

前端

前端即[网站前台](#)部分，运行在[PC端](#)，[移动端](#)等[浏览器](#)上展现给用户浏览的网页和界面。

前端开发中最基本也是最必须的三个技能。前端的开发中，在页面的布局时，

- HTML：页面结构：可以把他看成一个文档，定义展示页面的内容结构。
- CSS：页面表现：元素大小、颜色、位置、隐藏或显示、部分动画效果。
- JavaScript：页面行为：部分动画效果、页面与用户的交互、页面功能



一、HTML

HTML: 超文本标记语言，是网页制作必备的编程语言“[超文本](#)”就是指页面内可以包含图片、[链接](#)，甚至音乐、[程序](#)等非文字元素。

HTML 历史版本

- **HTML 2.0**：是 1996 年由 Internet 工程工作小组的 HTML 工作组开发的。
- **HTML 3.2**：作为 W3C 标准发布于 1997 年 1 月 14 日。HTML 3.2 向 HTML 2.0 标准添加了被广泛运用的特性，诸如字体、表格、applets、围绕图像的文本流，上标和下标。
- **HTML 4.0**：被发布于 1997 年 12 月 18 日。而仅仅进行了一些编辑修正的第二个版本发布于 1998 年 4 月 24 日。HTML 4.0 最重要的特性是引入了样式表（CSS）。
- **HTML 4.01**：发布于 1999 年 12 月 24 日。
- **XHTML 1.0**：发布于 2000 年 1 月 20 日，使用 XML 对 HTML 4.01 进行了重新地表示
- **HTML5**：目前最流行的html版本

1、HTML页面结构

超文本标记语言的结构包括“头”部分（**Head**）、和“主体”部分（**Body**），其中“头”部提供关于网页的信息，“主体”部分提供网页的[具体内容](#)

```
1  <!DOCTYPE html>
2  <html lang="zh-CN">
3  <head>
4      <!-- 头部信息 不会再页面显示-->
5      <meta charset="utf-8">
6      <title></title>
7  </head>
8  <body>
9      <!-- 主体部分 页面中显示内容-->
10     <h1>第一个网页页面</h1>
11 </body>
12 </html>
```

1、文档声明

DOCTYPE声明该html文件使用的HTML版本

HTML 5

```
1  <!DOCTYPE html>
```

HTML 4.01 Strict

```
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
```

2、页面头部

第二行“”标签和最后一行“”定义html文档的整体，“”标签中的lang="en"定义网页的语言为英文，定义成中文是lang="zh-CN"。“”标签和“”标签是它的第一层子元素，

<head>:标签里面负责对网页进行一些设置以及定义标题，设置包括定义网页的编码格式，外链css样式文件和javascript文件等，设置的内容不会显示在网页上，标题的内容会显示在标题栏。

3、页面内容

<body>:元素定义文档的主体，也就是页面显示的内容。

body 元素包含文档的所有内容（比如文本、超链接、图像、表格和列表等等），。

2、常见的HTML标签

1、注释:

定义和用法

注释标签用于在源代码中插入注释。注释不会显示在浏览器中。

您可使用注释对您的代码进行解释，这样做有助于您在以后的时间对代码的编辑。当您编写了大量代码时尤其有用。

例:

```
1  <!-- 这是一段注释。注释不会在浏览器中显示。-->
2
3  <p>这是一段普通的段落。</p>
```

2、标题标签:

定义和用法

-

标签可定义标。

定义最大的标题。

定义最小的标题。

由于 h 元素拥有确切的语义，因此请您慎重地选择恰当的标签层级来构建文档的结构。因此，请不要利用标题标签来改变同一行中的字体大小

```
1 <h1>这是标题 H1</h1>
2 <h2>这是标题 H2</h2>
3 <h3>这是标题 H3</h3>
4 <h4>这是标题 H4</h4>
5 <h5>这是标题 H5</h5>
6 <h6>这是标题 H6</h6>
```

3、段落和换行标签

定义和用法

- **p**：标签定义段落。元素会自动在其前后创建一些空白。浏览器会自动添加这些空间，您也可以在样式表中规定。

```
1 <p>元素会自动在其前后创建一些空白。浏览器会自动添加这些空间，您也可以在样式表中规定。</p>
```

- **br**：插入一个简单的换行符。标签是空标签，意味着它没有结束标签，

错误的用法：

- 1 | `<p>`元素会自动在其前后创建一些空白。`
`浏览器会自动添加这些空间，您也可以在样式表中规定。
`</p>`
- **hr：标签在 HTML 页面中创建一条水平线。水平分隔线（horizontal rule）可以在视觉上将文档分隔成各个部分，在 HTML 中，**
-
- 标签没有结束标签。**

1 | `<hr>`

4、块标签

- **div标签**: 标签块元素，表示一块内容，没有具体的语义。
div标签可以把文档分割为独立的、不同的部分。它可以用作严格的组织工具，并且不使用任何格式与其关联。
div中的部分会显示为红色

```
1 | <div style="color:red">
2 |   <h5>这是标题 H5</h5>
3 |   <h6>这是标题 H6</h6>
4 | </div>
```

- **span标签** 行内元素，表示一行中的一小段内容，没有具体的语义。
pan 没有固定的格式表现。当对它应用样式时，它才会产生视觉上的变化，

```
1 | <p>
2 |   元素会自动在其前后创建一些空白。浏览器会自动添加这些空间
3 |   <span style="color: cyan">span中的内容</span>
4 |   您也可以在样式表中规定。
5 | </p>
```

5、含样式和语义的行内标签

标签	含义
	行内元素，字体斜体。
	行内元素，语义为强调内容，表示重要（倾斜效果）
	行内元素，字体加粗
	行内元素，语义为强调内容，表示非常重要（效果加粗）

```
1 <em>这个是em标签</em>
2 <i>这个是i标签</i>
3 <strong>这个是strong</strong>
4 <b>这个是b标签</b>
```

6、图像标签和链接标签

- **img标签**

img 元素向网页中嵌入一幅图像。

请注意，从技术上讲， 标签并不会在网页中插入图像，而是从网页上链接图像。 标签创建的是被引用图像的占位空间。

标签有两个必需的属性：src 属性 和 alt 属性。

```
1 
```

属性	描述
alt	规定图像的替代文本。
src	规定显示图像的 URL。
height	定义图像的高度。
width	设置图像的宽度

- **a标签**：标签定义超链接，用于从一张页面链接到另一张页面

元素最重要的属性是 href 属性，它指示链接的目标。

```
1 <a href="http://www.baidu.com">点击链接跳转到百度</a>
```

- **link标签**(放在头部中)

链接到一个外部样式，即链接外部的css文件

7、音频标签&视频标签

- **audio**

```
1 <audio controls="controls">
2 </audio>
```

- **video**

```
1 <video width="320" height="240" controls="controls">
2 </video>
```

8、列表

- **有序列表**

在网页上定义一个有编号的内容列表可以用

1. 配合使用来实现，代码如下：

```

1 <ol>
2   <li>列表文字一</li>
3   <li>列表文字二</li>
4   <li>列表文字三</li>
5 </ol>

```

在网页上生成的列表，每条项目上会按1、2、3编号，有序列表在实际开发中较少使用。

- **无序列表**

在网页上定义一个无编号的内容列表可以用

- 、
- 配合使用来实现，代码如下：

```

1 <ul>
2   <li><a href="#">标题一</a></li>
3   <li><a href="#">标题二</a></li>
4   <li><a href="#">标题三</a></li>
5 </ul>

```

在网页上生成的列表，每条项目上会有一个小图标。

- **定义列表**

dl:标签定义了定义列表（definition list）。标签用于结合dt定义列表中的项目和dd描述列表中的项目

```

1 <dl>
2   <dt>python</dt>
3   <dd>解释性语言</dd>
4   <dt>java</dt>
5   <dd>编译型语言</dd>
6 </dl>

```

9、表格

- **table**：table标签定义 HTML 表格。

简单的 HTML 表格由 table 元素以及一个或多个 tr、th 或 td 元素组成。

tr 元素定义表格行，th 元素定义表头，td 元素定义表格单元。

```

1 <table border="1" >
2   <tr>
3     <th>name</th>
4     <th>java</th>
5     <th>python</th>
6   </tr>
7   <tr>
8     <td>小明</td>
9     <td>精通</td>
10    <td>熟悉</td>
11  </tr>
12  <tr>
13    <td>小张</td>
14    <td>不会</td>
15    <td>不会</td>
16  </tr>
17 </table>

```

3、HTML表单

1、form标签

- **form**: 标签用于为用户输入创建 HTML 表单，表单能够包含 [input 元素](#)，比如文本字段、复选框、单选框、提交按钮等等。

form 标签的属性：

属性	描述
action	定义表单数据提交地址
method	定义表单提交的方式（常见的有get和post）

- **form 表单中包含的元素**

元素标签

作用

为表单元素定义文字标注

定义通用的表单元素

标签

标签

标签</td><td>定

1<form>

2 账号: <input type="text" name="username" id="user" />

3

4 密码: <input type="password" name="password" id="pw" />

5 </form>

type属性

值	作用
text	定义单行文本输入框
password	定义密码输入框
radio	定义单选框
checkbox	定义复选框
file	定义上传文件
submit	定义提交按钮
button	定义一个普通按钮
reset	定义重置按钮
image	定义图片作为提交按钮, 用src属性定义图片地址
hidden	定义一个隐藏的表单域, 用来存储值

3、label标签

标签为 input 元素定义标注（标记）。

label 元素不会向用户呈现任何特殊效果。不过，它为鼠标用户改进了可用性。如果您在 label 元素内点击文本，就会触发此控件。就是说，当用户选择该标签时，浏览器就会自动将焦点转到和标签相关的表单控件上。

标签的 for 属性应当与相关元素的 id 属性相同

属性	对应值的描述
for	for 属性应当与相关元素的 id 属性相同

案例:

1 <form>

2 <label for="user">账号:</label>

3 <input type="text" name="username" id="user" />

4

5 <label for="pw">密码:</label>

6 <input type="password" name="password" id="pw" />

7 </form>

4、textarea标签

textarea标签定义多行的文本输入控件。

文本区中可容纳无限数量的文本，其中的文本的默认字体是等宽字体（通常是 Courier）。

可以通过 cols 和 rows 属性来规定 textarea 的尺寸，更好的办法是使用 CSS 的 height 和 width 属性。

1 <form action="l1l1l" method="get" id='fo'>

2 <textarea name="text"></textarea>

3 <input type="submit" name="" value="提交" >

4 </form>

textarea的属性

属性	值	描述
autofocus	autofocus	规定在页面加载后文本区域自动获得焦点。
cols	number	规定文本区内的可见宽度。
disabled	disabled	规定禁用该文本区。
form	form_id	规定文本区域所属的一个或多个表单。
maxlength	number	规定文本区域的最大字符数。
name	name_of_textarea	规定文本区的名称。
placeholder	text	规定描述文本区域预期值的简短提示。
readonly	readonly	规定文本区为只读。
required	required	规定文本区域是必填的。
rows	number	规定文本区内的可见行数。

5、select标签

select 元素可创建单选或多选菜单,也可以用于选择数据提交表单。

1<form>

2<select name='skill'>

3 <option value ="py">python</option>

4 <option value ="ht">html</option>

5 <option value ="css">css</option>

6 <option value ="java">javascript</option>

7</select>

8<input type="submit" name="">

9</form>

option 元素定义下拉列表中的一个选项（一个条目）。

浏览器将
标签中的内容作为 标签的菜单或是滚动列表中的一个元素显示。option 元素位于 select 元素内部。4、内联框架 1、iframeiframe 元素会创建包含另外一个文档的内联框架（即行内框架）。例子：将登录页面的内容链接到当前页面

```
1 | <iframe src="http://02form.html" height="200" width="400" >
```

二、CSS

1. css介绍和引用

1.1. CSS 概述

CSS：层叠样式表(英文名称: Cascading Style Sheets)是一种用来表现HTML或XML（标准通用标记语言的一个子集）等文件样式的计算机语言。CSS不仅可以静态地修饰网页，还可以配合各种脚本本语言动态地对网页各元素进行格式化，CSS 能够对网页中元素位置的排版进行像素级精确控制，支持几乎所有的字体字号样式，拥有对网页对象和模型样式编辑的能力

1.2. css的基本语法

CSS 规则由两个主要的部分构成：选择器，以及一条或多条声明

选择器通常是您需要改变样式的 HTML 元素。

每条声明由一个属性和一个值组成。

属性（property）是您希望设置的样式属性（style attribute）。每个属性有一个值。属性和值被冒号分开。



1.3. css引入方法:

1、外联式：通过link标签，链接到外部样式表到页面中。

```
1 | <link rel="stylesheet" type="text/css" href="css/main.css">
```

2、嵌入式：通过style标签，在网页上创建嵌入的样式表。

```
1 | <style type="text/css">
2 |   div{ width:100px; height:100px; color:red }
3 |   .....
4 | </style>
```

3、内联式：通过标签的style属性，在标签上直接写样式。

```
1 | <div style="width:100px; height:100px; color:red ">.....</div>
```

2. css背景

2.1. 设置背景色

background-color 设置背景色。这个属性接受任何合法的颜色值。

这条规则把元素的背景设置为红色：

```
1 | p {background-color: red;}
```

2.2. 设置背景图像

background-image: 要把图像放入背景，属性的默认值是 none，表示背景上没有放置任何图像。如果需要设置一个背景图像，必须为这个属性设置一个 URL 值：

```
1 | body {background-image: url(/i/eg_bg_04.gif);}
```

大多数背景都应用到 body 元素，不过并不仅限于此。

▪ 背景图平铺

如果需要在页面对背景图像进行平铺，可以使用 [background-repeat 属性](#)。

属性值：

- repeat 导致图像在水平垂直方向上都平铺，就像以往背景图像的通常做法一样
- repeat-x：图像只在水平上重复，
- repeat-y：图像在垂直方向上重复
- no-repeat：不允许图像在任何方向上平铺。

默认地，背景图像将从一个元素的左上角开始。请看下面的例子：

```
1 | body
2 | {
3 |   background-image: url(/i/eg_bg_03.gif);
4 |   background-repeat: repeat-y;
5 | }
```

2.3. 背景定位

可以利用 [background-position 属性](#)改变图像在背景中的位置。

定位方式：位置关键字、百分比。

下面的例子在 body 元素中将一个背景图像居中放置：

▪ 位置关键字:

```
1 | body
2 | {
3 |   background-image:url('/i/eg_bg_03.gif');
4 |   background-repeat:no-repeat;
5 |   background-position:center;
6 | }
```

单一关键字	等价的关键字
top	top center 或 center top
bottom	bottom center 或 center bottom
center	center center
right	right center 或 center right
left	left center 或 center left

▪ 百分数值

百分数值的表现方式更为复杂。假设你希望用百分数值将图像在其元素中居中，这很容易：

元素标签	作用
<pre>1 body 2 { 3 background-image:url('/i/eg_bg_03.gif'); 4 background-repeat:no-repeat; 5 background-position:50% 50%; 6 }</pre>	
<div><div>▪ 长度值</div><p>长度值解释的是元素内边距区左上角的偏移。偏移点是图像的左上角。</p><p>比如，如果设置值为 50px 100px，图像的左上角将在元素内边距区左上角向右 50 像素、向下 100 像素的位置上：</p><pre>1 body 2 { 3 background-image:url('/i/eg_bg_03.gif'); 4 background-repeat:no-repeat; 5 background-position:50px 100px; 6 }</pre></div>	
<div><div>2.4、背景关联</div><p>如果文档比较长，那么当文档向下滚动时，背景图像也会随之滚动，当文档滚动到超过图像的位置时，图像就会消失。</p><p>background-attachment 属性：声明图像相对于可视区是固定的，因此不会受到滚动的影响：</p><pre>1 body 2 { 3 background-image:url('/i/eg_bg_02.gif'); 4 background-repeat:no-repeat; 5 background-attachment:fixed 6 }</pre></div>	
<div><div>2.5、背景大小</div><pre>1 # 通过像素指定 2 background-size: 100px 20px; 3 # 通过百分比指定 4 background-size: 10% 20%; 5 # 原始图片大小 6 background-size:auto; 7 # 背景填满容器 8 background-size:cover;</pre></div>	
<div><div>3、颜色表示法</div><p>颜色三种表示方法：</p><ul style="list-style-type: none">▪ 颜色名表示，比如：red，cyan gold▪ rgb表示，比如：rgb(255,0,0)表示红色▪ 16进制数值表示，比如：#FFFF00 表示黄色<p>颜色对照表地址：http://tool.oschina.net/commons?type=3</p><p>🔖1553653444168</p></div>	
<div><div>4、css文本样式</div><p>常用的应用文本的css样式：</p><ul style="list-style-type: none">▪ color 设置文字的颜色，<ul style="list-style-type: none">▪ color:red;▪ font-size 设置文字的大小<ul style="list-style-type: none">▪ font-size:12px;▪ font-family 设置文字的字体<ul style="list-style-type: none">▪ font-family:'微软雅黑';▪ font-style 设置字体是否倾斜：<ul style="list-style-type: none">▪ 不倾斜: font-style:'normal';▪ 倾斜: font-style:'italic';▪ font-weight 设置文字是否加粗，<ul style="list-style-type: none">▪ font-weight:bold 设置加粗▪ font-weight:normal 设置不加粗▪ line-height 设置文字的行高 (行高相当于在每行文字的上下同时加间距)，<ul style="list-style-type: none">▪ line-height:30px;▪ 设置文字的几个属性，顺序如下： font: 是否加粗 字号 /行高 字体；<ul style="list-style-type: none">▪ font:normal 12px/36px '微软雅黑';▪ text-indent 设置文字首行缩进<ul style="list-style-type: none">▪ text-indent:24px; 设置文字首行缩进24px▪ text-align 设置文字水平对齐方式<ul style="list-style-type: none">▪ 如text-align:center 设置文字水平居中</div>	
<div><div>5、css选择器</div><div><div>5.1、标签选择器</div><p>最常见的 CSS 选择器是元素选择器。换句话说，文档的元素就是最基本的选择器。</p><p>如果设置 HTML 的样式，选择器通常将是某个 HTML 元素，比如 p、h1、em、a，甚至可以是 html 本身：</p><pre>1 <h1>h1标题</h1> 2 <h2>h2标题</h2> 3 4 h1 {color:blue;} 5 h2 {color:silver;}</pre></div></div>	
<div><div>5.2、id选择器</div><p>首先，ID 选择器前面有一个 # 号</p><p>通过id名来选择元素，元素的id名称不能重复，所以一个样式设置项只能对应于页面上一个元素，一个ID只能在文档中使用一次，id名一般给程序使用，所以不推荐在css设置样式时使用id作为选择器</p><pre>1 <input type="text" name="uname" id="user"> 2 #user {font-weight:bold;}</pre></div>	
<div><div>5.3、类选择器</div><p>必须将 class 指定为一个适当的值，类名前有一个点号 (.)，通过class类名来选择元素，一个类可应用于多个元素，一个元素上也可以使用多个类，应用灵活，可复用，是css中应用最多的一种选择器。</p><pre>1 <div class="menu"></div> 2 3 .menu {color:red;}</pre></div>	
<div><div>5.4、属性选择器</div></div>	

```
1 <a href="" name="nb"></a>
2 <a href=""></a>
3 <a href=""></a>
4
5 a[name] {color:red;}
```

5.5、包含选择器（层级选择器）

主要应用在选择父元素下的子元素，或者子元素下面的子元素，可与标签元素结合使用，减少命名，同时也可以通过层级，防止命名冲突。

```
1 <div class="menu">
2   <div>
3     <p>p标签</p>
4   </div>
5 </div>
6
7 .menu div p{color:cyan}
```

5.6、组选择器

多个选择器，，每个选择器之间用逗号隔开如果有同样的样式设置，可以使用组选择器。

```
1 h1, h2, h3 {color:blue;}
```

5.7、伪类及伪元素选择器

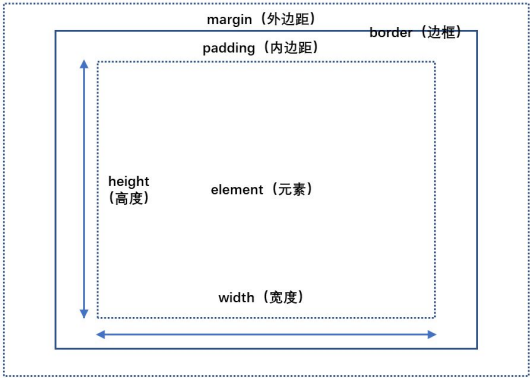
伪类

在支持 CSS 的浏览器中，链接的不同状态都可以不同的方式显示，这些状态包括：活动状态，已被访问状态，未被访问状态，和鼠标悬停状态。

属性	描述	CSS
:active	向被激活的元素添加样式。	1
:focus	向拥有键盘输入焦点的元素添加样式。	2
:hover	当鼠标悬停在元素上方时，向元素添加样式。	1
:link	向未被访问的链接添加样式。	1
:visited	向已被访问的链接添加样式。	1

6、css模型框

CSS 框模型 (Box Model) 规定了元素框处理元素内容、内边距、边框 和 外边距 的方式。



元素框的最内部分是实际的内容，直接包围内容的是内边距。内边距呈现了元素的背景。内边距的边缘是边框。边框以外是外边距，外边距默认是透明的，因此不会遮挡其后的任何元素。

1、边框

元素的边框 (border) 是围绕元素内容和内边距的一条或多条线。border 属性允许你规定元素边框的样式、宽度和颜色。

border-style 的默认值是 none，如果没有声明样式，就相当于 border-style: none，因此，如果您希望边框出现，就必须声明一个边框样式

CSS 边框属性

属性	描述
border	简写属性，用于把针对四个边的属性设置在一个声明。
border-style	用于设置元素所有边框的样式，或者单独地为各边设置边框样式。
border-width	简写属性，用于为元素的所有边框设置宽度，或者单独地为各边边框设置宽度。
border-color	简写属性，设置元素的所有边框中可见部分的颜色，或为 4 个边分别设置颜色。

除了以上四种，还可以对四个方向单独设种子

border的样式类型

Color #3E80F7
Font 17px Lato, Serif
Margin 30px 10px 10px 0px
Padding 15px 35px
Contrast 3.21 < 4.5

```
1 solid # 实线
2 dotted # 点线
3 dashed # 线
4 double # 双线
```

2、内边距

padding 属性定义元素边框与元素内容之间的空白区域。

CSS 内边距属性

属性	描述
padding	简写属性。作用是在一个声明中设置元素的所内边距属性。
padding-bottom	设置元素的下内边距。
padding-left	设置元素的左内边距。
padding-right	设置元素的右内边距。
padding-top	设置元素的上内边距。

设置的时候可以通过像素，也可以按元素的百分比进行设置。

元素标签	作用												
<p>h1 元素的各边都有 10 像素的内边距，只需要这样：</p> <pre>1 h1 {padding: 10px;}</pre> <p>也可以给四个方向设置不同值，这些值的顺序是从上外边距 (top) 开始围着元素顺时针旋转的：</p> <pre>1 margin: top right bottom left</pre> <p>另外</p> <p>把段落的内边距设置为父元素 width 的 10%：</p> <pre>1 p {padding: 10%;}</pre>													
<p>3、外边距</p> <p>设置外边距的最简单的方法就是使用 margin 属性，这个属性接受任何长度单位、百分数值甚至负值。</p> <p>外边距属性</p> <table><tr><th>属性</th><th>描述</th></tr><tr><td>margin</td><td>简写属性。在一个声明中设置所有外边距属性。</td></tr><tr><td>margin-bottom</td><td>设置元素的下外边距。</td></tr><tr><td>margin-left</td><td>设置元素的左外边距。</td></tr><tr><td>margin-right</td><td>设置元素的右外边距。</td></tr><tr><td>margin-top</td><td>设置元素的上外边距</td></tr></table> <p>▪ 超出范围内容隐藏</p> <pre>1 overflow: hidden;</pre>	属性	描述	margin	简写属性。在一个声明中设置所有外边距属性。	margin-bottom	设置元素的下外边距。	margin-left	设置元素的左外边距。	margin-right	设置元素的右外边距。	margin-top	设置元素的上外边距	
属性	描述												
margin	简写属性。在一个声明中设置所有外边距属性。												
margin-bottom	设置元素的下外边距。												
margin-left	设置元素的左外边距。												
margin-right	设置元素的右外边距。												
margin-top	设置元素的上外边距												
<p>7、浮动和定位</p> <p>1、CSS 定位机制</p> <p>CSS 有三种基本的定位机制：普通流、浮动和绝对定位。</p> <p>除非专门指定，否则所有框都在普通流中定位。也就是说，普通流中的元素的位置由元素在 (X)HTML 中的位置决定。</p> <p>块级框从上到下一个接一个地排列，框之间的垂直距离是由框的垂直外边距计算出来。</p> <p>通过使用 position 属性，我们可以选择不同类型的定位。</p> <p>2、相对定位</p> <p>注意，在使用相对定位时，无论是否进行移动，元素仍然占据原来的空间。因此，移动元素会导致它覆盖其它框。</p> <p>如果将 top 设置为 20px，那么框将在原位置顶部下面 20 像素的地方。如果 left 设置为 30 像素，那么会在元素左边创建 30 像素的空间，也就是将元素向右移动。</p> <pre>1 #box_relative { 2 position: relative; 3 left: 30px; 4 top: 20px; 5 }</pre>													
<p>3、绝对定位</p> <p>绝对定位使元素的位置与文档流无关，因此不占据空间。这一点与相对定位不同，相对定位实际上被看作普通流定位模型的一部分，因为元素的位置相对于它在普通流中的位置。</p> <p>因为绝对定位的框与文档流无关，所以它们可以覆盖页面上的其它元素。可以通过设置 z-index 属性来控制这些框的堆放次序。</p> <pre>1 #box { 2 position: absolute; 3 left: 30px; 4 top: 20px; 5 }</pre>													
<p>4、固定位置</p> <p>▪ 需要设置position:fixed;，直接以浏览器窗口作为参考进行定位，它是浮动在页面中，元素位置不会随浏览器窗口的滚动条滚动而变化</p>													
<p>5、浮动</p> <p>浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样。</p> <p>在 CSS 中，我们通过 float 属性实现元素的浮动。</p> <pre>1 img{ 2 float:right; 3 }</pre> <table><tr><th>值</th><th>描述</th></tr><tr><td>left</td><td>元素向左浮动。</td></tr><tr><td>right</td><td>元素向右浮动。</td></tr><tr><td>none</td><td>默认值。元素不浮动，并会显示在其在文本中出现的位置。</td></tr><tr><td>inherit</td><td>规定应该从父元素继承 float 属性的值。</td></tr></table>	值	描述	left	元素向左浮动。	right	元素向右浮动。	none	默认值。元素不浮动，并会显示在其在文本中出现的位置。	inherit	规定应该从父元素继承 float 属性的值。			
值	描述												
left	元素向左浮动。												
right	元素向右浮动。												
none	默认值。元素不浮动，并会显示在其在文本中出现的位置。												
inherit	规定应该从父元素继承 float 属性的值。												
<p>8、display属性</p> <p>display属性是用来设置元素的类型及隐藏的，常用的属性有：</p> <p>▪ none 元素隐藏</p> <p>▪ block 元素以块元素显示</p>													
<p>三、javascript</p>													
<p>1、介绍和基本使用</p>													
<p>1.1、javascript概述</p> <p>▪ JavaScript历史</p>													

元素标签	作用
<p>要了解JavaScript，我们首先要回顾一下JavaScript的诞生。在上个世纪的1995年，当时的网景公司正凭借其Navigator浏览器成为Web时代开启时最著名的第一代互联网公司。由于网景公司希望能在静态HTML页面上添加一些动态效果，于是叫Brendan Eich这哥们在两周之内设计出了JavaScript语言。你没看错，这哥们只用了10天时间。为什么起名叫JavaScript？原因是当时Java语言非常红火，所以网景公司希望借Java的名气来推广，但实际上JavaScript除了语法上有点像Java，其他部分基本上没啥关系。</p>	
<h3>1.2、嵌入页面的方式</h3>	
<p>HTML 中的脚本必须位于 标签之间。脚本可被放置在 HTML 页面的 和 部分中。</p>	
<p>1、行内事件（主要用于事件）</p>	
<pre>1 <input type="button" name="" onclick="alert('ok! ');"></pre>	
<p>2、页面script标签嵌入</p>	
<pre>1 <script type="text/javascript"> 2 alert('ok! '); 3 </script></pre>	
<p>3、外部引入</p>	
<pre>1 <script type="text/javascript" src="js/index.js"></script></pre>	
<h2>2、基本语法</h2>	
<h3>1、注释</h3>	
<p>我们可以添加注释来对 JavaScript 进行解释，或者提高代码的可读性。</p>	
<ul style="list-style-type: none">单行注释：以 // 开头。多行注释：以 /* 开始，以 */ 结尾。	
<h3>2、变量</h3>	
<ul style="list-style-type: none">变量命名：<ul style="list-style-type: none">变量可以使用短名称（比如 x 和 y），也可以使用描述性更好的名称（比如 age, sum, totalvolume）。变量必须以字母开头变量也能以 \$ 和 _ 符号开头（不过我们不推荐这么做）变量名区分大小写（y 和 Y 是不同的变量）驼峰式命名（小驼峰）变量声明	
<p>JavaScript中变量可以先声明后赋值。</p>	
<pre>1 var x; 2 x = 20;</pre>	
<p>也可以在声明的时候赋值。</p>	
<pre>1 var y=3;</pre>	
<p>同时给多个变量赋值</p>	
<pre>1 var a = 100, b = 200, c = 300;</pre>	
<h3>3、数据类型：</h3>	
<p>字符串、数字、布尔、Null、Undefined、数组</p>	
<ul style="list-style-type: none">number 数字类型	
<pre>1 var y=3;</pre>	
<ul style="list-style-type: none">string 字符串类型	
<pre>1 var st='python'</pre>	
<ul style="list-style-type: none">boolean 布尔类型<ul style="list-style-type: none">只有 true 或 false 两个值	
<ul style="list-style-type: none">undefined 类型,<ul style="list-style-type: none">变量已声明 未赋值，它的值就是undefined	
<pre>1 var unde</pre>	
<ul style="list-style-type: none">null 类型,<ul style="list-style-type: none">表示空对象，类似于python中的None	
<pre>1 var tr = null</pre>	
<ul style="list-style-type: none">arry 数组<ul style="list-style-type: none">数值类似于python中的列表，可以通过下标取值	
<pre>1 //创建数组 2 var skill= Array("html","css","js"); 3 alter(skill[0]);</pre>	
<ul style="list-style-type: none">length属性：获取数组长度：	
<pre>1 alter(skill.length)</pre>	
<ul style="list-style-type: none">pop() 方法从数组中删除最后一个元素	
<pre>1 skill.pop()</pre>	
<ul style="list-style-type: none">push() 方法（在数组结尾处）向数组添加一个新的元素	
<pre>1 skill.push('jquery')</pre>	
<h3>4、运算符</h3>	
<ul style="list-style-type: none">算术运算符：+(加)、-(减)、*(乘)、/(除)、%(求余)	
<ul style="list-style-type: none">赋值运算符：=、+=、-=、*=、/=、%=、++	
<ul style="list-style-type: none">条件运算符：? , > , >= , < , <= , != ,	
<ul style="list-style-type: none">逻辑运算符：&&(与)、 (或)、!(否)	

5. 条件语句 元素标签	作用
<p>通过条件来控制程序的走向，就需要用到条件语句。</p> <ul style="list-style-type: none">▪ if 语句 <pre>1 var a = 199; 2 if(a>200){ 3 alert('大于200'); 4 }</pre> <ul style="list-style-type: none">▪ else 语句 <pre>1 var a = 199; 2 if(a>200){ 3 alert('大于200'); 4 } 5 else { 6 alert(a); 7 }</pre> <ul style="list-style-type: none">▪ else if 语句<ul style="list-style-type: none">▪ 类似python中的 elif <pre>1 var a = 199; 2 if(a>200){ 3 alert('大于200'); 4 } 5 else if(a>100){ 6 alert('大于100') 7 } 8 else { 9 alert(a); 10 }</pre>	
<p>6、函数</p> <ul style="list-style-type: none">▪ 函数定义：<ul style="list-style-type: none">▪ 定义函数的关键词 function，函数内的代码块，包裹在花括号中： <pre>1 // 函数的定义 使用function 2 function func() { 3 alter(999) 4 } 5 // 函数调用 6 func()</pre> <ul style="list-style-type: none">▪ 提示：JavaScript 对大小写敏感。关键词 function 必须是小写的 <ul style="list-style-type: none">▪ 函数参数： <pre>1 function addNumber(var1,var2) { 2 alert(var1 + var2); 3 } 4 addNumber(11,22)</pre> <ul style="list-style-type: none">▪ 函数返回值：<ul style="list-style-type: none">▪ 使用 return 来返回值。在使用 return 语句时，函数会停止执行，并返回指定的值。 <pre>1 function addNumber(var1, var2) { 2 return var1 + var2; 3 } 4 var res = addNumber(11,222); 5 alert(res)</pre>	
<p>7、对象</p> <ul style="list-style-type: none">▪ 创建对象 <p>对象由花括号分隔。在括号内部，</p> <ul style="list-style-type: none">▪ 方式一： <pre>1 objA = new Object();</pre> <ul style="list-style-type: none">▪ 方式二：类式于python中的字典， <pre>1 var objA={};</pre> <ul style="list-style-type: none">▪ 对象属性<ul style="list-style-type: none">▪ 对象的属性以名称和值对的形式 (name : value) 来定义。属性由逗号分隔： <pre>1 var objA={name : "python",age : 18,id :123};</pre> <ul style="list-style-type: none">▪ 创建对象属性声明可横跨多行： <pre>1 var objA={ 2 name : "python", 3 age : 18, 4 id :123 5 };</pre> <ul style="list-style-type: none">▪ 对象方法<ul style="list-style-type: none">▪ 对象方法是作为属性来存储的函数。 <pre>1 var objC = { // 创建对象：方式3 2 name: 'abc', 3 age: 19, 4 fun: function (var1,var2) { 5 alert(var1); 6 alert(var2); 7 alert(this.age * 2); 8 } 9 }; 10 objC.fun(111,222)</pre>	
<p>8、循环</p>	

元素标签	作用
<h2>2. 获取标签注意问题</h2> <p>的getElementById方法来获取页面上设置了id属性的元素，</p> <pre>1 <script type="text/javascript"> 2 var bo = document.getElementById('box'); 3 //box对应的div这个元素定义在下面，此时获取不到，就会出错 4 </script> 5 6 <div id="box">这是div元素</div></pre> <p>上面的语句，如果把javascript写在元素的上边，就会出错，因为页面上从上往下加载执行的，javascript去页面上获取元素div1的时候，元素div1还没有加载。</p> <p>解决方法:</p> <ul style="list-style-type: none">■ javascript放到页面最下边 <pre>1 <div id="box">这是div元素</div> 2 3 <script type="text/javascript"> 4 var bo = document.getElementById('box'); 5 </script> 6 </body></pre> <ul style="list-style-type: none">■ 放入window.onload触发的函数里面 <p>window.onload函数在页面加载完后才执行，就不会出错了。</p> <pre>1 <script type="text/javascript"> 2 window.onload = function(){ 3 var bo = document.getElementById('box'); 4 } 5 </script> 6 7 <div id="box">这是div元素</div></pre>	
<h2>2. 操作标签内容</h2> <ul style="list-style-type: none">■ innerHTML 属性 <ul style="list-style-type: none">■ 获取元素内容的最简单方法是使用 innerHTML 属性。innerHTML 属性对于获取或替换 HTML 元素的内容很有用。 <ul style="list-style-type: none">■ 读取节点文本 <pre>1 var odv1 = document.getElementById('box'); 2 var content1 = odv1.innerHTML; 3 alert(content1)</pre> <ul style="list-style-type: none">■ 写入节点文本 <pre>1 var odv1 = document.getElementById('box'); 2 odv1.innerHTML = '哈哈'; 3</pre>	
<h2>3. 修改标签属性</h2> <ul style="list-style-type: none">■ 操作属性的方法 <ul style="list-style-type: none">■ 1、“.”操作■ 2、“[]”操作 <ul style="list-style-type: none">■ 读取属性 <pre>1 <input type="text" id="input_user" name="user" value="python"> 2 // 读取属性 3 var odv1 = document.getElementById('input_user'); 4 var idvalue = odv1.id; 5 var namevalue = odv1['name']; 6 alert(idvalue); 7 alert(namevalue);</pre> <ul style="list-style-type: none">■ 修改属性 <pre>1 柠檬班 2 3 // 修改属性 4 a1 = document.getElementById('a1'); 5 // a1.href = 'http://www.lemonban.com'; 6 a1['href'] = 'http://www.lemonban.com';</pre> <ul style="list-style-type: none">■ 注意点：操作class属性的时候，要将属性名字改为className <pre>1 a1.className = 'hied';</pre> <ul style="list-style-type: none">■ 修改css样式 <ul style="list-style-type: none">■ 方式一：节点.style.样式名■ 方式二：节点.style[‘样式名’] <pre>1 <div class="dv1" id="dv01" >这个是div1</div> 2 3 //修改css样式 4 var od 5</pre>	
<h2>4. onclick事件</h2> <ul style="list-style-type: none">■ onclick事件：点击了该 HTML 元素■ 事件属性添加 JavaScript 代码： <pre>1 // 给div定义一个onclick事件，触发该事件的时候，调用该事件指定的函数 2 <div class="dv1" id="dv01" onclick="tank()">这个是div1</div> 3 4 // 定义一个函数，负责弹框 5 function tank() { 6 alert('hello') 7 }</pre>	

1 元素标签	作用
<h2>四、jquery</h2>	
<h3>1、jquery介绍和引用</h3>	
<p>jQuery是目前使用最广泛的javascript函数库。</p> <p>jQuery的版本分为1.x系列和2.x、3.x系列，1.x系列兼容低版本的浏览器，2.x、3.x系列放弃支持低版本浏览器，目前使用最多的是1.x系列的。</p> <p>jquery的口号和愿望 Write Less, Do More （写得少，做得多）</p> <ul style="list-style-type: none">▪ 官方网站: http://jquery.com/▪ 在线手册: https://www.runoob.com/manual/jquery/▪ 下载<ul style="list-style-type: none">▪ 版本下载: https://code.jquery.com/▪ 引入页面<ul style="list-style-type: none">▪ jquery是一个函数库，一个js文件，页面用script标签引入这个js文件就可以使用。	
<pre>1 <script type="text/javascript" src="js/jquery-1.12.4.min.js"></script></pre>	
<h3>2、jquery选择器</h3>	
<ul style="list-style-type: none">▪ jquery用法思想— 选择某个网页元素，然后对它进行某种操作▪ jquery基本的选择器	
<p>jquery选择器可以快速地选择元素，选择规则和css样式相同，使用length属性判断是否选择成功。</p>	
<pre>1 // id选择器 2 \$(' #btn1') 3 // 类选择器 4 \$(' .box1') 5 // 标签选择器 6 \$('li') 7 // 层级选择器 8 \$(' #ul li span') 9 // 属性选择器 1 \$('input[name=user]')</pre>	
<ul style="list-style-type: none">▪ 选择同胞和父辈元素	
<pre>1 //选择div元素前面紧邻的同胞元素 2 \$('div').prev(); 3 //选择div元素之前所有的同胞元素 4 \$('div').prevAll(); 5 6 //选择div元素后面紧邻的同胞元素 7 \$('div').next(); 8 //选择div元素后面所有的同胞元素 9 \$('div').nextAll(); 1 10 //选择div的父元素 11 \$('div').parent(); 12 //选择div的所有子元素 13 \$('div').children(); 14 //选择div的同级元素 15 \$('div').siblings(); 16 17 //选择div内的class等于cs的元素 18 \$('div').find('#cs li');</pre>	
<ul style="list-style-type: none">▪ 选择过滤	
<pre>1 // 选择包含p元素的div元素 2 \$('div').has('p'); 3 4 //选择class不等于cs的div元素 5 \$('div').not('.cs'); 6 7 //选择class等于cs的div元素 8 \$('div').filter('.cs'); 9 1 //选择第6个div元素 0 \$('div').eq(5);</pre>	
<ul style="list-style-type: none">▪ 获取元素的索引值	
<p>有时候需要获得匹配元素相对于其同胞元素的索引位置，此时可以用index()方法获取</p>	
<pre>1 var \$li = \$('li li').eq(1); 2 alert(\$li.index()); // 弹出1 3 4 <ul class="list"> 5 1 6 2 7 4 8 5 9 6 1 </pre>	
<h3>3、jquery操作样式</h3>	
<ul style="list-style-type: none">▪ 获取元素样式	
<pre>1 // 获取div的样式 2 \$('div').css("width"); 3 \$('div").css("color");</pre>	
<ul style="list-style-type: none">▪ 修改元素样式	

元素标签	作用																						
<pre>1 //设置div的样式 2 \$('div').css("width", "30px"); 3 \$('div').css("height", "30px"); 4 \$('div').css({fontSize:"30px",color:"red"});</pre> <div>▪ 添加或者移除class属性</div> <pre>1 // 添加class属性 2 \$('#div1').addClass("cs2") 3 // 移除class属性 4 \$('#div1').removeClass("cs2") 5 6 // 重复切换样式 7 \$('#div1').toggleClass("cs2")</pre>																							
<h4>4、绑定click事件</h4> <p>给元素绑定click事件，可以用如下方法：</p> <pre>1 \$('#btn1').click(function(){ 2 3 // 内部的this指的是原生对象 4 5 // 使用jquery对象用 \$(this) 6 7 })</pre> <div>▪ jquery链式调用：</div> <p>jquery对象的方法会在执行完后返回这个jquery对象，所有jquery对象的方法可以连起来写</p> <pre>1 \$(this).next().stop().slideToggle().parent().siblings().children('ul').slideUp();</pre> <div>▪ 案例1：层级菜单制作</div> <div>▪ 案例2：弹出框的制作</div>																							
<h4>5、jquery操作属性</h4> <table><tr><th>方法</th><th>描述</th></tr><tr><td>attr()</td><td>设置或返回匹配元素的属性和值。</td></tr><tr><td>removeAttr()</td><td>从所有匹配的元素中移除指定的属性。</td></tr></table>	方法	描述	attr()	设置或返回匹配元素的属性和值。	removeAttr()	从所有匹配的元素中移除指定的属性。																	
方法	描述																						
attr()	设置或返回匹配元素的属性和值。																						
removeAttr()	从所有匹配的元素中移除指定的属性。																						
<h4>6、jquery操作元素</h4> <div>▪ 获取文本内容：</div> <ul style="list-style-type: none">text() - 设置或返回所选元素的文本内容html() - 设置或返回所选元素的内容（包括 HTML 标记）val() - 设置或返回表单字段的值 <div>▪ 添加新的 HTML 内容</div> <ul style="list-style-type: none">append() - 在被选元素的结尾插入内容prepend() - 在被选元素的开头插入内容after() - 在被选元素之后插入内容before() - 在被选元素之前插入内容 <div>▪ 删除元素/内容</div> <ul style="list-style-type: none">remove() - 删除被选元素（及其子元素）empty() - 从被选元素中删除子元素（清空元素中的内容） <div>▪ 案例3：动态插入选项框</div>																							
<h4>7、jquery效果函数</h4> <table><tr><th>方法</th><th>描述</th></tr><tr><td>animate()</td><td>对被选元素应用“自定义”的动画</td></tr><tr><td>fadeIn()</td><td>逐渐改变被选元素的不透明度，从隐藏到可见</td></tr><tr><td>fadeOut()</td><td>逐渐改变被选元素的不透明度，从可见到隐藏</td></tr><tr><td>hide()</td><td>隐藏被选的元素</td></tr><tr><td>show()</td><td>显示被选的元素</td></tr><tr><td>slideDown()</td><td>通过调整高度来滑动显示被选元素</td></tr><tr><td>slideToggle()</td><td>对被选元素进行滑动隐藏和滑动显示的切换</td></tr><tr><td>slideUp()</td><td>通过调整高度来滑动隐藏被选元素</td></tr><tr><td>stop()</td><td>停止在被选元素上运行动画</td></tr><tr><td>toggle()</td><td>对被选元素进行隐藏和显示的切换</td></tr></table>	方法	描述	animate()	对被选元素应用“自定义”的动画	fadeIn()	逐渐改变被选元素的不透明度，从隐藏到可见	fadeOut()	逐渐改变被选元素的不透明度，从可见到隐藏	hide()	隐藏被选的元素	show()	显示被选的元素	slideDown()	通过调整高度来滑动显示被选元素	slideToggle()	对被选元素进行滑动隐藏和滑动显示的切换	slideUp()	通过调整高度来滑动隐藏被选元素	stop()	停止在被选元素上运行动画	toggle()	对被选元素进行隐藏和显示的切换	
方法	描述																						
animate()	对被选元素应用“自定义”的动画																						
fadeIn()	逐渐改变被选元素的不透明度，从隐藏到可见																						
fadeOut()	逐渐改变被选元素的不透明度，从可见到隐藏																						
hide()	隐藏被选的元素																						
show()	显示被选的元素																						
slideDown()	通过调整高度来滑动显示被选元素																						
slideToggle()	对被选元素进行滑动隐藏和滑动显示的切换																						
slideUp()	通过调整高度来滑动隐藏被选元素																						
stop()	停止在被选元素上运行动画																						
toggle()	对被选元素进行隐藏和显示的切换																						
<h4>8、jquery事件</h4> <pre>1 2 click() // 鼠标单击 3 change() // 元素改变 4 mouseover() // 鼠标进入（进入子元素也触发） 5 mouseout() // 鼠标离开（离开子元素也触发） 6 mouseenter() // 鼠标进入（进入子元素不触发） 7 mouseleave() // 鼠标离开（离开子元素不触发） 8 hover() // 同时为mouseenter和mouseleave事件指定处理函数 9 ready() // DOM加载完成 10 resize() // 浏览器窗口的大小发生变化 11 scroll() // 滚动条的位置发生变化 12 submit() // 用户递交表单</pre>																							
<h4>9、ajax</h4> <p>ajax技术的目的是让javascript发送http请求，与后台通信，获取数据和信息。ajax技术的原理是实例化xmlhttp对象，使用此对象与后台通信。ajax通信的过程不会影响后续javascript的执行，从而实现异步。</p> <p>同步和异步 现实生活中，同步指的是同时做几件事情，异步指的是做完一件事后再做另外一件事，程序中的同步和异步是把现实生活中的概念对调，也就是程序中的异步指的是现实生活中的同步，程序中的同步指的是现实生活中的异步。</p>																							

元素标签	作用
<p>局部刷新和无刷新 ajax可以实现局部刷新，也叫做无刷新，无刷新指的是整个页面不刷新，只是局部刷新，ajax可以自己发送http请求，不用通过浏览器的地址栏，所以页面整体不会刷新，ajax获取到后台数据，更新页面显示数据的部分，就做到了页面局部刷新。</p> <p>同源策略 ajax请求的页面或资源只能是同一个域下面的资源，不能是其他域的资源，这是在设计ajax时基于安全的考虑。特征报错提示：</p> <p>\$ajax使用方法 常用参数：1、url 请求地址 2、type 请求方式，默认是'GET'，常用的还有'POST' 3、dataType 设置返回的数据格式，常用的是'json'格式，也可以设置为'html' 4、data 设置发送给服务器的数据 5、success 设置请求成功后的回调函数 6、error 设置请求失败后的回调函数 7、async 设置是否异步，默认值是'true'，表示异步</p> <p>以前的写法：</p> <pre>1 \$.ajax({ 2 url: 'js/data.json', 3 type: 'GET', 4 dataType: 'json', 5 data:{'aa':1} 6 success:function(data){ 7 alert(data.name); 8 }, 9 error:function(){ 10 alert('服务器超时，请重试！'); 11 } 12 });</pre> <p>▪ jsonp</p> <p>jsonp ajax只能请求同一个域下的数据或资源，有时候需要跨域请求数据，就需要用到jsonp技术，只需要将dataType:设置为jsonp.就可以跨域请求数据(前提是服务器允许跨域请求)</p> <pre>1 // 跨域请求 2 \$('#d1').click(function () { 3 //获取账号密码 4 var user = \$('#username').val(); 5 var pwd = \$('#password').val(); 6 \$.ajax({ 7 url: 'http://test.lemonban.com/futureloan/mvc/api/member/login', 8 dataType: 'jsonp', 9 method: 'POST', 10 data: {"mobilephone": user, "pwd": pwd}, 11 }).done(function (data) { 12 console.log(data); 13 alert('登录成功') 14 }).fail(function () { 15 alert('登录失败') 16 }) 17 })</pre>	