

CSS简介

CSS样式类别

选择器介绍

样式-属性

文字样式-font

文本样式-text

背景样式-background

链接样式

表单样式

列表样式

表格样式

盒模型 (box model)

块级元素和内联元素的区别 (含: inline-block布局vs float布局)

浮动-float

定位-position

版心及布局

三大特性:

补充 (过渡、转换、精灵图)

## CSS简介

CSS (Cascade Style Sheet) 层叠样式表

由CSS样式规则组成, 用来修饰HTML标签怎样在页面中显示。

CSS规则样式:

选择符{属性1: 值; 属性2: 值; }

每个选择符的复合样式应该用分号隔开。如果值为多个单词，则要给值输引号。

## CSS样式类别

- 内联样式：在HTML元素内部，使用如下：

```
<input style="width:20px; height:20px; background:red; color:green; border(线宽):  
线宽 颜色 线型;" />
```

- 内部样式：在<head>头部，使用如下：

```
<style type="text/css">  
  选择器{属性1： 值； 属性2： 值； }  
</style>
```

- 外部样式：创建CSS样式表，连接至源文件，link在<head>中。

```
<link type="text/css" rel="stylesheet" href="样式表位于这个文件的相对连接" />
```

ps: text/css 表明信息的样式为CSS文件；rel属性表明html文件和所要连接的文档间的关系，固定为stylesheet。

- 浏览器样式：浏览器自带样式

说明：CSS样式优先级依次降低。

## 选择器介绍

**1.标签样式选择器：**又叫元素选择器，最常见。直接以标签名命名的选择器，选中的是html中所有的此标签。

**2.类选择器：**1.class选择器用于描述一组元素的样式，class选择器可以在多个元素中使用，可以指定特定的HTML元素使用。2.在CSS中，类选择器以一个“.”显示。3.在html中在开始标签和单标签中命名，命名方式为<p class="duanluo">。4.class命名，第一个字符不能使用数字.部分浏览器不识别。（经常在CSS中使用）

**3.id选择器：**1.同一个id属性只能在HTML出现一次，同一个标签也只能有一个id名。2.在html中用id属性引用，eg：<h id="title">。3.CSS中id选择器以“#”定义。4.id属性命名不要以数字开头，一般用在网页的开头或结尾。（id 在js中使用多）

优先级：id>class>标签样式

**4.伪类选择器：**详情请见“链接样式”。中间带冒号“:”，用来说明选择器功能。：

link、: visited、: hover、: active、: selection（选中时改变）、:focus(获取焦点，常用于表单)、: after、: before（之后，常用于消除浮动）、: first-child（首个子对象）、: nth-child (3)、: lang（向带有指定 lang 属性的元素添加样式）

**5.子孙、后代选择器（又叫派生选择器）：**中间用空格隔开，用来向浏览器具体制定同类标签中的哪个标签。

**6.子元素选择器：**h1>strong{color:red}，只能选择作为某元素子元素的元素

**7.相邻兄弟选择器：**h1 + p {margin-top:50px;} 选择紧接在另一元素后的元素，且二者有相同父元素

**6.并集选择器（选择器的分组）：**h1,h2,h3,h4,h5,h6 { color: green; }

**7.属性选择器：**对带有指定属性的 HTML 元素设置样式。在 IE6 及更低的版本中，不支持属性选择。

eg1: [title]{color:red;}，eg2: [title~="W3School"]{border:5px solid blue;} = ^ +

**8.通配符、\*选择器：**统统匹配，把所有的元素选择。\*{ margin: 0px ; padding: 0px}

## 样式-属性

### 文字样式-font

font-family, 字体，属性值=各种字体

font-style, 字体样式，属性值：italic（倾斜）normal清除格式

font-weight, 字体粗细，属性值：bold, bolder, lighter.....normal清除格式

font-size, 字体大小，属性值：px/em/%/数字，normal清除格式

color, 字体颜色，属性值：颜色值

### 文本样式-text

text-align, 对齐，属性值：left, right, center, *justify*（两端对齐）

**vertical-align, 垂直位置对齐，属性值：top、bottom/middle--通用于表格，使用时需将元素转换成table；text-top、text-bottom、middle--适用于图文混排，注意，此时class选择器应该放在img标签中。**

text-indent, 缩进，属性值：n em/%

line-height, 行高，属性值：px/n em/%

letter-spacing, 字符间距，属性值：em/px

word-spacing, 字间距，改变**单词**间的间隔，属性值：默认normal

text-transform, 字符转换，**处理文本（英文）的大小写**，属性值：uppercase（大写）

lowercase（小写）capitalize（首字母大写）

text-decoration, 文本装饰，属性值：none underline overline line-through

text-shadow, 文本阴影，（**水平阴影**，*v-shadow, blur, color*）

box-shadow

### 背景样式-background

background-color, 背景色，颜色值

background-image: url（路径），背景图片，图片路径 默认铺满整个盒子

background-repeat, 背景图片重复样式, repeat-x, repeat-y, no-repeat  
background-position, 背景定位, 属性值有三种表示方法: center center; right  
bottom ; left top; 100px 200px (先水平再垂直) ; 10% 20%;  
background-attachment: fixed, 背景关联, 背景图片位置固定, 不会随着图片滚动跑掉。该属性与background-position同时使用时, position属性值用px。  
background: 背景总属性, 将以上属性统筹均可在此处属性值中显示  
transparent 清除背景色

## 链接样式

1、链接样式, 又叫伪类选择器, 该选择器可以使用任何CSS属性, 如字体、背景、颜色等。

a:link      正常, 未访问过的链接状态  
a:visited   被访问过的链接状态  
a:hover     当鼠标划过时的状态  
a:active    当鼠标点击时的状态

ps: 1.a:hover必须跟在a:link和a:visited之后; 2.a:active必须跟在a:hover之后; 3. 实际使用时, 我们常把a:link和a:visited合并成a使用。

2、链接属性。

下划线的设置 text-decoration none/underline

字体颜色 color

指定链接背景 background-color transparent 清除背景色

## 表单样式

background-color 背景色: 颜色值;

border (border-width、border-color、border-style) 边框: px red solid (顺序可以颠倒)

border-radius 边框样式: 20px、%

width/height

## 列表样式

对应标签: <ul><ol><dl><dt><dd><li>

去掉序列标注 list-style: none;

设置列表项标志的类型 list-style-type: disc、circle、square; lower-roman, upper-alpha

**列表项的图像标记 list-style-image: url**

经常配合子孙选择器使用, 举例如下:

```

ul{
    list-style-type:square;
}
.ol1 li{
    list-style:none;
}
.ol2 li{
    list-style-image:url(qq.png)
}

```

## 表格样式

对应标签：<table> <th> <tr> <td>

边框、边框曲线、背景颜色、尺寸、文本颜色、文本对齐

内边距 padding：20px；表格内边距填充在table中，文本与边框的距离在td中；

单元格间距 border-spacing：20px；在<table>中。

## 盒模型 (box model)

CSS将每个**元素**看做由一个盒子表示，每个盒子均由“内容区、内边距padding、边框border、外边距margin”四部分组成。

**1.内容区：**包含文字、图片、列表、表格等所有内容

**2.内边距（内填充）：**

a.复合属性，【padding-left/top/bottom/right: 10px;】；

b.【padding: 10px 30px 40px 50px;】顺时针方向显示。值为1个，默认四个方向值一致；为2个，默认第一个值上下，第二个值左右；值为3个，第一个值代表上，第二个至代表左右，第三个值代表下。

**3.外边距：**

a.复合属性，用法参考padding属性；

b.默认是透明的，因此不会遮挡其后的任何元素。

c.垂直方向：多个元素外边距一起会自动重合，外边距默认取最大值；水平方向：外边距不发生重合，正常相加。

d.外边距可以是负值（正值向右移，负值向左移）

f.margin属性用于设置盒子的对齐，margin:0px auto;用来设置盒模型居中对齐（text-align只能设置文本内容的对齐，）

g.<h><p>自带margin

**4.边框：**

a.border: red 20px solid;

b.复合属性: border-left/right/top/bottom, border-bottom-color, border-bottom-left-radius。

## 5.其他内容补充:

a.元素总宽度=内容宽度+padding左+padding右+border\*2+margin左+margin右

b.背景应用于由内容和内边距、边框组成的区域。

c.内边距颜色同内容区。

d.百分数值是相对于其父元素的 **width** 计算的, 而不是相对于高度

e.盒模型嵌套时, 默认从父盒子左上开始;

f.margin塌陷问题解决:

i.利用border, 给父盒子一个边框, 不可以给子盒子;

ii.给父盒子设置overflow: hidden;

iii.在父盒子中打一些文字, 设置大小为0;

g.真正的高手一般不设置盒子的宽高, 盒子的大小可以被它所装的内容撑开;

## 块级元素和内联元素的区别 (含: inline-block布局vs float布局)

1.元素: 一对标签的总称 (元素与标签的区别), 可以相互转换。

### 2.block块级元素:

i.独立成行;

ii.支持设置宽高:

1.不设置宽度时, 宽度是父元素内容区域的宽度, 即浏览器整个宽度

2.不设置高度时, 高度由内容撑开

3.设置宽度和高度时, 高度和宽度不能撑开

///margin, padding 四个方向支持, margin:auto支持

iv.可以包含某些块级元素或内联元素 (<h><p>标签只能包含内联元素, 不可包含块级元素。)

### 3.inline内联 (行内) 元素:

i.和其他内容共成一行;

ii.不可以设置宽高, 完全由内容撑开宽高;

iii.margin只支持左右 不支持上下 margin:auto不支持;

iv.padding支持显示 但是上下的padding撑大的部分对别的元素没有影响;

v.行内元素和行内元素之间会产生空格 (包括标签之间的空格, 换行)。解决办法: 给body标签一个设置字体大小为0, 再用他自己的选择器设置字体。

vi.存在基线对齐问题

iv.不可以包含其他任何元素。

**4.inline-block块级内联元素（又叫行内块）：**融合行内于块级，不独立成行，但可以设置宽高，margin，padding属性全部奏效。通俗点讲就是不独占一行的块级元素。input、img为行内块元素。

	inline	block	inline-block
独立成行	X	V	X
设置宽高	X	V	V
Margin-top/bottom	X	V	V
可否包含其他块级元素	X	V	V

## 5.元素转换：

display: inline; 把块级元素转换成内联元素

display: block; 让元素转换成块级元素，**让消失或隐藏的元素显示出来**

**display: inline-block;** 让元素同时具有内联和块级元素的功能

display: none; 实现元素的隐藏，同时不占位置（visibility: hidden; 实现元素的隐藏，但占据原位置）

display: table-cell; 让元素转换成单元格，配合vertical-align使用。

visibility: hidden; 实现元素的隐藏，但占据原位置

## 6.inline-block布局VS float布局：

a.对于横向排列东西来说，倾向与使用inline-block来布局，因为这样清晰，也不用再像浮动那样清除浮动，害怕布局混乱等等。inline-block布局时设置父容器文字为零{font-size: 0; }

b.对于浮动布局就用于需要文字环绕的时候，毕竟这才是浮动真正的用武之地，水平排列的是就交给inline-block了。

## 7.扩展隐藏样式：

display: none/block; 隐藏不占位置

visibility:hidden/visible 隐藏占有位置,就好比停职留薪

overflow:hidden/visible 隐藏超出的部分

## 浮动-float

float:left/right

1.脱离文档流，不占位置，其他盒子为标准文档流正常显示

2.并列浮动：几个盒子同样的浮动，依次排列起来。当浏览器装不下时，排列规则：如果该盒子很小被卡住时，在卡住的地方显示，如果该盒子很大放不下时，向下走继续从头排列。

（换句话说：如果包含框太窄，无法容纳水平排列的三个浮动元素，那么其它浮动块向下移动，直到有足够的空间。如果浮动元素的高度不同，那么当它们向下移动时可能被其它浮动元素“卡住”）

3.嵌套浮动：嵌套在盒子中的浮动不出父盒子。



4.文字环绕效应：当浮动遇到文字时，按照标准文档流显示，文字不会出现在浮动下方，而是环绕显示，浮动最早是用来做文字环绕效果的。

5.清除浮动：clear属性

a.为什么要清除浮动？一般情况下，我们只是使用浮动来完成文档的水平布局，但我们并不需要让其他盒子内容隐藏在浮动盒子下方，而是要正常显示，且希望嵌套浮动的父盒子大小被盒子所包含内容自动撑开缩放。例如淘宝商品首页，当商品多时，盒子变大，商品小时，盒子变小。在保留浮动布局的同时，父盒子自动收放，缩放自如。

b.怎么实现清除浮动？参考资料：

<https://www.cnblogs.com/nxl0908/p/7245460.html>

最佳方法，给父盒子定义伪类after和zoom：

【.clearfloat:

after{content:"";display:block;clear:both;height:0px;visibility:hidden;

.clearfloat{zoom:1}】

其他方法：一，给父盒子添加高度；二、子盒子、br加clear: both（全部浮动清除）；父盒子添加overflow:hidden/auto；父子divYIQI FUDONG ;父盒子定义table: display: table; **父盒子定义伪类**：after和zoom,.clearfloat:after{content:"";display:block;clear:both;height:0px;visibility:hidden;}

## 定位-position

1.相对定位：【position:relative;left:100px;top:100px;】，以自身左上角为基准进行位置调整；原位置保留，不会被取代；层级变高，层级相同的情况下，后来者居上，会覆盖其他盒子；**相对定位脱离文档流了吗？**

2.绝对定位：【position:absolute;left:100px;top:100px;】，以父级左上点为基准调整位置；脱离标准文档流，不占位置，被后来者占领位置；层级较后来者高；（效果类似浮动）

3.固定定位：【position:fixed;right:10px;bottom:10px;】，以浏览器窗口为基准调整位置，couser:pointer/move（模拟鼠标的手势）；

4.取消定位：【position:static;】

5.**原则总结**：定位的子元素会找离他最近的加了定位的祖先元素；这个祖先元素一般会添加相对定位（我们不希望它飘起来）；这个定位元素（需要操作的元素）我们一般添加绝对定位（我们希望它飘起来不占位置，方便布局）。--子绝父相

## 版心及布局

一句话：真正的高手都不设置盒子的宽和高的，

wed前端不仅要兼顾浏览器的兼容性，还要照顾到浏览器缩放过程中内容正常显示；



cursor: pointer;move pointer

## 三大特性：

层叠性：样式冲突，就近原则；样式不冲突，各自执行；

继承性：子标签会继承父标签的某些样式，例如：text-，line-，font-，颜色；

优先级：继承<标签<class<id<行内样式<!important（无穷大） 四位算法0，0，0，1，不进位；

## 补充（过渡、转换、精灵图）

过渡：transition：all（过渡元素，多种分开描述时用逗号隔开） 2s（过渡时间） 方式，元素从一种样式逐渐改变为另一种的效果，搭配：hover使用

转换：transform：rotate（0），transform：rotate（360），它有多种转换方式，我们需要使用的它的旋转rotate()，旋转

兼容处理：-webkit-transtation -moz-

精灵图：background-position：-115px,-130px；第一个值代表水平方向，第二个值代表垂直方向，正值向右向下，负值向左向上。【css sprites 可翻译为css精灵，也有人翻译为css雪碧，因为sprite也有雪碧的意思。在国内习惯称它为图片整合技术，因为他确实是多张图片整合为一整图片的技术。这样做有什么好处呢？因为这样的话：①**当访问该页面时，载入的图片就不会像以前那样一幅一幅地慢慢显示出来了**②**降低浏览器向服务器的请求次数，大大提高网页的加载速度（重点面试题）**】

特殊处理：使用一像素处理法，repeat-x

reset：样式重置的CSS