于边框线

Border:2px solid red;//设置边框线：粗细 线型 颜色

Border-style:none;//设置边框样式

none	定义无边框。
hidden	与 "none" 相同。不过应用于表时除外，对于表，hidden 用于解决边框冲突。
dotted	定义点状边框。在大多数浏览器中呈现为实线。

dashed	定义虚线。在大多数浏览器中呈现为实线。
solid	定义实线。
double	定义双线。双线的宽度等于 border-width 的值。
groove	定义 3D 凹槽边框。其效果取决于 border-color 的值。
ridge	定义 3D 垄状边框。其效果取决于 border-color 的值。
inset	定义 3D inset 边框。其效果取决于 border-color 的值。
outset	定义 3D outset 边框。其效果取决于 border-color 的值。
inherit	规定应该从父元素继承边框样式。

Border-radius:50%;//宽高相等时变成圆形，可设置 px 值，调整圆角大小  
Outline:1px dotted red;//轮廓线：粗细 线型 颜色  
Outline-offset:100px;//设置轮廓线与内容的距离  
Box-shadow:0px 0px 10px red;//边框阴影：上下 左右 大小 颜色

## 5.5 关于 position

Position: fixed;  
Top/bottom/left/right:0;//固定位置，必须这两句话一块用，也可以设置具体值，会有空隙

position:absolute;//绝对定位，相对于所在区域上下左右移动  
Position:relative;//相对定位

Relative 的 2 个作用：a.相对于原先位置偏移，通常用于对标签位置进行微调  
B.作为内部标签，absolute 定位参照物

## 5.6 关于居中

Margin:0 auto;  
Text-align:center;//只对文字有效  
Vertical-align:center;//垂直居中，只对自身有效，对内部子元素无效

(1)\*{ /\*表示所有标签，此处声明后，全文生效  
Box-sizing:border-box;//设置的 width 包含内容及内边距与边框宽度  
Box-sizing:content-box;//设置的 width 只包含内容的宽度（默认）  
}

(2)背景设置  
Background-color:red;//设置背景颜色  
background-color: rgb(red, green, blue);//0 是最暗的，F 是最亮的 三位数：红 绿 蓝

Background-repeat 定义了图像的平铺模式

repeat	默认。背景图像将在垂直方向和水平方向重复。
--------	-----------------------

repeat-x	背景图像将在水平方向重复。
repeat-y	背景图像将在垂直方向重复。
no-repeat	背景图像将仅显示一次。
inherit	规定应该从父元素继承 background-repeat 属性的设置。

**Background-position** 属性设置背景图片的起始位置

top left top center top right center left center center center right bottom left bottom center bottom right	如果您仅规定了一个关键词，那么第二个值将是"center"。  默认值：0% 0%。
x% y%	第一个值是水平位置，第二个值是垂直位置。 左上角是 0% 0%。右下角是 100% 100%。 如果您仅规定了一个值，另一个值将是 50%。
xpos ypos	第一个值是水平位置，第二个值是垂直位置。 左上角是 0 0。单位是像素 (0px 0px) 或任何其他 CSS 单位。 如果您仅规定了一个值，另一个值将是 50%。 您可以混合使用 % 和 position 值。

**background-origin**: content-box;

//background-origin: content-box;背景图片从内容开始;

background-origin: border-box;背景图片从外边框开始

background-origin: padding-box;背景图片从内边框开始

**background-clip**: content-box;clip: //对背景图片和背景颜色都有影响

//background-clip: border-box;背景被裁剪到边框盒

background-clip: Padding-box;背景被裁剪到内边距框

background-clip: content-box;背景被裁剪到内容框

**background-size**: 50%;//只影响背景图片，不影响背景颜色

//background-size: length|percentage|cover|contain;

**Length**: 设置背景图像的高度和宽度。第 1 个值设置宽度，第 2 个值设置高度。如果只设置 1 个值，则第 2 个值会被设置为 "auto"。

**Percentage**: 以父元素的百分比来设置背景图像的宽度和高度。第 1 个值设置宽度，第 2 个

值设置高度。如果只设置 1 个值，则第 2 个值会被设置为 "auto"。

**Cover:** 把背景图像扩展至足够大，以使背景图像完全覆盖背景区域。背景图像的某些部分也许无法显示在背景定位区域中。

**Contain:** 把图像扩展至最大尺寸，以使其宽度和高度完全适应内容区域。

**background-image:** url(01.jpg),url(ouba.jpg);//当设置超过一张背景图片时可使用下边的背景图片混合模式

```
background-blend-mode: screen;//背景图片混合模式
background-blend-mode: normal;    //正常
background-blend-mode: multiply;   //正片叠底
background-blend-mode: screen;    //滤色
background-blend-mode: overlay;   //叠加
background-blend-mode: darken;    //变暗
background-blend-mode: lighten;   //变亮
background-blend-mode: color-dodge; //颜色减淡
background-blend-mode: color-burn; //颜色加深
background-blend-mode: hard-light; //强光
background-blend-mode: soft-light; //柔光
background-blend-mode: difference; //差值
background-blend-mode: exclusion; //排除
background-blend-mode: hue;       //色相
background-blend-mode: saturation; //饱和度
background-blend-mode: color;     //颜色
background-blend-mode: luminosity; //亮度

background-blend-mode: initial;    //初始
background-blend-mode: inherit;    //继承
background-blend-mode: unset;     //复原
```

## 5.7 样式继承

大部分样式都不能继承，如 width、height、background-color 等，

只有少数样式可以继承，会继承的大部分都是跟文字有关的，比如 color、font-size、font-weight、text-align、font-weight、font-family、text-decoration、font-style、opacity、line-height

## 5.8 其他

`Z-index:100;`//设置层的位置，数越大位置越高，越不容易被覆盖

`Overflow:scroll;`//加滚动条，英文单词和图片默认是不能换行的

`Overflow:hidden;`//直接切掉多出来的内容，不用滚动条

`Resize:vertical;`//在高度上可以自己拖动调整

`Resize:both;`//在宽度和高度上都可以自己拖动调整

`Resize:horizontal;`//在水平方向上可以自己拖动调整

`Min-width:300px;`//设置宽度的最小值

`Max-width:500px;`//设置宽度的最大值

`Width:400px;`//width 优先级大于 min-width 和 max-width，有他在直接定宽

**裁剪标签：**

**clip:** rect(30px,180px,135px,0);//绝对定位才能使用 clip 裁剪（rectangle：长方形，矩形）

//clip: rect(30px: 上方要切掉的,180px 右边要保留的,135px 下边要保留的,0 左边要切掉的);

## 6. 实体符号（&开头的就是实体符号）

`&lt;==<`

`&gt;==>`

`&copy;==`圈 C(版权所有用的)

`&nbsp;==`英文空格//在英文中，手敲的空格可以换行，`&nbsp;space`不能折断的空白)不能换行！

`&#x3000;==`中文空格

`&hearts;`红桃

`&clubs;`梅花

`&spades;`黑桃

`&diams;`方片

## 7. 文字效果

### 7.1 加粗效果、粗体:

```
Font-weight:bold;  
Font-weight:400px;(400 以上加粗, 范围 400-700)  
<strong></strong>//强调  
<b></b>
```

### 7.2 字体倾斜

```
<em></em>//emphasis 强调  
<i></i>  
Font-style:italic;
```

### 7.3 下划线

```
Text-decoration:underline;  
<U></u>//underline 的缩写  
<ins></ins>//insert 插入
```

### 7.4 删除线

```
<del></del>//delete 删除  
<s></s>  
Text-decoration:line-through;
```

## 8. 关于表格

```
Border-collapse:collapse;//表格边框线合并  
<caption>标题</caption>//表格标题  
Caption-side 属性设置表格标题的位置  
    //top: 默认值。把表格标题定位在表格之上。  
    //Bottom: 把表格标题定位在表格之下。  
    //Inherit: 规定应该从父元素继承 caption-side 属性的值。  
<thead></thead>//表头  
<tr></tr>//行  
<th></th>//目录行, 自定义加粗显示
```

<td></td>//单元格  
 <td rowspan="2">自习</td> //rowspan: 合并上下两行  
 <td colspan="2">休息</td> //colspan: 合并左右两列

## 9. 关于列表

**List-style-type:none;//设置列表项标记的类型**

none	无标记。
disc	默认。标记是实心圆。
circle	标记是空心圆。
square	标记是实心方块。
decimal	标记是数字。
decimal-leading-zero	0 开头的数字标记。(01, 02, 03, 等。)
lower-roman	小写罗马数字(i, ii, iii, iv, v, 等。)
upper-roman	大写罗马数字(I, II, III, IV, V, 等。)
lower-alpha	小写英文字母 The marker is lower-alpha (a, b, c, d, e, 等。)
upper-alpha	大写英文字母 The marker is upper-alpha (A, B, C, D, E, 等。)
lower-greek	小写希腊字母(alpha, beta, gamma, 等。)
lower-latin	小写拉丁字母(a, b, c, d, e, 等。)
upper-latin	大写拉丁字母(A, B, C, D, E, 等。)
hebrew	传统的希伯来编号方式
armenian	传统的亚美尼亚编号方式
georgian	传统的乔治亚编号方式(an, ban, gan, 等。)
cjk-ideographic	简单的表意数字
hiragana	标记是: a, i, u, e, o, ka, ki, 等。(日文片假名)
katakana	标记是: A, I, U, E, O, KA, KI, 等。(日文片假名)
hiragana-iroha	标记是: i, ro, ha, ni, ho, he, to, 等。(日文片假名)
katakana-iroha	标记是: I, RO, HA, NI, HO, HE, TO, 等。(日文片假名)

**list-style-position** 属性设置在何处放置列表项标记。

inside	列表项目标记放置在文本以内, 且环绕文本根据标记对齐。
outside	默认值。保持标记位于文本的左侧。列表项目标记放置在文本以外, 且环绕文本不根据标记对齐。
inherit	规定应该从父元素继承 list-style-position 属性的值。

**list-style-image** 属性使用图像来替换列表项的标记。

URL	图像的路径。
none	默认。无图形被显示。
inherit	规定应该从父元素继承 list-style-image 属性的值。

Inherit: 规定应该从父元素继承 list-style 属性的值



### 1) 有序列表

```
<ol>//order list
  <li></li>
</ol>
```

### 2) 无序列表

```
<ul>//unorder list
  <li></li>
</ul>
list-style: none; //去掉默认无序列表符号（黑色圆点）
```

### 3) 定义列表/详细列表

```
<dl>//definition list 定义列表，对名词解释时适合用
  <!--定义列表：带标题的列表-->
  <dt>HTML5</dt>
<!--t:title 定义列表的标题，定义了定义列表中的项目（即术语部分）-->
  <dd>HTML 的第五个版本</dd>
<!--d:detail 定义列表的详情，定义列表中定义条目的定义部分-->
  <div></div>
  <dt>CSS3</dt>
  <dd>增加了图形变换、动画、圆角边框等</dd>
  <div></div>
  <dt>JavaScript</dt>
  <dd>动态脚本语言</dd>
</dl>
```

## 10. 前缀后缀

```
Div:before{
  Content:'这是点缀：' ;
}
Div:after{
  Content:'-----这是后缀'
}
```

## 11. 自动编号

```
body{
    counter-reset: h2;
}
h2:before{
    counter-increment: h2;//increment 递增
    content: counter(h2) ': ';
}
section{
    counter-reset: myh3;
}
h3:before{
    counter-increment: myh3;
    content: counter(h2) '.' counter(myh3) '.';//二级编号
}
section>section{
    counter-reset: myh4;
}
h4:before{
    counter-increment: myh4;
    content: counter(h2) '.' counter(myh3) '.' counter(myh4) '.';//
三级编号
}
```

## 12. 行元素 与 块元素

行内元素	块元素	图片
inline	block	Inline-block
排列在其他行内元素之后，宽高等样式无效	单独占一行，可设置宽高	与其他行元素一起排列，而且可以设置宽高
a、span、i、strong	Div、section、header、h1、table、figure、figcaption	Img、td

Visibility:hidden;//元素消失，但继续占用空间

Display 样式总结

none	此元素不会被显示，并且不再占用任何空间
------	---------------------

block	此元素将显示为块级元素，此元素前后会带有换行符。
inline	默认。此元素会被显示为内联元素，元素前后没有换行符。
inline-block	行内块元素，既拥有行元素的性质，又拥有块元素的性质，既可以在一行显示，又能设置宽高
list-item	此元素会作为列表显示。
run-in	此元素会根据上下文作为块级元素或内联元素显示。
compact	CSS 中有值 <code>compact</code> ，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。
marker	CSS 中有值 <code>marker</code> ，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。
table	此元素会作为块级表格来显示（类似 <code>&lt;table&gt;</code> ），表格前后带有换行符。 普通标签堆成表格
inline-table	此元素会作为内联表格来显示（类似 <code>&lt;table&gt;</code> ），表格前后没有换行符。 行内表格
table-row-group	此元素会作为一个或多个行的分组来显示（类似 <code>&lt;tbody&gt;</code> ）。
table-header-group	此元素会作为一个或多个行的分组来显示（类似 <code>&lt;thead&gt;</code> ）。
table-footer-group	此元素会作为一个或多个行的分组来显示（类似 <code>&lt;tfoot&gt;</code> ）。
table-row	此元素会作为一个表格行显示（类似 <code>&lt;tr&gt;</code> ）。
table-column-group	此元素会作为一个或多个列的分组来显示（类似 <code>&lt;colgroup&gt;</code> ）。
table-column	此元素会作为一个单元格列显示（类似 <code>&lt;col&gt;</code> ）
table-cell	此元素会作为一个表格单元格显示（类似 <code>&lt;td&gt;</code> 和 <code>&lt;th&gt;</code> ）
table-caption	此元素会作为一个表格标题显示（类似 <code>&lt;caption&gt;</code> ）
inherit	规定应该从父元素继承 <code>display</code> 属性的值。

### vertical-align: 设置元素的垂直对齐方式

baseline	默认。元素放置在父元素的基线上。
sub	垂直对齐文本的下标。
super	垂直对齐文本的上标
top	把元素的顶端与行中最高元素的顶端对齐
text-top	把元素的顶端与父元素字体的顶端对齐
middle	把此元素放置在父元素的中部。
bottom	把元素的顶端与行中最低的元素的顶端对齐。
text-bottom	把元素的底端与父元素字体的底端对齐。
length	
%	使用 "line-height" 属性的百分比值来排列此元素。允许使用负值。
inherit	规定应该从父元素继承 <code>vertical-align</code> 属性的值。

## 13. 并排

```
section{
  float: left;
  width: 33.33333333333333%;
}
<body>
  <main>
    <section>老大</section>
    <section>老二</section>
    <section>老三</section>
  </main>
</body>
```

## 14. 单位

Width:5px;  
Width:5mm;  
Width:5cm;  
Width:5in;//inch: 英寸  
Width:5pt;//pound: 磅/点  
Width:5%;  
Width:5em;//em 是相对单位，相对于当前文字的大小  
Width:5ex;//1em=2ex, em 常用, ex 不常用, ex 是相对于英文字母 x 的大小来计算尺寸的  
font-size:0.5rem;//rem=root(根)em(根标签的相对大小), 相对于 html,body 的字体大小变化, 不受父类字体大小影响

## 15. 浮动

float: left;//一个元素设了 float, 就会缩小空间, 最小跟内容一样大小, 没有设置 float 就会占据尽可能大的空间

float 会影响紧贴着他的那个元素, 需要一个替死鬼:

(法 1) <div style="clear:left"></div>, 不能确定是往哪个方向浮动就用

```
clear:both;
(法 2)<br>
(法 3)ul:after{
    content: '1';
    color: white;
    clear: left;
}
```

## 16. 关于鼠标效果

`cursor: url(logo.png), auto;` // 鼠标移上去变成这个图片，图片尺寸必须小一点  
128px 以下才能显示出来

<b>url</b>	需使用的自定义光标的 URL。 注释：请在此列表的末端始终定义一种普通的光标，以防没有由 URL 定义的可用光标。
<b>default</b>	默认光标（通常是一个箭头）
<b>auto</b>	默认。浏览器设置的光标。
<b>crosshair</b>	光标呈现为十字线。
<b>pointer</b>	光标呈现为指示链接的指针（一只手）
<b>move</b>	此光标指示某对象可被移动。
<b>e-resize</b>	此光标指示矩形框的边缘可被向右（东）移动。
<b>ne-resize</b>	此光标指示矩形框的边缘可被向上及向右移动（北/东）。
<b>nw-resize</b>	此光标指示矩形框的边缘可被向上及向左移动（北/西）。
<b>n-resize</b>	此光标指示矩形框的边缘可被向上（北）移动。
<b>se-resize</b>	此光标指示矩形框的边缘可被向下及向右移动（南/东）。
<b>sw-resize</b>	此光标指示矩形框的边缘可被向下及向左移动（南/西）。
<b>s-resize</b>	此光标指示矩形框的边缘可被向下移动（南）。
<b>w-resize</b>	此光标指示矩形框的边缘可被向左移动（西）。
<b>text</b>	此光标指示文本。
<b>wait</b>	此光标指示程序正忙（通常是一只表或沙漏）。
<b>help</b>	此光标指示可用的帮助（通常是一个问号或一个气球）。

## 17. 定宽居中

```
width: 960px;
margin: 100px auto; // margin: 0 auto; 定宽居中的必要且充分条件
```

## 18. 浮动分栏

```
li{
    list-style: none;/*去掉默认无序列表符号（黑色圆点）*/
    float: left;
    width: 25%;//四等分
}
<ul>
    <li>&hearts;</li>
    <li>&clubs;</li>
    <li>&spades;</li>
    <li>&diams;</li>
</ul>
```

//固定位置和定宽居中有冲突，所以要用嵌套的 div，外层固定位置，内层居中

## 19. 内容分栏

```
article{ /*由于内容分栏现在没有被普遍认可,所以必须用浏览器私有样式才能看到效果*/
    column-count: 3;
    -moz-column-count: 4;/*使用火狐浏览器私有样式，分成4栏*/
    -webkit-column-count: 4;/*使用谷歌浏览器私有样式，分成4栏，
    以后在移动端都要用-webkit*/
    -moz-column-gap: 50px;/*火狐浏览器下设置分栏间隙，gap 间隙*/
    -moz-column-rule: 100px solid black;//火狐浏览器私有样式下设置分
    栏中间的线的格式，相当于 border，
    不同于 borderde 是太宽会盖住字
}
h1{
    -webkit-column-span: all;/*谷歌浏览器设置 标题跨多栏*/
}
```

## 20. 图片横幅

```
section{
    height: 600px;
    background-image: url(da.jpg);
    background-position: 50% 50%; // 不管窗口缩放，始终看到的是图
片中间位置
}
```

## 21. box 垂直居中

```
Section{
    display: -moz-box;
    -moz-box-pack: center; // 使用浏览器私有样式
Span{
    display: -moz-box;
    -moz-box-align: center;
}
<section><span></span></section>
```

## 22. flex 弹性布局

设为 flex 布局以后，子元素的 float、clear 和 vertical-align 属性将失效。

Display: flex;

Display: inline-flex; // 行内元素使用 flex 布局

容器的属性：

Flex-direction 属性决定主轴的方向（即项目的排列方向）

// row: 默认值，主轴为水平方向，起点在左端

Row-reverse: 主轴为水平方向，起点在右端

Column: 主轴为垂直方向，起点在上沿

Column-reverse: 主轴为垂直方向，起点在下沿

Flex-wrap 属性定义：如果一条轴线排不下，如何换行

// nowrap: 默认，不换行

Wrap: 换行，第一行再上方

Wrap-reverse: 换行，第一行在下方



Flex-flow 属性是 flex-direction 属性和 flex-wrap 属性简写的形式，默认值是 row nowrap。

Justify-content 属性定义了项目在主轴上的对齐方式

//flex-start: 默认，左对齐

Flex-end: 右对齐

Center: 居中

Space-between: 两端对齐，项目之间的间隔都相等

Space-around: 每个项目两侧的间隔相等，所以，项目之间的间隔比例比边框的间隔大一倍

Align-items 属性定义项目在交叉轴上的对齐方式

//flex-start: 交叉轴的起点对齐

Flex-end: 交叉轴的终点对齐

Center: 交叉轴的中点对齐

Baseline: 项目的第一行文字的基线对齐

Stretch: (默认值) 如果项目未设置高度或设为 auto，将占满整个容器的高度

Align-content 属性定义了多根轴线的对齐方式，如果项目只有一根轴线，该属性不起作用

//flex-start: 与交叉轴的起点对齐

Flex-end: 与交叉轴的终点对齐

Center: 与交叉轴的中点对齐

Space-between: 与交叉轴两端对齐，轴线之间的间隔平均分布

Space-around: 每根轴线两侧的间隔相等，所以，轴线之间的间隔比例比轴线与边框的间隔大一倍

Stretch: (默认值) 轴线占满整个交叉轴

### 项目的属性:

Order 属性定义项目的排列顺序，数值越小，排列越靠前，默认为 0。

Flex-grow 属性定义项目的放大比例，默认为 0，即如果存在剩余空间，也不放大。

Flex-shrink 属性定义了项目的缩小比例，默认为 1，即如果空间不足，该项目将缩小。

当 flex-shrink 与 flex-wrap 同时存在，优先执行 flex-wrap

Flex-basis 属性定义了分配多余空间之前，项目占据的主轴空间，浏览器根据这个属性计算主轴是否有多余空间，默认值为 auto，即项目的本来大小。

可设为跟 width 和 height 属性一样的值，则项目将占据固定空间。

Flex 属性是 flex-grow，flex-shrink 和 flex-basis 的简写，默认值是 0 1 auto，后两个属性可选。

该属性有两个快捷键: auto(1 1 auto)和 none(0 0 auto)。

Align-self 属性允许单个项目有与其他项目不一样的对齐方式，可覆盖 align-items 属性。

(换行时不起作用)

默认值为 auto，表示继承父元素的 align-items 属性，如果没有

父元素，则等同于 stretch。

Auto

lex-start: 交叉轴的起点对齐

Flex-end: 交叉轴的终点对齐

Center: 交叉轴的中点对齐

Baseline: 项目的第一行文字的基线对齐

Stretch: (默认值) 如果项目未设置高度或设为 auto，将占满整个容器的高度

## 23. 空白与换行

Word-break 属性规定自动换行的处理方法

//normal: 使用浏览器默认的换行规则。

Break-all: 允许在单词内换行。(长单词、短单词换行时都会被折断，造成误解)

Keep-all: 只能在半角空格或连字符处换行。

Word-wrap 属性允许长单词或 URL 地址换行到下一行

//normal: 只在允许的断字点换行(浏览器保持默认处理)。

Break-word: 在长单词或 URL 地址内部进行换行。(只会折断长单词，使误解尽可能少)

White-space 属性设置如何处理元素内的空白

//normal: 默认。空白会被浏览器忽略。

Pre: 空白会被浏览器保留。其行为方式类似 HTML 中的 <pre> 标签。

Nowrap: 文本不会换行，文本会在同一行上继续，直到遇到 <br> 标签为止。

Pre-wrap: 保留空白符序列，但是正常地进行换行。

Pre-line: 合并空白符序列，但是保留换行符。

Inherit: 规定应该从父元素继承 white-space 属性的值。

## 24. 2D 变换 3D 变换

**transition** 属性是一个简写属性，用于设置四个过渡属性：

transition-property 规定设置过渡效果的 CSS 属性的名称。

None: 没有属性会获得过渡效果。

All: 所有属性都将获得过渡效果。

Property: 定义应用过渡效果的 CSS 属性名称列表，列表以逗号分隔。

**transition-duration** 规定完成过渡效果需要多少秒或毫秒。

默认值是 0

**transition-timing-function** 规定速度效果的速度曲线。

**Linear**: 规定以相同速度开始至结束的过渡效果（等于 **cubic-bezier(0,0,1,1)**）。

**Ease**: 规定慢速开始，然后变快，然后慢速结束的过渡效果（**cubic-bezier(0.25,0.1,0.25,1)**）。

**Ease-in**: 规定以慢速开始的过渡效果（等于 **cubic-bezier(0.42,0,1,1)**）。

**Ease-out**: 规定以慢速结束的过渡效果（等于 **cubic-bezier(0,0,0.58,1)**）。

**Ease-in-out**: 规定以慢速开始和结束的过渡效果（等于 **cubic-bezier(0.42,0,0.58,1)**）。

**Cubic-bezier(n,n,n,n)**: 在 **cubic-bezier** 函数中定义自己的值。可能的值是 0 至 1 之间的数值。

**transition-delay** 定义过渡效果何时开始。默认值为 0。

**Transform** 属性向元素应用 2D 或 3D 转换。该属性允许我们对元素进行旋转、缩放、移动或倾斜。

值	描述
none	定义不进行转换。
matrix(n,n,n,n,n,n)	定义 2D 转换，使用六个值的矩阵。
matrix3d(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)	定义 3D 转换，使用 16 个值的 4x4 矩阵。
translate(x,y)	定义 2D 转换。
translate3d(x,y,z)	定义 3D 转换。
translateX(x)	定义转换，只是用 X 轴的值。
translateY(y)	定义转换，只是用 Y 轴的值。
translateZ(z)	定义 3D 转换，只是用 Z 轴的值。
scale(x,y)	定义 2D 缩放转换。
scale3d(x,y,z)	定义 3D 缩放转换。
scaleX(x)	通过设置 X 轴的值来定义缩放转换。
scaleY(y)	通过设置 Y 轴的值来定义缩放转换。
scaleZ(z)	通过设置 Z 轴的值来定义 3D 缩放转换。
rotate(angle)	定义 2D 旋转，在参数中规定角度。
rotate3d(x,y,z,angle)	定义 3D 旋转。0 不旋转，非 0 旋转
rotateX(angle)	定义沿着 X 轴的 3D 旋转。
rotateY(angle)	定义沿着 Y 轴的 3D 旋转。
rotateZ(angle)	定义沿着 Z 轴的 3D 旋转。

skew(x-angle,y-angle)	定义沿着 X 和 Y 轴的 2D 倾斜转换。
skewX(angle)	定义沿着 X 轴的 2D 倾斜转换。
skewY(angle)	定义沿着 Y 轴的 2D 倾斜转换。
perspective(n)	为 3D 转换元素定义透视视图。

**transform-origin** 属性允许您改变被转换元素的位置。

默认值:	50% 50% 0
值	描述
x-axis	定义视图被置于 X 轴的何处。可能的值: left center right length %
y-axis	定义视图被置于 Y 轴的何处。可能的值: top center bottom length %
z-axis	定义视图被置于 Z 轴的何处。可能的值: length

**transform-style** 属性规定如何在 3D 空间中呈现被嵌套的元素。

值	描述
flat	子元素将不保留其 3D 位置。
preserve-3d	子元素将保留其 3D 位置。

**perspective** 属性定义 3D 元素距视图的距离，以像素计。该属性允许您改变 3D 元素查看 3D 元素的视图。

number	元素距离视图的距离，以像素计。
none	默认值。与 0 相同。不设置透视。

**perspective-origin** 属性定义 3D 元素所基于的 X 轴和 Y 轴。该属性允许您改变 3D 元素的底部位置。

当为元素定义 **perspective-origin** 属性时，其子元素会获得透视效果，而不是元素本身。

注释：该属性必须与 [perspective](#) 属性一同使用，而且只影响 3D 转换元素。

默认值:	50% 50%
值	描述

x-axis	定义该视图在 x 轴上的位置。默认值：50%。 可能的值： left center right length %
y-axis	定义该视图在 y 轴上的位置。默认值：50%。 可能的值： top center bottom length %

**backface-visibility** 属性定义当元素不面向屏幕时是否可见。

如果在旋转元素不希望看到其背面时，该属性很有用。

值	描述
visible （默认值）	背面是可见的。
hidden	背面是不可见的。

## 25. 动画

**@keyframes** 规则用于创建动画。在 **@keyframes** 中规定某项 CSS 样式，就能创建由当前样式逐渐改为新样式的动画效果。

以百分比来规定改变发生的时间，或者通过关键词 "from" 和 "to"，等价于 0% 和 100%。

0% 是动画的开始时间，100% 动画的结束时间。

**@keyframes animationname {keyframes-selector {css-styles;}}**

//animation name 定义动画的名称。

//keyframes-selector 必需。动画时长的百分比。合法的值：0-100%，from（与 0% 相同）to（与 100% 相同）

//css-styles 必需。一个或多个合法的 CSS 样式属性。

**animation** 属性是一个简写属性，用于设置六个动画属性：

animation-name//规定需要绑定到选择器的 keyframe 名称

**animation-duration**//规定完成动画所花费的时间，以秒或毫秒计，默认 0。

**animation-timing-function**//规定动画的速度曲线。

**animation-delay**//规定在动画开始之前的延迟，默认 0。

**animation-iteration-count**//规定动画应该播放的次数。

**animation-direction**//规定是否应该轮流反向播放动画。

注释：请始终规定 **animation-duration** 属性，否则时长为 0，就不会播放动画了。

默认值：none 0 ease 0 1 normal

**animation-timing-function**//规定动画的速度曲线。

linear	动画从头到尾的速度是相同的。
ease	默认。动画以低速开始，然后加快，在结束前变慢。
ease-in	动画以低速开始。
ease-out	动画以低速结束。
ease-in-out	动画以低速开始和结束。
cubic-bezier( <i>n,n,n,n</i> )	在 cubic-bezier 函数中自己的值。可能的值是从 0 到 1 的数值。

**animation-iteration-count** 规定动画应该播放的次数，默认 0。

//n: 定义动画播放次数的数值

//infinite: 规定动画应该无限次播放

**animation-direction** 规定是否应该轮流反向播放动画。

//normal: 默认值，动画应该正常播放

//alternate: 动画应该轮流反向播放

**animation-play-state** 属性规定动画正在运行还是暂停。

//paused: 规定动画已暂停

//running: 规定动画正在播放

**animation-fill-mode** 属性规定动画在播放之前或之后，其动画效果是否可见。

注释：其属性值是由逗号分隔的一个或多个填充模式关键词。

//none: 不改变默认行为

//Forwards: 当动画完成后，保持最后一个属性值（在最后一个关键帧中定义）

//backwards: 在 **animation-delay** 所指定的一段时间内，在动画显示之前，应用开始属性值（在第一个关键帧中定义）。

//both: 向前和向后填充模式都被应用。

## CSS 动画库

`<section class="animated swing flipInY infinite">html5</section>`// 同时应用好几个类动画，用空格隔开

## 26. CSS 过渡

通过 CSS3，我们可以在不使用 Flash 动画或 JavaScript 的情况下，当元素从一种样式变换为另一种样式时为元素添加效果。

transition 属性是一个简写属性，用于设置四个过渡属性：

transition-property  
transition-duration  
transition-timing-function  
transition-delay

注释：请始终设置 [transition-duration](#) 属性，否则时长为 0，就不会产生过渡效果。

语法

**transition: property duration timing-function delay;**//默认值：all 0 ease 0

**transition-property** 属性规定应用过渡效果的 CSS 属性的名称。（当指定的 CSS 属性改变时，过渡效果将开始）。

none	没有属性会获得过渡效果。
all(默认值)	所有属性都将获得过渡效果。
property	定义应用过渡效果的 CSS 属性名称列表，列表以逗号分隔。

**transition-duration** 属性规定完成过渡效果需要花费的时间（以秒或毫秒计）。

**transition-timing-function** 属性规定过渡效果的速度曲线。

linear	规定以相同速度开始至结束的过渡效果（等于 cubic-bezier(0,0,1,1)）。
ease（默认值）	规定慢速开始，然后变快，然后慢速结束的过渡效果（cubic-bezier(0.25,0.1,0.25,1)）。
ease-in	规定以慢速开始的过渡效果（等于 cubic-bezier(0.42,0,1,1)）。
ease-out	规定以慢速结束的过渡效果（等于 cubic-bezier(0,0,0.58,1)）。
ease-in-out	规定以慢速开始和结束的过渡效果（等于 cubic-bezier(0.42,0,0.58,1)）。
cubic-bezier(n,n,n,n)	在 cubic-bezier 函数中定义自己的值。可能的值是 0 至 1 之间的数值。

**transition-delay** 属性规定过渡效果何时开始，延时。



## 27. Viewport

**Viewport:** 视口，所有视频输出设备的显示区域。

桌面浏览器会忽略 **viewport** 设置，即使切换到响应式设计视图也会忽略 **viewport**，但可以手动调整到任何视口尺寸。

移动设备会读取 **viewport** 设置，并按照设置的要求渲染页面。

```
<meta name="viewport"
```

```
content="
```

```
height = device-height ,//height: 和 width 相对应，指定高度
```

```
width = device-width ,//width: 指定视口的逻辑宽度，控制 viewport
```

的大小，可以指定的一个值或者特殊的值，如 **device-width** 为设备的宽度（单位为缩放为 100% 时的 CSS 的像素），如果希望在移动设备上显示桌面级的 web 页面，需要设置 **width** 略大于页面的主体宽度。

```
initial-scale = 1.0 ,//initial-scale 的值不是用来相乘的，而是相除
```

的。

```
minimum-scale = 1.0 ,
```

```
maximum-scale = 1.0 ,
```

```
user-scalable = 0 ,
```

```
target-densitydpi = device-dpi
```

```
" />
```

//**target-densitydpi**: 一个屏幕像素密度是由屏幕分辨率决定的，通常定义为每英寸点的数量（**dpi**）。Android 支持三种屏幕像素 密度：低像素密度，中像素密度，高像素密度。一个低像素密度的屏幕每英寸上的像素点更少，而一个高像素密度的屏幕每英寸上的像素点更多。Android Browser 和 WebView 默认屏幕为中像素密度。

下面是 **target-densitydpi** 属性的 取值范围

- **device-dpi** –使用设备原本的 **dpi** 作为目标 **dp**。 不会发生默认缩放。
- **high-dpi** – 使用 **hdpi** 作为目标 **dpi**。 中等像素密度和低像素密度设备相应缩小。
- **medium-dpi** – 使用 **mdpi** 作为目标 **dpi**。 高像素密度设备相应放大， 像素密度设备相应缩小。 这是默认的 **target density**.
- **low-dpi** -使用 **mdpi** 作为目标 **dpi**。中等像素密度和高像素密度设备相应放大。
- **<value>** – 指定一个具体的 **dpi** 值作为 **target dpi**. 这个值的范围必须在 70–400 之间。



为了防止 Android Browser 和 WebView 根据不同屏幕的像素密度对你的页面进行缩放，你可以将 viewport 的 target-densitydpi 设置为 device-dpi。当你这么做了，页面将不会缩放。相反，页面会根据当前屏幕的像素密度进行展示。在这种情形下，你还需要将 viewport 的 width 定义为与设备的 width 匹配，这样你的页面就可以和屏幕相适应。

### initial-scale

初始缩放。即页面初始缩放程度。这是一个浮点值，是页面大小的一个乘数。例如如果你设置初始缩放为“1.0”，那么，web 页面在展现的时候就会以 target density 分辨率的 1:1 来展现。如果你设置为“2.0”，那么这个页面就会放大为 2 倍。

### maximum-scale

最大缩放。即允许的最大缩放程度。这也是一个浮点值，用以指出页面大小与屏幕大小相比的最大乘数。例如，如果你将这个值设置为“2.0”，那么这个页面与 target size 相比，最多能放大 2 倍。

### user-scalable

用户调整缩放。即用户是否能改变页面缩放程度。如果设置为 yes 则是允许用户对其进行改变，反之为 no。默认值是 yes。如果你将其设置为 no，那么 minimum-scale 和 maximum-scale 都将被忽略，因为根本不可能缩放。

所有的缩放值都必须在 0.01–10 的范围之内。

例：

(设置屏幕宽度为设备宽度，禁止用户手动调整缩放)

```
<meta name="viewport" content="width=device-width,user-scalable=no" />
```

(设置屏幕密度为高频，中频，低频自动缩放，禁止用户手动调整缩放)

```
<meta name="viewport" content="width=device-width,target-densitydpi=high-dpi,initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-scalable=no"/>
```

## 28. 表单

Resize 属性规定是否可由用户调整元素的尺寸

None(默认)	用户无法调整元素的尺寸。
----------	--------------

both	用户可调整元素的高度和宽度。
horizontal	用户可调整元素的宽度。
vertical	用户可调整元素的高度。

```
<form action="#">
  <fieldset>
    //出现一个框把里边的内容包裹起来了
    <legend>请选择</legend> //字出现在大框线上
    <div>
```

### 1) 单选框

```
    <input type="radio" id="boy" name="isMale" checked="1">
    //checked: 默认此项被选中，可以不写值，也可以写任意值
    //id 的作用：在页面上唯一的标识一个标签，只用于浏览器或者 JS（只
    在前端使用）
    //name：向服务端提交数据时使用 name，通常用于表单元素
    <label for="boy">男孩</label> //or 关联的是 input 里的 id
    <input type="radio" id="girl" name="isMale" checked>
    //name 相同的 radio 只能选中一个，如果两个相同 name 里都写了
    checked，只选中最后一个
    <label for="girl">女孩</label>
  </div>
</div>
```

### 2) 多选框

```
    <input type="checkbox" id="html5" name="skills" checked>
    <label for="html5">HTML5</label>
    <input type="checkbox" id="css3" name="skills">
    <label for="css3">CSS3</label>
    <input type="checkbox" id="javascript" name="skills">
    <label for="javascript">JavaScript</label>
  </div>
</fieldset>
</div>
```

### 3) 下拉选择菜单

```
    <select name="skill" id="skill" multiple> //multiple 鼠标拖动或按住 shift 或 Ctrl
    可多选
    <optgroup label="技能">
      <option value="1" selected>HTML5</option> //selected 默认选中 html5
```

```

        <option value="2" selected>CSS3</option>
        <option value="3">JS</option>
        <option value="4">PHP</option>
    </optgroup>
    <optgroup label="装备">
        <option value="5">jQuery</option>
        <option value="6">Bootstrap</option>
    </optgroup>
</select>

```

#### 4) 多行文本框

```

<textarea name="" id="" cols="30" rows="10"></textarea>
//row: 显示行数, 会+1 显示, cols: 列

```

#### 5) 文本框

```

<input type="text" placeholder="请输入账号">

```

#### 6) 密码框

```

<div class="controls">
    <label for="password">密码: </label>
    <input type="password" name="password" id="password" placeholder="请输入密码">
</div>

```

#### 7) 按钮

```

<input type="button" value="登 录">

```

#### 8) 媒体查询技术

```

@media(min-aspect-ratio:250/350) and (max-height:350px){
//media 媒介 aspect-ratio 长宽比、纵横比
    #forget{
        position: absolute;//flex: 会忽略绝对定位、固定位置
        bottom: 2vh;
        left: 14vw;
    }
}

```

**@media(min-aspect-ratio:1/1){**

//凡是长宽比例大于 1: 1 的，即横屏，执行下列变化

## 29. Less

### 1) 导入头文件

```
<link rel="stylesheet/less" href="test.less">
<script src="less.js"></script>
```

### 2) 样式

```
@btnColor:#ff9933;
@btnPadding:5px 20px;
@btnRadius:10px;
```

### 3) 基本格式

less 变量及赋值	@var.value;
less 函数	颜色、单位、数学函数、字符串函数...
less 继承/扩展	:extend( )
less Mixin/混入/混合	.name(@param:5px);

### 4) 基本 DOS 命令

lessc 命令	将 less 转化为 css//如 lessc test.less test.css
转换盘符	盘符名:
进入文件夹	cd 文件夹名字/path
列出文件	dir
清屏	cls
查看版本号	lessc -v
npm install -g less	查看有没有安装成功

### 5) 颜色操作函数

Saturate 饱和度  
Desaturate 去色调整  
Lighten 减轻，变成淡色，变亮  
Darken 变暗  
Fadein 淡入，渐显

Fadeout 淡出，渐没  
Fade 褪色  
Spin 旋转  
Mix 混合  
Tint 色相  
Shade 阴影  
Greyscale 灰度  
Contrast 对比度

## 30. 关于 PS

快捷键：

调整图像大小快捷键：alt+ctrl+i

关闭文件：ctrl+w（会提示要不要保存）

另存为：ctrl+shift+s

选框工具：m（选定区域后，ctrl+c，ctrl+n 新建一个文件，跟选定区域一般大，  
ctrl+v 粘贴上去）

应用前景色：alt+delete

应用背景色：ctrl+delete

图片尺寸为 100%：ctrl+1

图片填满工作区：ctrl+0

放大：ctrl+

缩小：ctrl-

标尺显示或隐藏：ctrl+r（在标尺上右键可改变标尺单位，一般用像素）

标尺辅助线显示或隐藏：ctrl+h（视图--->新建参考线--->位置可设置为百分数）

魔棒：w

移动工具：v（即第一个图标工具，箭头）

## 31. Bootstrap

Lg	large 大
Md	middle 中
Sm	small 小
Xs	extra small 超小