

Emmet语法

快速生成HTML结构语法:

- 生成标签直接输入标签名按tab键即可
- 想要生成多个标签, 加上`就可以`了, 比如`div3`就可以生成3个div(打`div*3`时中间不能有空格)
- 如果有父子级的标签, 可以用`>`, 比如`ul>li`
- 如果有兄弟关系的标签, 用`+`就可以了, 比如`div+p`
- 如果生成带有类名或者id名字的, 直接写`.demo`或者`#two` tab键就可以了(“.”是带类的, “#”是带id的)
- 如果生成的div类名是有顺序的, 可以用自增符号`$`
- 如果想要在生成的标签内部写内容可以用`{}`表示(内容写在大括号里面)

快速生成css样式语法:

css基本采取简写的形式即可, 例如:

```
text-align:center;=tac+tab键
text-indent:2em;=ti2em+tab键
width:100px;=w100+tab键
line-height:26px=lh26px+tab键(px不能少打)
text-decoration:none;=tdn+tab键
.....
```

快速格式化代码

右键选择格式化代码或者`shift+alt+f`

css的复合选择器

复合选择器是基础选择器的组合

后代选择器

又称为包含选择器, 可以选择父元素中的子元素

- **快速复制所在行到下一行: `alt+shift+下`**

语法: 元素1 元素2 {样式声明} 元素1、2中间有一个空格

- 最终选择的是最后一级元素
- 可以多重嵌套: 元素1 元素2 元素3{样式声明}
- 元素1和元素2可以是任意基础选择器

子选择器

只能选择作为某元素最近的一级子元素

语法：元素1>元素2{样式声明}

- 元素2必须是元素1的亲儿子

并集选择器

可以选择多组标签，同时为他们定义相同的样式，通常用于集体声明

通过英文逗号 (,)连接而成，任何形式的选择器都可以作为并集选择器的一部分

语法：元素1,元素2{样式声明} 例如：

```
ul,div{样式声明}
```

并集选择器都喜欢竖着写，如：

```
div,  
p,  
.pig li{  
  color:pink;  
}
```

伪类选择器

用于向某些选择器添加特殊的效果，比如给连接添加特殊效果，或选择第1个，第n个元素

伪类选择器书写的最大特点是用冒号表示，如：

```
hover,:first-child
```

链接伪类选择器

- a:link 选择所有未被访问的连接
- a:visited 选择所有已被访问的连接
- a:hover 选择鼠标指针位于其上的链接
- a:action 选择活动链接（指针按下未弹起的链接）
-

注意事项：为了确保生效，请按照lvha的顺序声明：**link->visited->hover->active**

a链接在浏览器中具有默认样式，所以我们实际工作中都需要给链接单独指定样式

链接伪类在实际开发中的写法：

```
a{
  color: gray
}
a:hover{
  color:red;
}
```

focus伪类选择器

用于获取焦点的表单元素 焦点就是光标，一般情况<input>类表单元素才能获取，因此这个选择器也主要针对与表单元素来说

语法:

```
input:focus{
  background-color:yellow;
}
```

元素显示模式

元素显示模式就是元素（标签）以什么方式进行显示

HTML元素可以分为块元素和行内元素两种

块元素

<div>标签是最典型的块元素

块级元素特点:

- 自己独占一行
- 高度，宽度，外边距，内边距都可以控制
- 默认宽度是容器（父级宽度）的100%
- 是一个容器及盒子，里面可以放行内或者块级元素
-

注意事项:

- 文字类的元素内不能放块级元素
- <p>标签主要存放文字，不能放块级元素

切记: <p>标签中不能放<div>标签 <h1>~<h6>等都是文字类块级标签，里面不能放其他块级元素

行内元素

也叫内联元素 是最典型的行内元素

特点:

- 相邻的行内元素，在一行上可以显示多个
- 高宽直接设置是无效的
- 默认的宽度就是其内容本身的宽度
- 行内元素只能容纳文本或其他行内元素

注意事项：

- 链接内不能放链接
- 特殊情况链接里面可以放块级元素，但是给

行内块元素

行内元素中有几个特殊的标签 <input> <td>,它们同时具有块元素和行内元素的特点

特点：

- 和相邻行内元素（行内块）在一行上，但是他们会有空白缝隙，一行可以显示多个（行内元素特点）
- 默认宽度就是其内容本身的宽度（行内元素特点）
- 高度，行高，外边距，内边距都可以控制（块级元素特点）

元素显示模式转换

特殊情况下，我们需要元素模式的转换，简单理解：一个模式的元素需要另一种模式的特性

比如想要增加链接

语法：

```
转换为块级元素：display: block;(最常用)
转换为行内元素：display:inline;
转换为行内块：display:inline-block; (比较常用)
```

实现单行文字垂直居中：

css并没有提供文字垂直居中的代码，这里可以用一个小技巧来实现：让文字的行高（line-height)等于盒子的高度，就可以让文字在当前盒子内垂直居中

CSS背景

背景颜色

background-color 一般情况下元素背景颜色默认值是transparent（透明），也可以手动设置背景颜色为透明色

背景图片

background-image属性描述了元素的背景图像 实际开发常见于logo或者一些装饰性的小图片或者是超大的背景图片，优点是便于控制位置

```
background-image:none|url(url)
```

背景平铺

如果需要在HTML页面上对被景图像进行平铺，可以使用background-repeat属性

```
background-repeat:repeat|no_repeat|repeat-x|repeat-y
```

- 默认为repeat
- 页面元素既可以添加背景图片也可以添加背景颜色

背景图片位置

background-position 语法：

```
background-positin:x y;
```

- **1.参数是方位名词** 如果x,y两个参数是方位名词，则两个值的前后顺序无关，比如left top 和 top left 的效果一样 如果只指定了一个方位名词，另一个值省略，则第二个默认居中对齐
- **2.参数是精确单位** 如果参数值是精确坐标，那么第一个肯定是x坐标，第二个肯定是y坐标 如果只指定一个数值，那该数值一定是x坐标，另一个默认垂直居中
- **3.参数是混合单位** 如果指定的两个值是精确单位和方位名词混合使用，则第一个值是x坐标，第二个值是y坐标

背景图像固定（背景附着）

background-attachment属性设置图像是否固定或者随着页面其余部分滚动

语法：

```
background-attachment: scroll|fixed
```

背景复合写法

5个背景属性简写在一个属性background中，节省代码量 当简写属性时，没有特定的书写顺序，一般习惯约定顺序为：**background:背景颜色 背景图片地址 背景平铺 背景图像滚动 背景图片位置**

背景色半透明

```
background rgba(0,0,0,0.3)
```

- 最后一个参数是alpha透明度，取值范围在0~1之间
- 我们习惯把0.3的0省略掉，写为background: raba(0,0,0,.3); 注意，背景色半透明是指盒子背景半透明，盒子内容内容不受影响

竖向输入同一个内容: *shift+alt* 不松开向下拖动，之后输入即可