[2021.6.1]

前端基础页面布局及样式学习

摘要

[HTML5+CSS3]

[日期及邮编地址]

[ID名称]

[前端学习笔记]

目录

[HTML 5 7](#_Toc20670)

[Html基本结构 7](#_Toc1490)

[基础标签语义 7](#_Toc29791)

[标题标签<h1>—<h6>：作为标题使用，并且依据重要性递减 7](#_Toc28108)

[段落标签<p> </p>：把html文档内容分割为若干段落 7](#_Toc9974)

[换行标签<br />：将文本强制换行 7](#_Toc27930)

[文本格式化标签：为文字设置粗体、斜体、下划线等 7](#_Toc24570)

[布局标签<div> </div>和<span> </span>：无语义，单纯的盛放内容,div是division缩写，表示分割分区。span意为跨度、跨距 7](#_Toc10464)

[图像标签<img>：单标签，定义html页面中的图像 8](#_Toc21834)

[路径 8](#_Toc14254)

[根目录：打开目录文件夹的第一层就是根目录 8](#_Toc8912)

[相对路径：以引用文件所在位置为参考建立出的目录路径。如：images/XXX.jpg编程常用 8](#_Toc3823)

[绝对路径：目录下的绝对位置，通常从盘符开始。如C:\User\XXX\XX 。编程不常用 8](#_Toc21682)

[超链接标签 8](#_Toc27892)

[链接分类 8](#_Toc4026)

[注释 9](#_Toc11319)

[特殊字符 9](#_Toc12802)

[表格标签 9](#_Toc4549)

[表格属性 10](#_Toc22080)

[表格结构标签 10](#_Toc30708)

[合并单元格 10](#_Toc13851)

[列表标签 10](#_Toc32719)

[无序列表（重点）：<ul> </ul>，列表项用<li> </li>标签定义 10](#_Toc3178)

[有序列表：<ol> </ol>，列表项用<li> </li> 11](#_Toc1759)

[自定义列表（重点）：<dl> </dl> ，定义项目/名字用<dt> </dt>，描述项目/名字用<dd> </dd> 11](#_Toc24158)

[表单标签 11](#_Toc19512)

[表单域：<form> </form>定义表单域，包含表单元素的区域，实现用户信息的的收集和传递。<form>会把它范围内的表单元素信息提交给服务器 11](#_Toc2772)

[表单元素 11](#_Toc18165)

[HTML5新特性 13](#_Toc14695)

[HTML5新增语义化标签 13](#_Toc388)

[HTML5新增多媒体标签 13](#_Toc20225)

[视频标签<video> 13](#_Toc10361)

[音频标签<audio> 14](#_Toc31735)

[HTML5新增的input类型 15](#_Toc12473)

[HTML5新增的表单属性 15](#_Toc22513)

[CSS 16](#_Toc22588)

[CSS概述 16](#_Toc25137)

[CSS推荐代码风格： 16](#_Toc7617)

[选择器（选择符）：选择html标签 16](#_Toc12534)

[1. 标签选择器:选择页面中某一大类的全部标签 17](#_Toc28444)

[2. 类选择器（常用）：实现差异化、单独、某些标签的样式更改 17](#_Toc12554)

[3. id选择器：与类部分相似，id属性只能在每个html文档里出现一次，只能调用一次 17](#_Toc2831)

[4. 通配符选择器：更改当前html文档全部标签的样式，只需在<style>中定义就能自动生效，不需要主动调用 17](#_Toc7875)

[CSS字体： 18](#_Toc15051)

[font-family属性定义文字的字体 18](#_Toc27643)

[font-size设置文字的大小，font-size: 20px; 配合选择器使用 18](#_Toc26758)

[font-weight设置文字的粗细，font-weight: normal/bold/bolder/lighter/数字 18](#_Toc8902)

[font-style设置文字的风格，font-style: normal/italic 18](#_Toc978)

[font复合属性，将文字属性整合到一条语句，减少代码量，属性设置有先后顺序，配合选择器使用。font: 属性值1 属性值2 属性值3 ... 18](#_Toc4856)

[CSS文本属性 18](#_Toc27030)

[color设置文本的颜色，配合选择器使用 18](#_Toc18616)

[text-align设置文本水平对齐，配合选择器使用 19](#_Toc29909)

[text-decoration添加修饰，下划线、删除线、上划线等，配合选择器使用 19](#_Toc31910)

[text-indent设置首行缩进，通常设置段落首行缩进，配合选择器使用 19](#_Toc18088)

[line-height设置行间距，行间距=上间距+文本高度+下间距，配合选择器使用 19](#_Toc29298)

[CSS引入方式 19](#_Toc1816)

[1. 行内样式表 19](#_Toc8393)

[2. 内部样式表 19](#_Toc11708)

[3. 外部样式表（推荐） 20](#_Toc5634)

[chrome调试工具 20](#_Toc6210)

[emmet语法 20](#_Toc26363)

[快速生成html结构语法 20](#_Toc9808)

[快速生成CSS样式语法 20](#_Toc8702)

[快速格式化代码： 20](#_Toc3425)

[复合选择器 20](#_Toc19876)

[后代选择器（重要） 21](#_Toc27780)

[子选择器（重要） 21](#_Toc31768)

[并集选择器（重要） 21](#_Toc10031)

[伪类选择器（可用于所有元素，不限于a标签） 21](#_Toc32038)

[链接伪类 22](#_Toc1917)

[focus伪类选择器 22](#_Toc19955)

[CSS元素显示模式 22](#_Toc27540)

[块元素有<h1>-<h6>、<p>、<div>、<ul>、<ol>、<li>等，<div>是最典型的块元素 22](#_Toc7000)

[行内元素有<a>、<strong>、<b>、<em>、<i>、<del>、<s>、<ins>、<u>、<span>等，<span>是最典型的行内元素，行内元素也称内联元素 22](#_Toc7943)

[行内块元素 22](#_Toc5173)

[行内块元素特点 23](#_Toc18539)

[元素显示模式转换 23](#_Toc29494)

[CSS背景属性 23](#_Toc21630)

[背景颜色 23](#_Toc9182)

[背景色半透明效果 23](#_Toc21385)

[背景图片 23](#_Toc17623)

[背景图片平铺 23](#_Toc8942)

[背景图片的位置 23](#_Toc6616)

[背景图固定 24](#_Toc19451)

[背景属性复合写法 24](#_Toc20895)

[扒背景图片 24](#_Toc2681)

[div相对位置 24](#_Toc26830)

[CSS三大特性：层叠性、继承性、优先级 24](#_Toc16528)

[层叠性 24](#_Toc30901)

[继承性 24](#_Toc2809)

[行高继承 24](#_Toc31042)

[优先级 25](#_Toc32664)

[选择器权重 25](#_Toc30647)

[权重叠加 25](#_Toc12775)

[盒子模型 25](#_Toc4058)

[看透网页布局的本质 25](#_Toc13290)

[盒子模型组成 25](#_Toc9133)

[边框border 26](#_Toc16252)

[内边距padding 26](#_Toc28155)

[外边距margin 27](#_Toc21292)

[清除li的小圆点 28](#_Toc11836)

[CSS3新增特性（IE9及以上支持） 28](#_Toc10005)

[圆角边框 28](#_Toc24729)

[盒子阴影 28](#_Toc16718)

[文字阴影 28](#_Toc19229)

[浮动 29](#_Toc28380)

[标准流（普通流/文档流） 29](#_Toc12622)

[浮动最典型应用：可以让多个块级元素一行内排列显示 29](#_Toc19800)

[浮动float 29](#_Toc18893)

[浮动特性（重难点） 29](#_Toc12965)

[浮动布局注意点 32](#_Toc11689)

[清除浮动 32](#_Toc27494)

[为什么要清除： 32](#_Toc30597)

[清除浮动方法： 32](#_Toc19990)

[1. 额外标签法 32](#_Toc14536)

[2. 父级添加overflow 32](#_Toc20464)

[3. :after伪元素法 32](#_Toc23916)

[4. 双伪元素法 33](#_Toc29195)

[CSS属性书写顺序（重要） 33](#_Toc6953)

[页面布局整体思路 33](#_Toc20559)

[定位模式 34](#_Toc12044)

[边偏移 34](#_Toc10552)

[静态定位static 34](#_Toc1213)

[相对定位relative（重点） 34](#_Toc11389)

[绝对定位absolute（重点） 34](#_Toc32002)

[子绝父相（重点） 35](#_Toc4113)

[固定定位fixed（重点） 35](#_Toc30014)

[粘性定位sticky 35](#_Toc9757)

[定位总结 36](#_Toc32762)

[定位叠放次序z-index 36](#_Toc17382)

[绝对定位盒子的居中 36](#_Toc32451)

[定位特殊特性 36](#_Toc2235)

[定位对标准流的文字的影响 36](#_Toc4059)

[网页布局总结 36](#_Toc19795)

[元素的显示和隐藏 37](#_Toc5778)

[display显示隐藏（重点） 37](#_Toc1876)

[visibility可见性 37](#_Toc6975)

[overflow溢出 37](#_Toc2519)

[CSS高级技巧 37](#_Toc26677)

[精灵技术 37](#_Toc4889)

[为什么需要精灵图： 37](#_Toc13094)

[精灵图的使用 38](#_Toc22723)

[字体图标 38](#_Toc24272)

[字体图标优点 38](#_Toc3952)

[字体图标的使用 38](#_Toc32491)

[CSS三角 39](#_Toc3926)

[CSS用户界面样式 39](#_Toc22705)

[鼠标样式cursor 39](#_Toc1852)

[轮廓线outline 40](#_Toc11383)

[文本域防拖拽resize 40](#_Toc30758)

[vertical-align 40](#_Toc14633)

[解决图片底部默认空白缝隙问题 40](#_Toc18270)

[溢出的文字用省略号代替 40](#_Toc12468)

[单行情况： 40](#_Toc25025)

[多行情况： 40](#_Toc19610)

[布局技巧 41](#_Toc636)

[margin负值的应用 41](#_Toc27716)

[文字围绕应用 41](#_Toc149)

[行内块巧妙运用 41](#_Toc13010)

[CSS三角巧妙运用 41](#_Toc25088)

[CSS初始化 41](#_Toc2164)

[CSS3新特性 41](#_Toc10253)

[CSS3新增选择器 41](#_Toc17303)

[属性选择器 41](#_Toc10340)

[结构伪类选择器 42](#_Toc8299)

[伪元素选择器（重点） 43](#_Toc7933)

[伪元素法清楚浮动详解 43](#_Toc8579)

[CSS3盒子模型 44](#_Toc18606)

[CSS3其他特性 44](#_Toc2728)

[filter滤镜： 44](#_Toc4928)

[calc函数： 44](#_Toc28435)

[CSS过渡transition（重点） 44](#_Toc5942)

[2D转换transform 45](#_Toc22187)

[移动translate 45](#_Toc28832)

[旋转rotate 45](#_Toc23034)

[转换中心点transform-origin 46](#_Toc30639)

[缩放scale 46](#_Toc21233)

[flex布局 46](#_Toc7925)

[传统布局： 47](#_Toc28859)

[flex弹性布局： 47](#_Toc11365)

[布局原理 47](#_Toc12070)

[常见父元素属性 47](#_Toc27537)

[flex-direction属性决定主轴的方向（即项目的排列方向） 47](#_Toc30326)

[justify-content设置主轴上的子元素排列方式 47](#_Toc10948)

[flex-wrap设置子元素是否换行 48](#_Toc3735)

[align-items设置侧轴上得到子元素排列（单行） 48](#_Toc8668)

[align-content设置侧轴上的子元素排列方式（多行） 48](#_Toc10498)

[flex-flow 48](#_Toc6426)

[flex布局子项常见属性 48](#_Toc7910)

[flex 48](#_Toc5948)

[align-self控制子项自己在侧轴上的排列方式 48](#_Toc11581)

[order属性定义项目的排列顺序 48](#_Toc22942)

HTML 5

Html标签必须在尖括号里<html>,格式：<开始> </结束>

Html标签关系：1.包含关系 2.并列关系

Html基本结构

<!DOCTYPE html>

<html lang=”en”>

<head>

<meta charset=”UTF-8”>

<title> </title>

</head>

<body>

<h1> </h1>

</body>

</html>

<!DOCTYPE html>声明该文档使用html5

<html lang=”en”>表明网页为英文网页，zh-CN为中文网页

<meta charset=”UTF-8”>设定字符编码,utf8为万国码

基础标签语义

标题标签<h1>—<h6>：作为标题使用，并且依据重要性递减

1. 加了标题的文字会加粗，字号也会一次变大
2. 一个标题独占一行

段落标签<p> </p>：把html文档内容分割为若干段落

1. 文本在一个段落中会根据浏览器串口的大小自动换行
2. 段落之间保有空隙

换行标签<br />：将文本强制换行

1. 换行标签是单标签
2. 换行标签知识简单的开始新的一行

文本格式化标签：为文字设置粗体、斜体、下划线等

<strong> </strong>或<b> </b>：加粗，推荐<strong>语义更强烈

<em> </em>或<i> </i>：倾斜，推荐<em>语义更强烈

<del> </del>或<s> </s>：删除线，推荐<del>语义更强烈

<ins> </ins>或<u> </u>：下划线，推荐<ins>语义更强烈

布局标签<div> </div>和<span> </span>：无语义，单纯的盛放内容,div是division缩写，表示分割分区。span意为跨度、跨距

1. <div>用来布局，独占一行，是个大盒子
2. <span>用来布局，一行上可以有多个<span>小盒子

图像标签<img>：单标签，定义html页面中的图像

<img scr=”图像url” />

**src**：必要属性，指定图像文件的路径和文件名

alt：替换文本，图像不能显示时展示的文字

title：提示文本，鼠标放到图像上显示的文字

width：设置图像的宽度

height：设置图像的高度

border：设置图像的边框粗细

1. 图像标签可以有多个属性，必须写在标签名的后面
2. 属性之间部分先后顺序，之间必须用空格分开
3. 属性采取键值对的格式，即key=”value”

路径

目录文件夹：普通文件夹，里面存放做页面所徐娅萍的相关素材如html文件，图片

根目录：打开目录文件夹的第一层就是根目录

相对路径：以引用文件所在位置为参考建立出的目录路径。如：images/XXX.jpg编程常用

1. 同一级路径：图像文件位于html文件同一级，直接写图片名称（带后缀）
2. 下一级路径：图像文件位于html下一级，src=”下一级的目录名/名称（带后缀）
3. 上一级路径：图像位于文件上一级，src=”../名称（带后缀）

绝对路径：目录下的绝对位置，通常从盘符开始。如C:\User\XXX\XX 。编程不常用

超链接标签

<a href=”目标url” targrt=”目标窗口的弹出方式”> 文本或图像</a>

a为anchor的缩写，意为锚

1. **href**：用于指定链接目标url地址，必要属性，当为标签应用href属性时，其就有了超链接功能
2. target：用于指定链接页面的打开方式，其中\_self（默认值）在本窗口打开，\_blank为在新窗口打开

链接分类

1. 外部链接：必须以http://开头，如<a href=”http://www.baidu.com”> 百度</a>

2. 内部链接：网站内部页面之间的相互链接，<a href=”页面相对路径”> </a>

3. 空链接：<a href=”#”> </a>

4. 下载链接：若url地址指向的是一个文件或压缩包，点击后将下载文件。<a href=”文件相对路径”> </a>

5. 网页元素链接：在网页中的各种元素如文本、图像、表格、音频、视频等都可以添加超链接。<a href=”url”> <img src=”相对路径” /> </a>

6. 锚点链接：点击锚点链接可以快速定位到页面的某个位置。

1)将href属型设置为#名字的形式，如<a href=”#two”> 第二集</a>

2)找到目标位置标签，里面添加一个id属性=刚才的名字，如<h3 id=”two”>第二集介绍</a>

注释

以<!-- 开头，以-->结束。快捷键：ctrl+/

<!--注释语句-->

特殊字符

表格标签

表格主要用于显示、展示数据，可读性很好。

表格框架：

<table>

<tr>

<th> 表头</th>

<td> 内容</td>

...

</tr>

</table>

<table></table>定义表格的标签

<tr></tr>定义表格中行，必须嵌套在<table></table>中

<td></td>用于定义表格中的单元格，必须嵌套在<tr></tr>中

<th></th>表头单元格标签，用于第一行或第一列，表头单元格里的文本内容加粗居中显示

表格属性



属性键在<table>里

表格结构标签

<thead> </thead>表格头部，包含所有的表头<th>(整行)，必须包含一个<tr>

<tbody> </tbody>表格主体，剩下所有内容都可以放在里面

合并单元格

方式

1. 跨行合并：rowspan=”合并单元格的个数”，最上测单元格为目标单元格，写合并代码
2. 跨列合并：colspan=”合并单元格的个数”，最左侧单元格为目标单元格，写合并代码

步骤：

1. 先确定时跨行还是跨列合并
2. 找到目标单元格，写上合并代码eg：<td colspan=”4”> </td>
3. 删除多余单元格

列表标签

用来布局，分为无序列表，有序列表，自定义列表

无序列表（重点）：<ul> </ul>，列表项用<li> </li>标签定义

1. 无序列表的各个列表项之间没有顺序级别之分，是并列的
2. <ul>里只能放<li>标签
3. <li>里面可以放任何元素，相当于一个容器
4. 无序列表会带有自己的样式属性，但在实际使用时，会用CSS来设置

有序列表：<ol> </ol>，列表项用<li> </li>

1. <ol>里只能放<li>标签
2. 自带1.2.3样式，实际使用时用CSS设置

自定义列表（重点）：<dl> </dl> ，定义项目/名字用<dt> </dt>，描述项目/名字用<dd> </dd>

1. <dl>里面只能有<dt>和<dd>
2. <dt>和<dd>没有数量限制，通常是一个<dt>包含多个<dd>

结构为

<dl>

<dt>

<dd>XX</dd>

<dd>XX</dd>

...

</dt>

</dl>

表单标签

用来与用户交互收集信息，包含表单域、表单控件（也称表单元素）和提示信息三部分。

表单域：<form> </form>定义表单域，包含表单元素的区域，实现用户信息的的收集和传递。<form>会把它范围内的表单元素信息提交给服务器

结构：

<form action=”url地址” method=”提交方式” name=”表单域名称”>

各种表单元素控件

</form>



表单元素

<input type=”属性值” name=”” value=”” checked=”” maxlength=””/>



type可以为以下属性值



注：radio单选按钮需name属性相同才能体现单选

submit按钮需在form标签内才能生效

<label>：为input元素定义标注，用于绑定一个表单元素，当点击 <label>标签内的文本时，浏览器就会自动将焦点(光标)转到或选择对 应的表单元素上，用来增加用户体验

结构：

<label for=”sex”>男</label>

<input type=”radio” name=”nan” id=”sex” />

核心：<label>标签的for属性应当与相关元素的id属性相同

<select>下拉列表

1. <select>中至少包含一对<option>
2. <option>中定义selected=”selected”时，当前项即为默认选项

结构：

<select>

<option>选项1</option>

<option selected=”selected”>选项2</option>

...

</select>

<textarea>文本域标签：多行大量文本输入

1.cols=”每行中的字符数”，rows=”显示的行数”，实际中用CSS设置

结构：

<textarea rows=”3” cols=”20”>

内容

</textarea>

HTML5新特性

HTML5新增特性主要针对以前的不足，增加了一些新的标签、表单、表单属性等。这些新特性都有兼容性问题，IE9+以上版本才支持。

广义的HTML5包含HTML5、CSS3、JavaScript

HTML5新增语义化标签

<header>：头部标签

<nav>：导航标签

<article>：内容标签

<section>：定义文档某个区域

<aside>：侧边栏标签

<footer>：尾部标签



注：

这种语义化标准主要是针对搜索引擎的

新标签在页面中可以使用多次

在IE9中，需要把这些元素转换为块级元素

移动端多用这些标签

HTML5新增多媒体标签

使用该标签可以很方便的在页面中嵌入音频和视频，而不再去使用flash和其他浏览器插件

视频标签<video>

HTML5在不使用插件的情况下可以原生的支持视频格式文件，支持的格式有限



语法：

<video src=”文件路径” controls=”controls”></video>

兼容写法：

<video controls=”controls” width=”xxx”>

<source src=”movie.mp4” type=”video/mp4”>

<source src=”movie.ogg” type=”video/ogg”>

您的浏览器暂不支持<video>标签播放视频

</video>

由于代码执行有顺序性，所以会依次尝试执行，若成功执行则停止后续

<video>常见属性

autoplay：autoplay 视频就绪自动播放（谷歌需要搭配muted属性）

controls：controls 向用户显示播放控件

width：像素 设置播放器宽度

height：像素 设置播放器高度

loop：loop 是否循环播放

preload：auto|none 规定是否预加载视频，如果有了autoplay就忽略该属性

src：url 视频url地址

poster：imgurl 加载等待的画面图片

muted：muted 静音播放

音频标签<audio>

支持的格式



语法：

<audio src=”文件路径” controls=”controls”></audio>

兼容写法：

<audio controls=”controls”>

<source src=”happy.mp3” type=”audio/mpeg”>

<source src=”happy.ogg” type=”audio/ogg”>

您的浏览器暂不支持<audio>标签

</audio>

<audio>常见属性

autoplay：autoplay 设置自动播放音频（谷歌禁止自动播放音频，用JS）

controls：controls 显示控件

loop：loop 自动循环

src：url 音频的url

HTML5新增的input类型



HTML5新增的表单属性

required：required 表单拥有该属性表示其内容不能为空

placeholder：提示文本 表单提示信息，默认不显示

autofocus：autofocus 自动聚焦属性，页面加载完成自动聚焦到指定表单

autocomplete：off/on 用户开始键入时，浏览器基于之前键入过的值显示填 写过的值，默认打开，需要name属性且成功提交过

multiple：multiple 可以多选文件提交，type=”file”中的属性

CSS

CSS概述

CSS主要设置html页面内的文本内容，美化html，结构交给html，样式交给CSS

CSS主要部分构成：选择器+样式，位于html<head>标签，<style></style>内

<head>

<style>

选择器 {属性:值; 属性:值}

</style>

</head>

1. 选择器用于指定更改哪个html标签的CSS样式，花括号内是对该对象设置具体样式
2. 属性和属性值以键值对的形式出现
3. 属性是对指定的对象设置的样式属性，例如字体大小、文本颜色等
4. 属性和属性值之间用英文冒号隔开 :
5. 多个键值对之间用英文分号隔开 ；

**注：**

1. **修改完样式建议ctrl+F5强制刷新，否则可能出现无变化的情况**
2. **推荐使用百分确定容器大小和字体大小，这样可以更好的适配不同 分辨率的显示器**

CSS推荐代码风格：

1. 展开
2. 全部小写
3. 在属性值和冒号之间打一个空格，在选择器和大括号之间打一个空格

P {

color: red;

Font-size: 12px;

...

}

选择器（选择符）：选择html标签

基础选择器：由单个选择器成，包含标签选择器、类选择器、id选择器、通配符选择器

1. 标签选择器:选择页面中某一大类的全部标签

标签名 {

属性1: 属性值1;

属性2: 属性值2;

属性3: 属性值3;

...

}

1. 类选择器（常用）：实现差异化、单独、某些标签的样式更改

1>首先在<style>中定义类，英文句点+自定义类名

.类名 {

属性1: 属性值1;

属性2: 属性值2;

...

}

1. 在想更改样式的标签调用该类class=”类名”

<p class=”类名”>

内容

</p>

多类名调用：写在一个class即可，类名之间打一个空格

<p class=”类名 类名”

1. id选择器：与类部分相似，id属性只能在每个html文档里出现一次，只能调用一次
2. 首先在<style>中定义id样式，井号+自定义id名

#id名 {

属性1: 属性值1;

属性2: 属性值2;

...

}

1. 在想更改样式的标签调用该id，id=”id名”

<p id=”id名”>

内容

</p>

1. 通配符选择器：更改当前html文档全部标签的样式，只需在<style>中定义就能自动生效，不需要主动调用
2. 在<style>中定义通配符样式，单独一个\*

\* {

属性1: 属性值1;

属性2: 属性值2;

...

}

CSS字体：

font-family属性定义文字的字体

在<style>里配合选择器使用font-family: 字体1, 字体2, ...

p {

font-family: ”Mircosoft YaHei”, Arial, .... ;

}

1. 可以有多个字体，用英文逗号隔开，按顺序判断匹配，兼容性较好
2. 中间带空格的字体名用引号包住
3. 尽量使用系统自带的字体
4. 字体名尽量用英文，中文也行，但兼容性不如英文

font-size设置文字的大小，font-size: 20px; 配合选择器使用

1. px（像素），网页常用单位
2. Chrome文字默认16px
3. 可以给<body>指定整个页面文字的大小
4. 标题标签特殊，需要单独指定文字大小

font-weight设置文字的粗细，font-weight: normal/bold/bolder/lighter/数字

在<style>标签里配合选择器使用

1. 默认normal相当于400
2. Bold加粗，相当于700
3. 数字100—900

font-style设置文字的风格，font-style: normal/italic

在<style>里配合选择器使用

1. normal正常
2. italic倾斜

font复合属性，将文字属性整合到一条语句，减少代码量，属性设置有先后顺序，配合选择器使用。font: 属性值1 属性值2 属性值3 ...

1. font-style font-weight font-size font-family
2. 属性值之间用空格隔开
3. font-size和font-family不能省，否则font不生效，其他属性可以省略

CSS文本属性

定义文本的外观，如文本颜色、对齐文本、装饰文本、文本缩进、行间距等

color设置文本的颜色，配合选择器使用

color: 英文/#16进制颜色名/rgb(255,255,255);

text-align设置文本水平对齐，配合选择器使用

text-align: center/left/right;

text-decoration添加修饰，下划线、删除线、上划线等，配合选择器使用

text-decoration: underline/none/overline/line-through;

1. underline下划线
2. none无装饰
3. overline上划线
4. line-through删除线

text-indent设置首行缩进，通常设置段落首行缩进，配合选择器使用

text-indent: 20px/2em;

1. em是相对单位，1em相当于当前一个文字大小的距离
2. 缩进距离可以为负数

line-height设置行间距，行间距=上间距+文本高度+下间距，配合选择器使用

line-height: 20px;

当line-height=盒子高度时，效果为垂直居中（原理讲解视频P114）

CSS引入方式

按照CSS样式的书写位置可分为三大类

1. 行内样式表

在html标签内直接设定CSS样式style=”...”，适合简单修改

<p style=”color: pink; font-size: 20px; ...”>内容</p>

1. 只控制当前标签的样式
2. 权重高
3. 内部样式表

写到html页面内部，将所有CSS代码取出单独放到<style>标签中

<style>

选择器 {

属性1: 属性值1;

属性2: 属性值2;

...

}

</style>

1. <style>标签理论上可以放到html文档的任何地方，一般放到<head>里
2. 可控制当前整个页面中的元素样式设置
3. 代码结构清晰，但是没有实现结构与样式完全分离
4. 练习时常用，开发时不常用
5. 外部样式表（推荐）

样式单独写到CSS文件中，之后把CSS文件引入到html页面中使用

1>新建一个后缀名为.css的文件，把所有CSS代码放到里面

2>在html页面中使用<link>标签引入该CSS文件，一般放到<head>

<link rel=”stylesheet” href=”CSS文件路径”>

1. 可以控制多个页面的样式
2. 做到了结构样式完全分离

chrome调试工具

1. 左边为html标签，右边是CSS样式代码
2. 如果点击元素，发现右边没有样式引入，极有可能是类名或样式引入错误。
3. 如果有样式，但样式前面有黄色感叹号提示，则是样式属性书写错误
4. 如果有样式，样式前面是对勾，但整句被划掉，效果未生效，则该样式被同名样式覆盖掉了，即存在更高权重的同名样式

emmet语法

使用缩写来提高html/CSS的编写速度，vscode内部集成该语法

快速生成html结构语法

1. 生成标签直接输入标签名按tab补全
2. 生成多个相同标签加上\*即可，如div\*3可快速生成3个<div>
3. 生成父子级标签，用>，如ul>li
4. 生成兄弟级标签，用+，如div+p
5. 生成带有类或id的标签，用 . 和 # ，如p.red 生成<p class=”red”></p> p.#pink 生成<p id=”pink”></p>
6. 生成有顺序的带类标签，用$，如p.red$\*5 生成<p class=”red1”></p> <p class=”red2”></p> ...<p class=”red5”></p>
7. 生成带有文本的标签，用{ }，如p{hello world} 生成<p>hello world</p>

快速生成CSS样式语法

1. w200，按tab可以生成width: 200px;
2. lh26，按tab可以生成line-height: 26px;

快速格式化代码：

pycharm快捷键：Ctrl+Shift+L

vscode快捷键：Shift+Alt+F

vscode还能在settings里设置保存自动格式化代码 视频P96

复合选择器

1. 复合选择器可以更准确、更高效的选择目标元素
2. 复合选择器是由两个或多个基础选择器，通过不同的方式组合而成的
3. 包括：后代选择器、子选择器、并集选择器、伪类选择器等

后代选择器（重要）

又称包含选择器，可以选择父元素里面的子元素，写法为把父标签写在前面，子标签写在后面，中间用空格隔开

ol li {

属性1: 属性值1;

...

}

1. 元素1和元素2可以是父子关系，也可以是爷孙关系，嵌套层推荐写完整
2. 元素1和元素2可以是任意基础选择器
3. 后代选择器会选择所有匹配的后代标签，不分关系远近

子选择器（重要）

只选择作为某一元素的最近一级子元素，即选亲儿子元素

div>p {

属性1: 属性值1;

...

}

1. 元素1和元素2之间用大于号>隔开
2. 只选亲儿子元素2，孙子、重孙都不管

并集选择器（重要）

可以选择多组标签，同时定义相同的样式，通常用于集体声明，并集选择器是各选择器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

元素1, 元素2 {

属性1: 属性值1;

...

}

1. 并集选择器一般竖着写，最后一个选择器末尾不加逗号
2. 逗号理解为“and”的意思

伪类选择器（可用于所有元素，不限于a标签）

1. 用于向某些选择器添加特殊的效果，如给链接添加特殊效果，或选择第一个，第n个元素
2. 伪类选择器用冒号表示 : 如 :hover
3. 包含链接伪类、结构伪类等

a:link {

属性1: 属性值1;

...

}

链接伪类

a:link 选择所有未被访问的链接

a:visited 选择所有已被访问的链接

a:hover 选择鼠标指针位于其上的链接

a:active 选择活动链接（鼠标按下未弹起的链接）

1. 链接伪类按照顺序声明，LVHA：link visited hover active
2. 给链接指定样式需要单独声明，即使用链接伪类

focus伪类选择器

用于选取获得焦点的表单元素，焦点就是光标，通常<input>类表单元素才能获取，该选择器主要针对于表单元素

input:focus {

属性1: 属性值1;

...

}

CSS元素显示模式

即元素（标签）以什么方式进行显示，html元素一般分为块元素和行内元素

块元素有<h1>-<h6>、<p>、<div>、<ul>、<ol>、<li>等，<div>是最典型的块元素

块元素特点

1. 独占一行
2. 高度、宽度、外边距以及内边距可以设置
3. 宽度默认是容器（父级宽度）的100%
4. 是一个容器及盒子，里面可以放行内或者块级元素

注：文字类元素内不能放块级元素，如<p>、<h1>-<h6>

行内元素有<a>、<strong>、<b>、<em>、<i>、<del>、<s>、<ins>、<u>、<span>等，<span>是最典型的行内元素，行内元素也称内联元素

行内元素特点

1. 一行可显示多个
2. 高、宽直接设置无效
3. 默认宽度是它内容的宽度
4. 行内元素只能容纳文本或其他行内元素

注：链接里不能再放链接，链接里可以放块级元素，给链接<a>转换一下块级模式最安全

行内块元素

<img />、<input />、<td>同时具有块元素和行内元素的特点，有些资料称之为行内块元素

行内块元素特点

1. 和相邻行内（行内块）元素在一行上，之间有空白间隙，一行可以显示多个（行内元素特点）
2. 默认宽度就是它所含内容的宽度（行内元素特点）
3. 高度、行高、外边距以及内边距都可以控制（块级元素特点）

元素显示模式转换

一个模式的元素需要另外一种模式的特性，如想要增加链接<a>的出发范围，这时就需要元素显示模式转换

转换为块元素：display : block;

转换为行内元素：display : inline;

转换为行内块元素：display : inline-block;

CSS背景属性

可以设置背景颜色、背景图片、背景平铺、背景图片位置、背景图像固定等。

背景颜色

background-color: 颜色值; 颜色值默认为transparent（透明）

背景色半透明效果

1. rgba(0,0,0,X) X为0-1.0更改透明度
2. 16进制颜色名末尾额外加00-99调整透明度
3. CSS3新特性，IE9+才支持

背景图片

background-image: url(图片相对路径)

背景图片平铺

backgound-repeat: repeat/no-repeat/repeat-x/repeat-y

1. repeat：背景图在X和Y轴方向平铺（默认值）
2. no-repeat：背景图不平铺
3. repeat-x：背景图在X轴上平铺
4. repeat-y：背景图在Y轴上平铺

背景图片的位置

background-position: x y;

1. 可以写百分数
2. 可以写方位名词：top/center/bottom/left/right 顺序无先后，若指定了一个方位名词，另一个值省略，则省略的默认垂直居中
3. 可以写像素数，有顺序，先X后Y，若指定一个值，一定是X，Y默认垂直居中
4. 可以数值和方位名词混用，有顺序，先X后Y

背景图固定

设置背景图是否固定或随着页面其余部分滚动

background-attachment: scroll/fixed

背景属性复合写法

可以把背景属性写在一个background属性中，无特定顺序，一般为，中间空格隔开

background: 颜色、url、平铺、滚动、位置

扒背景图片

检查元素—>找到图片url—>右键open in tab

div相对位置

设置该div与同级div的相对位置，div父级定位用relative

postion: absolute;

left: 100px;

top: 100px;

bottom: 100px;

right: 100px;

CSS百分比大小

设定为当前元素是父元素的百分之X。

CSS三大特性：层叠性、继承性、优先级

层叠性

相同选择器给设置相同过的样式，此时一个样式就会覆盖（层叠）另一个冲突的样式。层叠性主要解决样式冲突的问题

层叠性原则

样式冲突，遵循就近原则，哪个样式离结构近（执行顺序靠后的），就执行哪个样式

样式不冲突不会层叠

继承性

子标签会继承父标签的某些样式，如文本颜色、字号。

子元素可以继承父元素的样式，text-、font-、line-这些元素开头的可以继承，以及color属性

行高继承

font: 12px/18px; /\* 带单位的写法 \*/

font: 12px/1.5; /\* 不加单位的写法 \*/

行高为1.5倍当前元素的字体大小，若父元素写1.5，子元素font-size: 14px;则子元素行高为14\*1.5=21px

行高可以跟单位也可以不跟

优先级

选择器权重



判断权重时从左到右比较数字大小，大的权重高

权重叠加

复合选择器的权重会叠加，如ul li 为0,0,0,1+0,0,0,1=0,0,0,2。

权重会叠加但是永远不会进位

底层逻辑是描述的越精确权重越高

盒子模型

看透网页布局的本质

1. 准备好相关网页元素，网页元素基本都是盒子box
2. 利用CSS设置好盒子样式，然后摆放到相应位置
3. 往盒子里装内容

盒子模型组成

border：边框

content：内容

margin：外边距，控制盒子与盒子的距离

padding：内边距，控制盒子边框与内容的距离



边框border

border可以设置元素的边框，边框由三部分组成：边框粗细、边框样式、边框颜色。语法：

border: border-width || border-style || border-color

border-width：定义边框粗细，px

border-style：边框样式

border-color：边框颜色

简写：border: 1px solid red; /\* 没有顺序\*/

边框可以分开写

border-top、border-bottom、border-left、border-right

边框具有层叠性（就近原则），先写border，再写border-top之类的可以覆盖指定边框

表格的细线边框

border-collapse属性控制浏览器绘制表格边框的方式，哈控制相邻单元格的边框

border-collapse: collapse;

collapse：合并，使得表格边框变细

边框会影响盒子大小

内边距padding

设置内容与边框的距离

padding-top、padding-bottom、padding-left、padding-right

简写：

4个值：padding: top right bottom left; /\* 顺时针，分别设置上、右、下、左\*/

3个值：padding: top right&left bottom; /\* 设置上，左右、下\*/

2个值：padding: top&bottom right&left; /\* 设置上下，左右\*/

1个值：padding: top&bottom&right&left; /\* 设置上下左右\*/

padding会影响盒子的实际大小

1. 元素具有width属性时，设置padding左右值会撑大盒子宽度
2. 元素具有heigth属性时，设置padding上下值会撑大盒子高度

利用该特性可以做不同字数的盒子具有相同的间距

padding不会影响盒子大小的情况

不给元素设置宽度属性（width）、高度属性（height），padding就不会影响盒子大小

外边距margin

设置盒子与盒子的距离

margin-top、margin-bottom、margin-left、margin-right

简写：与padding格式一致

外边距可以让块级盒子水平居中，但是必须满足两个条件

1. 盒子必须指定了width
2. margin的左右设置为auto

写法：margin: 0 auto; 或 auto;

行内元素或行内块元素水平居中

给其父元素添加text-align: center;

外边距合并问题

使用margin定义块元素的垂直外边距时，可能会出现外边距的合并

问题1：相邻块元素垂直外边距的合并

当上下相邻的两个块元素（兄弟）相遇时，如果上面的元素有margin-bottom，下面的元素有margin-top，则他们之间的垂直间距不是margin-bottom+margin-top，而是取二者中较大的值，这种现象称为相邻块元素垂直外边距的合并

解决方案：尽量只给一个盒子添加margin值

问题2：嵌套块元素初始外边距的塌陷

对于两个嵌套关系（父子关系）的块元素，父元素有上外边距同时子元素也有上外边距，此时父元素会塌陷较大的外边距值

解决方案：

1. 可以为父元素定义上边框，border-top: 1px solid transparent(透明);
2. 可以为父元素定义上内边距
3. 可以为父元素添加overflow: hidden;

清除自带内外边距（css样式第一句，清除默认布局）

网页很多元素都带有默认的内外边距，而且不同浏览器默认的不一样，因此在布局前，首先要清除网页元素自带的内外边距

语法：

\* {

padding: 0;

margin: 0;

}

注：行内元素为了照顾兼容性，尽量只设置左右内外边距，不要设置上下内外边距。但是转换为块级和行内块元素就可以了

清除li的小圆点

list-style: none;

CSS3新增特性（IE9及以上支持）

圆角边框、盒子阴影、文字阴影

圆角边框

border-radius: length;

radius半径（圆的半径）原理：（椭）圆与边框的交集形成圆角效果

1. 参数可以为数值或百分比
2. 若是正方形，设置为圆，把length设置为50%；若是矩形，设置左右为圆形，length设置为高度的一半
3. 该属性是一个简写属性，可以写四个值，左上、右上、右下、左下（顺时针）
4. 分开写：border-top-left-radius、border-bottom-right-radius以此类推

盒子阴影

box-shadow: h-shadow v-shadow blur spread color inset;

参数：

h-shadow：必需，水平阴影的位置，允许负值

v-shadow：必需，垂直阴影的位置，允许负值

blur：可选，模糊距离（虚实）

spread：可选，阴影尺寸（大小）

color：可选，阴影颜色，一般写rgba(0,0,0,0.3)

inset：可选，将外部阴影（outset默认）改为内部阴影

注：盒子阴影不占用空间

文字阴影

text-shadow: h-shadow v-shadow blur color;

参数：

h-shadow：必需，水平阴影的位置，允许负值

v-shadow：必需，垂直阴影的位置，允许负值

blur：可选，模糊的距离

color：可选，阴影的颜色

浮动

CSS三种布局：普通流、浮动、定位

实际开发中，一个页面基本包含这三种方式布局

标准流（普通流/文档流）

按照标签规定好的默认方式排列，是最基本的布局方式

1. 块级元素会独占一行，从上向下顺序排列

常用元素：div、hr、p、h1-h6、ul、ol、dl、form、table

1. 行内元素会按照顺序，从左到右顺序排列，碰到父元素边缘则自动换行

常用元素：span、a、i、em等

浮动最典型应用：可以让多个块级元素一行内排列显示

网页布局第一准则：多个块级元素纵向排列找标准流，多个块级元素横向排列找浮动

浮动float

float属性用于创建浮动框，将其移动到一边，直到左边缘或右边缘触及包含块或另一个浮动框的边缘

语法：

选择器 {

float: 属性值;

}

参数：

none：元素不浮动（默认）

left：元素向左浮动

right：元素向右浮动

浮动特性（重难点）

1. 浮动元素会脱离标准流（脱标）

浮动的盒子不再保留原先的位置，浮动元素不占空间

1. 浮动元素会一行内显示并且元素顶部对齐
2. 浮动的元素会具有行内块元素的特性

①任何元素都可以浮动，不管原先是什么模式的元素，添加浮动之后具有行内块元素相似的特性

②如果行内元素转换为float，可以直接给宽高属性

③如果块级盒子没有设置宽度，默认宽度和父级一样宽，但是添加浮动后，大小由内容决定

④浮动的盒子中间没有缝隙

1. 浮动元素经常和标准流父级搭配使用

标准流管上下，浮动管左右

网页布局第二准则：先设置盒子大小，再设置盒子的位置

常见网页布局







浮动布局注意点

浮动和标准流的父盒子搭配

一个元素浮动了，理论上其余的兄弟元素也要浮动

浮动的盒子指挥影响浮动盒子后面的标准流，不会影响前面的标准流

清除浮动

为什么要清除：

开发中会有不方便给父元素定义高度的时候（希望父元素随子元素的个数变化高度），但是不定义父元素高度会使父元素的高度塌陷为0，浮动元素不占空间导致父元素检测不到子元素，故高度为0。所以此时需要清除浮动所带来的对标准流的影响

语法：

选择器 {

clear: 属性值;

}

属性值：

left：不允许左侧有浮动元素（清除左侧浮动的影响）

right：不允许右侧有浮动元素（清除右侧浮动的影响）

both：同时清除左右两侧浮动的影响

实际中几乎只用clear: both;

清除浮动的策略是：闭合浮动

清除浮动方法：

1. 额外标签法

也称为隔墙法，W3C推荐

在浮动元素末尾添加一个空标签，如：<div style=”clear: both”></div>，或其他标签（如<br/>等）

优点：通俗易懂，书写方便

缺点：添加许多无意义的标签，结构化较差

要求：添加的空元素必须是块级元素

1. 父级添加overflow

给父级添加overflow，将属性值设置为hidden、auto、scroll都可以

1. :after伪元素法

该方法是额外标签法的升级版，也是给父元素添加

.clearfix:after {

content: “”;

display: block;

height: 0;

clear: both;

visibility: hidden;

}

.clearfix {

\*zoom: 1; /\* IE6、7专用 \*/

}

优点：没有增加标签，结构更简单

缺点：要照顾低版本浏览器

1. 双伪元素法

给父元素添加前后伪元素

.clearfix:before, .clearfix: after {

content: “”;

display: table;

}

.clearfix: after {

clear: both;

}

.clearfix {

\*zoom: 1;

}

优点：代码简洁

缺点：要照顾低版本浏览器

CSS属性书写顺序（重要）

遵循以下顺序：

1. 布局定位属性：display/position/float/clear/visibility/overflow
2. 自身属性：width/height/margin/padding/border/background
3. 文本属性：color/font/text-decoration/text-align/vertical-align/white-space/break-word
4. 其他属性（CSS3）：content/cursor/border-radius/box-shadow/text-shadow/background: linear-gradient

页面布局整体思路

1. 确定页面的版心（可视区），测量可得
2. 分析页面中的行模块。即页面布局第一准则
3. 分析行模块内的列模块，浮动布局，即页面布局第二准则
4. 制作HTML结构，遵循先有结构后有样式的原则，结构永远最重要
5. 先理清楚布局结构，再写代码

CSS定位position

定位：将盒子定在某一个位置，所以定位也是在摆放盒子，按照方位的方式移动盒子

定位=定位模式+边偏移

定位模式用于指定一个元素在文档中的定位方式。边偏移则决定了该元素的最终位置

定位模式

定位模式决定元素的定位方式，它通过CSS的position属性来设置，其值可以分为四个：

static：静态定位

relative：相对定位

absolute：绝对定位

fixed：固定定位

边偏移

边偏移就是定位的盒子移动到最终位置。有top、bottom。left、right4个属性



这四个属性只能移动position盒子，不能移动标准流盒子

静态定位static

静态定位是元素的默认定位方式，无定位的意思

选择器 {

position: static;

}

特点：

1. 静态定位按照标准流特性摆放位置，没有边偏移
2. 静态定位在布局时很少用到

相对定位relative（重点）

相对定位是元素在移动位置的时候，是相对于它原来的位置来说（左上角）

选择器 {

position: relative;

}

特点：

1. 相对于自己原来的位置来移动
2. 保留原位置不被替补，不脱标

绝对定位absolute（重点）

绝对定位是相对于元素的祖先元素来说的

选择器 {

position: absolute;

}

特点：

1. 若没有祖先元素或祖先元素没有定位(position)，则以浏览器为准定位
2. 若祖先元素有定位，则以最近一级的有定位的祖先元素作为参考系
3. 绝对定位不再占有原先的位置，脱标

子绝父相（重点）

子级使用绝对定位，父级则需要使用相对定位

1. 子级绝对定位，不会占有位置，可以放到父盒子里面的任何一个地方， 不会影响其他兄弟盒子
2. 父盒子需要加定位限制子盒子在父盒子内显示
3. 父盒子只能添加相对定位(relative)来使布局保持原样不混乱

总结：因为父级需要占有位置，一那次是相对定位，子盒子不需要占有位置，则是绝对定位

子绝父相不是永远不变，若父元素不需要占有位置，子绝父绝也会用到

固定定位fixed（重点）

固定定位是元素固定于浏览器可视区的位置，可以在浏览器页面滚动时元素位置不变

选择器 {

position: fixed;

}

特点：

1. 以浏览器的可视窗口为参照移动元素
2. 跟父元素没有任何关系
3. 不随滚动条滚动
4. 固定定位不占位置，脱标

粘性定位sticky

粘性定位可以被认为是相对定位和固定定位的混合

选择器 {

position: sticky;

}

特点

1. 以浏览器的可视窗口为参照点移动元素（fixed特点）
2. 粘性定位占有原先的位置（relative特点）
3. 必须添加top、left、bottom、right其中一个才生效
4. 兼容性差，IE不支持

定位总结



定位叠放次序z-index

z-index: xx; /\* xx为数字，数越大优先级越高 \*/

1. 数值可以是正整数，负整数或0，默认是auto

2. 若数值相同，后来者居上

3. 数字后不能加单位

4. 只有有定位的盒子才有z-index属性

绝对定位盒子的居中

加了绝对定位的盒子不能通过margin: 0 auto; 水平居中，可以通过小算法居中

1. left: 50% /\* 父级的一半 \*/
2. margin-left: 自身长度的一半;

注：如果一个盒子既有left属性又有right属性，默认只执行left；top、bottom 同理，默认只执行top

定位特殊特性

绝对定位和固定定位也和浮动类似

1. 行内元素添加绝对或者固定定位，可以直接设置高度和宽度
2. 块级元素添加绝对或者固定定位，如果不给宽度或者高度，默认大小是 内容的大小
3. 脱标的盒子（浮动元素、绝对定位、固定定位）不会触发外边距塌陷

定位对标准流的文字的影响

浮动元素只会压住下面的标准流的盒子，不会压住标准流盒子里的文字、图片

绝对定位、固定定位会压住下面标准流的所有内容

浮动之所以不会压住文字，因为浮动产生的目的最初是为了做文字环绕效果的，文字会围绕浮动元素

网页布局总结

通过盒子模型，清楚知道html大部分标签是一个盒子

通过css浮动、定位可以让每个盒子排列成为网页

一个完整的网页是标准流、浮动、定位一起完成布局的，每个都有自己的专门用法

1. 标准流

可以让盒子上下或者左右排列，垂直的块级盒子显示就用标准流布局

1. 浮动

可以让多个块级元素一行显示或者左右对齐盒子，多个块级盒子水平显 示就用浮动

1. 定位

定位最大的特点是有层叠的概念，可以让多个盒子前后叠压来显示。如 果元素自由在某个盒子内移动就用定位布局

元素的显示和隐藏

点击关闭不见，刷新页面会重新出现

本质：让一个元素在页面中隐藏或者显示出来

display、visibility、overflow

display显示隐藏（重点）

display: none; /\* 隐藏对象 \*/

display: block /\* 除了转换为块级元素外，还有显示元素的意思 \*/

display隐藏元素后不再占有原来的位置

应用极其广泛，搭配js可以做很多的网页特效

visibility可见性

visibility: visible; 元素可视

visibility: hidden; 元素不可视

visibility隐藏元素后继续占有原位置

overflow溢出

overflow属性指定了如果内容溢出元素的框时，会发生什么

overflow: visible; 溢出可见

overflow: hidden; 溢出隐藏

overflow: scroll; 永久添加滚动条

overflow: auto; 需要时添加滚动条

注：有定位的盒子慎用overflow: hidden; 它会隐藏超出的部分

CSS高级技巧

精灵技术

为什么需要精灵图：

一个网页中往往会应用很多小的背景图像作为修饰，图片过多时服务器就会 频繁的接收和发送请求图片，这会造成服务器请求压力过大，大大降低页面 的加载速度，因此使用精灵图可以减少请求次数，优化加载速度

精灵图的使用

1. 精灵技术主要针对于背景图片的使用，就是把多个小背景图片整合到一 张大图片中，父图也叫精灵图，sprite图
2. 移动背景图片的位置，background-position: x y;
3. 移动的距离就是这个目标图片的x和y坐标

字体图标

字体图标使用场景：主要用于显示网页中通用、常用的一些小图标

精灵图缺点：

1. 图片文件比较大
2. 图片本身放大缩小会失真
3. 一旦图片制作完，更换很麻烦

字体图标iconfont可以提供一种方便高效的图标使用方式，展示的是图标， 本质属于文字

字体图标优点

1. 轻量级：字体图标比图像要小，优化加载速度
2. 灵活性：本质是文字，可以在CSS内更改颜色、大小、各种效果等
3. 兼容性：几乎所有浏览器都支持

注：字体图标做不到复杂效果，故字体图标不能替代精灵技术

字体图标的使用

下载→引入→使用→追加

下载：

icomoon字库：<http://icomoon.io> 外国网站，免费（操作视频P256）

阿里iconfont字库：<http://www.iconfont.cn/> 免费

不同浏览器所识别的文件不同



引入：

1. 把下载包里面的fonts文件夹放到页面根目录下
2. 将style.css内的声明代码复制到页面的<style>标签内（详见视频P257）



使用：

1. 点开demo.html复制图标对应的小方块，或者复制前面的编码，并在前面添加反斜杠转义，例如：\e91e
2. 给小方块设置font-family: icomoon; 该处以icomoon为例，字体名字可更 改，在字体声明代码内更改

追加：

1. 在网站里点import icons按钮，选择selection.json
2. 添加新icon
3. 点击generate font按钮生成新的字体图标文件
4. 点donwload下载到本地

CSS三角

网页中一些常见的三角形，使用css直接画出来就可以，不比做成图片或字体图标

div {

width: 0;

height: 0;

line-height: 0; /\* line-height和font-size为了兼容性 \*/

font-size: 0;

border: 50px solid transparent;

border-left-color: pink;

}

CSS用户界面样式

界面样式就是用户操作样式，以便提供更好的用户体验

鼠标样式cursor

default：默认，白鼠标

pointer：小手

move：移动

text：文本

not-allowed：禁止

轮廓线outline

给表单添加outline: 0; 或 none; 就可以去掉默认的蓝色边框

文本域防拖拽resize

resize: none;

vertical-align

用于设置图片或者表单（行内块元素）和文字垂直对齐

写在图片样式中

vertical-align: baseline|top|middle|bottom

baseline：默认，元素放置在父元素的基线上

top：把元素的顶端与行中最高元素的顶端对齐

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



解决图片底部默认空白缝隙问题

图片底部会有一个空白缝隙，原因是行内块元素默认和文字的基线对齐

解决方法：

1. 给图片添加vertical-align: top|bottom|middle; 中的任意一种（推荐）
2. 把图片转换为块级元素display: block;

溢出的文字用省略号代替

单行情况：

1. 强制一行内显示文本

white-space: nowrap;

1. 超出的部分隐藏

overflow: hidden;

1. 文字超出部分用省略号代替

text-overflow: ellipsis;

多行情况：

该情况存在兼容性问题，适用于webkit浏览器或移动端

overflow: hidden;

text-overflow: ellipsis;

display: -webkit-box; /\* 弹性伸缩盒子模型显示\*/

-webkit-line-clamp: 2; /\* 限制在一个快元素显示的文本的行数\*/

-webkit-box-orient: vertical; /\* 设置或检索伸缩盒对象的子元素排列方式\*/

布局技巧

margin负值的应用

margin-left: -1px; /\* 值为border的宽度\*/

可以抵消多个盒子边框叠加的bug

鼠标经过盒子边框变化

鼠标经过某个盒子时，提高当前盒子的层级即可（若没定位则加relative，有定位则加z-index）

文字围绕应用

将带文字的盒子挤到右边（视频P269）

行内块巧妙运用

用来做分页模块（视频P270）

CSS三角巧妙运用

制作梯形盒子（视频P271）

CSS初始化

不同浏览器的初始样式是不一样的，需要统一消除；并且有些兼容性设置需要导入（视频P275）

CSS3新特性

有兼容性问题，IE9+才支持

移动端支持优于PC端

主要学习新增选择器、盒子模型、其他特性

CSS3新增选择器

1. 属性选择器
2. 结构伪类选择器
3. 伪元素选择器

属性选择器

可以根据元素特定属性来选择元素，这样可以不借助于类或者id选择器



注：val可以不加引号，att=val，类选择器、属性选择器、伪类选择器权重为 10（0,0,1,0）

结构伪类选择器

结构伪类选择器主要根据文档结构来选择元素，常用于根据父级选择器里面的子元素



nth-child(n)详解

选择某个父元素的一个或多个特定的子元素

n可以是数字，关键字和公式

n：是数字

就是选择第n个子元素，里面数字从1开始

n：是关键字

even：偶数关键字

odd：奇数关键字

n：是公式

n：从0开始，累加1,0和溢出的自动忽略，相当于选择所有孩子



注：nth-child()会把所有孩子排序，执行时先看nth-child()所选择的序号，再看前面的选择条件，若无符合的则会不执行。

nth-of-type(n)详解

大致与nth-child用法相同

不同点在于，nth-of-type会先看指定元素的类型并排号，再根据n选择指定的元素

伪元素选择器（重点）

伪元素选择器利用CSS创建新标签元素，而不需要HTML标签，从而简化HTML结构

::before、::after（CSS3写法）

:before、:after（CSS2写法，兼容）

::before

在元素内部的前面插入内容

::after

在元素内部的后面插入内容

注：

1. before和after创建一个元素，但是属于行内元素
2. 新创建的元素在文档树中没有，所以叫伪元素
3. 语法：element::before{ }
4. before和after必须有content属性
5. before在父元素内容的前面创建元素，after在父元素内容的后面插入元素
6. 伪元素选择器和标签选择器一样，权重为1

伪元素法清楚浮动详解



CSS3盒子模型

CSS3中可以通过box-sizing来指定盒模型，有两个值：content-box、border-box这样计算盒子大小的方式就发生了改变

语法：

1. box-sizing: content-box; 盒子大小为width+padding+border（旧版本默认）
2. box-sizing: border-box; 盒子大小为width

若盒子模型改为了border-box，那padding和border就不会撑大盒子，前提是padding和border小于盒子的width

CSS3其他特性

filter滤镜：

filter属性将模糊或颜色偏移等图形效果应用于元素

语法：

filter: 函数();

filter: blur(5px); blur模糊处理，数值越大越模糊

calc函数：

calc()能在声明css属性值时执行一些计算

width: calc(100% - 80px); /\*减号前后加空格\*/

括号里可以使用+、-、\*、/来进行计算

CSS过渡transition（重点）

语法：

transition: 要过渡的属性 花费时间 运动曲线 何时开始;

谁做过渡给谁加

超过一个属性变化的写法：

transition: width .1s, height .1s; 使用逗号隔开

属性：想要变化的css属性，宽度高度、背景颜色、内外边距等。如果想要所有属性都过渡变化，写all

花费时间：单位是秒（必须写单位）eg：0.5s

运动曲线：默认是ease（可省略）

何时开始：单位是秒（必须写单位）可以设置延迟触发时间，默认是0s（可省略）

运动曲线详解



2D转换transform

转换（transform）是CSS3中具有颠覆性的特征之一，可以实现元素的位移、旋转、缩放等效果，可以简单理解为变形

translate：移动

rotate：旋转

scale：缩放

移动translate

可以用来做鼠标悬停时上跳效果

语法：

transform: translate(x,y);

transform: translateX(n);

transform: translateY(n);

重点：

1. 定义2D转换中的移动，沿着X和Y轴元素，上移负值，左移负值
2. translate最大的优点：不会影响到其他元素的位置
3. translate中的百分比单位是相对于自身元素的translate:(50%, 50%);
4. 对行内标签没效果

旋转rotate

2D旋转指的是让元素在二维平面内顺时针旋转或者逆时针旋转

语法：

transform: rotate(度数);

重点：

1. 度数单位是deg，rotate(45deg)
2. 角度为正时，顺时针，负时，为逆时针
3. 默认旋转的中心点是元素的中心点

rotate制作三角

将正方形盒子旋转45度并隐藏两个边框即可得到三角

转换中心点transform-origin

设置元素转换的中心点

transform-origin: x y;

重点：

1. 参数用空格隔开
2. x y默认中心点是元素的中心点（50% 50%）
3. 还可以给x y设置像素或者方位名词（top bottom left right center）

缩放scale

设置该属性控制元素放大缩小

语法：

transform: scale(x,y);

注意：

1. 参数之间用空格隔开，没有单位，即倍数
2. transform: scale(1,1)：宽和高倍数为1，即无放大效果
3. transform: scale(2,2)：宽和高倍数为2，即放大至2倍
4. 只写一个参数scale(2)：相当于scale(2,2)，等比例缩放
5. scale(0.5,0.5)：缩小为0.5倍
6. 可以设置转换中心点缩放，默认以中心点缩放，不影响其他盒子
7. 添加overflow: hidden;可以做成放大内容不放大盒子的效果

transform综合写法

同时使用多个转换，格式为：

transform: translate() rotate() scale() 等

注：

1. 顺序会影响转换效果（先旋转会改变坐标轴方向）
2. 当同时有位移translate和其他属性的时候，要将translate放在最前面

CSS3动画animation

动画（animation）是CSS3中具有颠覆性的特征之一，可通过设置多个节点来精确控制一个或一组动画，常用来实现复杂的动画效果

相比较过渡，动画可以实现更多变化，更过控制，连续自动播放等效果

制作动画分为两步：

1. 定义动画
2. 调用动画

keyframes定义动画

@keyframes 动画名称 {

0%{

属性: 值;

}

100%{

属性: 值;

}

}

动画序列：

1. 0%是动画的开始，100%是动画的完成，这样的规则就是动画序列
2. 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
3. 动画是使元素从一种样式逐渐变化为另一种样式的效果，可以改变任意 多的样式任意多的次数
4. 用百分比来规定变化发生的时间，或用关键词“from”和“to”，等同于 0%和100%

元素使用动画

语法：

选择器 {

animation-name: 动画名称; /\* 调用动画 \*/

animation-duration: 持续时间; /\* 持续时间，单位s \*/

}

常见动画属性



速度曲线细节

animation-timing-function：规定动画的速度曲线，默认是“ease”



动画简写

animation: 名称 持续时间 运动曲线 何时开始 播放次数 正反方向 起始 或结束的状态

注：简写里不包含animation-play-state

单个元素可以添加多个动画

flex布局

display: flex;

传统布局：

兼容性好，布局繁琐，局限性，不能在移动端很好的布局

flex弹性布局：

操作方便，pc端浏览器支持情况较差，IE11或更低版本，不支持或仅部分支持

pc端还是用传统布局

移动端或者不考虑兼容性问题，用flex弹性布局

布局原理

任何一个容器元素都可以指定为flex布局

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

采用flex布局的元素，称为flex容器，它的所有子元素自动成为容器的成员，称为flex项目

常见父元素属性

flex-direction：设置主轴的方向

justify-content：设置主轴上的子元素排列方式

flex-wrap：设置子元素是否换行

align-content：设置侧轴上的子元素排列方式（多行）

align-items：设置侧轴上的子元素排列方式（单行）

flex-flow：复合属性，相当于同事设置了flex-direction和flex-wrap

flex布局中是分为主轴和侧轴两个方向，同样的叫法有：行和列、x轴和y轴

默认主轴方向就是x轴方向，水平向右

默认侧轴方向就是y轴方向，水平向下

flex-direction属性决定主轴的方向（即项目的排列方向）

注：主轴和侧轴是会变化的，就看flex-direction设置谁为主轴，剩下的就是侧轴，而子元素是跟着主轴排列的

row：默认值，从左到右

row-revrese：从右到左

column：从上到下

column-reverse：从下到上

justify-content设置主轴上的子元素排列方式

注：使用该属性前要确定好主轴是哪个

flex-start：默认值，从头部开始，如果主轴是x轴，则从左到右

flex-end从尾部开始排列

center：在主轴居中对齐（如果主轴是x轴则水平居中）

space-around：平分剩余空间

space-between：先两边贴边再平分剩余空间（重要）

flex-wrap设置子元素是否换行

默认情况下，项目都排在一条线（轴线）上。flex-wrap定义元素是否换行

nowrap：默认值，不换行

wrap：换行

align-items设置侧轴上得到子元素排列（单行）

该属性是控制子项在侧轴（默认是y轴）上的排列方式，在子项为单项的时候使用

flex-start：默认值，从上到下

flex-end：从下到上

center：挤在一起居中（垂直居中）

stretch：拉伸

align-content设置侧轴上的子元素排列方式（多行）

只能用于子项出现换行的情况（多行），在单行下是没有效果的

flex-start：默认值，在侧轴的头部开始排列

flex-end：在侧轴的尾部开始排列

center：在侧轴中间显示

space-around：子项在侧轴平分剩余空间

space-between：子项在侧轴先分布在两头，再平分剩余空间

stretch：设置子元素高度平分父元素高度

flex-flow

flex-flow属性是flex-direction和flex-wrap属性的复合属性

语法：flex-flow: row wrap;

flex布局子项常见属性

flex

该属性定义子项目分配剩余空间，用flex来表示占多少份数

.item {

flex: x; // x为数字，表示该子元素占多少份，默认为0

}

align-self控制子项自己在侧轴上的排列方式

align-self属性允许单个项目有与其他项目不一样的对其方式，可覆盖align-items属性。默认是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

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

注：和z-index不一样

