logo.png

HTML基础

H5 web前端

[www.bufanui.com](http://www.bufanui.com)

# Html简介

## 1.1 概念

* HTML（Hypertext Markup Language） 超文本标记语言。
* 它负责网页的三个要素之中的结构。
* HTML使用标签的的形式来标识网页中的不 同组成部分。
* 所谓超文本指的是超链接，使用超链接可 以让我们从一个页面跳转到另一个页面。

## 1.2 页面结构

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>网页标题</title>  </head>  <body>  <h1>网页正文</h1>  </body>  </html> |

## 1.3 标签

* HTML中的标记指的就是标签。
* HTML使用标记标签来描述网页。
* 结构

<标签名> 内容 </标签名>

## 1.4 属性

可以为HTML标签设置属性。

* 通过属性为HTML元素提供附加信息。
* 属性需要设置在开始标签或自结束标签中。
* 属性总是以名称/值对的形式出现。
* 比如：name=“value”
* 有些属性可以是任意值，有些则必须是指定值。

|  |
| --- |
| <p title=”这是p标签” >我是一个p标签</p> |

* 常见属性

|  |  |
| --- | --- |
| id | id属性作为标签的唯一标识，在同一个网页中不能 出现相同的id属性值。 |
| class | class属性用来为标签分组，拥有相同class属性的 标签我们认为就是一组，可以出现相同的class属 性，可以为一个元素指定多个class。 |
| title | title属性用来指定标签的标题，指定title以后，鼠  标移入到元素上方时，会出现提示文字。 |

## 1.5 注释

HTML注释中的内容不会在网页中显示。

* 格式: <!-- 注释内容 -->
* 合理的使用注释可以帮助开发人员理解网页的代码。
* 注释不能嵌套

# html标签

## 2.1 常用标签

### <meta>

* 设置页面的字符集

– <meta charset="utf-8">

* 设置网页的描述

– <meta name="description" content="不凡学院，做擅长的事，分享知识与快乐！">

* 设置网页的关键字

– <meta name="keywords" content="郑州不凡学院、不凡学院、ui培训、web前端培训">

* 请求的重定向 （了解）

– <meta http-equiv="refresh" content=5;url=xxx"/>

### <html>

* <html>标签用于告诉浏览器这个文档中包含的信息是用HTML编写的。
* 所有的网页的内容都需要编写到html标签中,一个页面中html标签只能有一个。
* html标签中有两个子标签head和body 。

### <head>

* <head>标签用来表示网页的元数据，head中包含了浏览器和搜索引擎使用的其他不可见信息。

### <title>

* <title>标签表示网页的标题，一般会在网页的标题栏上显示。

### <body>

* <body>标签用来设置网页的主体，所有在页面中能看到的内容都应该编写到body标签中。

### <h1> ~ <h6>

* h1~h6都是网页中的标题标签，用来表示网页中的一个标题，不同的是，从h1~h6重要性越来越低。
* 一个页面中只会使用一个h1标签。

### <p>

* <p>标签表示网页中的一个段落。
* 一般浏览器会在段落的前和后各加上一个换行,也就是段落会在页面中自成一行。

### <br>

* 换行符

### <hr>

* 直线分隔符

<img>

* < img />标签是图片标签，可以用来向页面中引入一张外部的图片。
* src 图片路径
* alt 图片的描述

### <a>

* <a>标签是超链接标签，通过a标签，可以跳转到其他页面。
* href 指向一个链接
* target设置打开目标页面的位置

— self 默认

— blank 新窗口打开

— parent 父类窗口

— top 顶级窗口

## 2.2 字体标签

* <em>和<strong> em标签用于表示一段内容中的着重点。strong标签用于表示一个内容的重要性。
* <i>和<b> 跟上面类似，不过没有实际意义。
* <sup>和<sub> 上标和下标
* <ins>和<del> 删除线

## 2.3 列表

* 无序列表： ul li
* 有序列表： ol li
* 自定义列表： dl dt dd

## 2.4 table

使用table标签创建一个表格。

* tr表示表格中的一行。
* tr中可以编写一个或多个th或td。
* th表示表头。
* td表示表格中的一个单元格
* colspan / rowspan 合并单元格
* <caption>表头文字</caption>

### table 属性

* border
* width / height
* cellspacing / cellpadding
* align : left | center | right
* bgcolor:
* border-collapse:collapse 合并边框

## 2.5 表单

表单是用来提交信息的。表单用标签 <form> 表示。

****

* 属性：

— action 提交的地址

— method get/post 表单提交的类型

## 2.6表单控件

* input是我们使用的最多的表单项，它可以根据不同的type属性呈现不同的状态。
* type属性可选值：

– text：文本框

– password：密码框

– submit：提交按钮

– radio：单选按钮 cehcked为默认选中

– checkbox：多选框 cehcked为默认选中

– reset ：重置按钮

* select、option 用于创建一个下拉列表 select为默认选中
* textarea 可以创建多行输入框

— 属性： cols / rows

* fieldset、legend

— fieldset用来为表单项进行分组。legend用于指定每组的名字。

* label 标签用来为表单项定义描述文字。

## 2.7 转义字符

在HTML中预留了一些字符，这些预留字符是不能在网页中直接使用的。比如<和>,我们不能直接在页面中使用<和>号，因为浏览器会将它解析为html标签。所以我们需要使用转义字符。

比如：

|  |
| --- |
| 空格 &nbsp; &#160;  < 小于号 &lt; &#60;  > 大于号 &gt; &#62;  & 和号 &amp; &#38;  " 引号 &quot; &#34;  ' 撇号  &apos; (IE不支持) &#39; |

# css简介

## 3.1 概念

css全称层叠样式表 (Cascading Style Sheets)

* css可以用来为网页创建样式表，通过样式表可以对网页进行装饰。
* 所谓层叠，可以将整个网页想象成是一层一层的结构，层次高的将会覆盖层次低的。css就可以分别为网页的各个层次设置样式。

## 3.2 基本语法

* CSS的样式表由一个一个的样式构成，一个样式又由选择器和声明块构成。
* 语法

|  |
| --- |
| – 选择器 {样式名:样式值；样式名:样式值 ; }  – 标签{color:yellow; font-size:12px;} |

## 3.3 书写位置

* **行内样式**：可以直接将样式写到标签内部的style属性中。

|  |
| --- |
| <p style="color:yellow; font-size:12px;"></p> |

* 这种方式编写简单，定位准确。但是由于直接将css代码写到了html标签的内部，导致结构与表现耦合，同时导致样式不能够复用，所以这种方式我们不使用。
* **内部样式**： 可以直接将样式写到<style>标签中。

|  |
| --- |
| <style>  p{color:yellow; font-size:12px;}  </style> |

* 这样使css独立于html代码，而且可以同时为多个元素设置样式，这是我们演示用的比较多的一种方式。
* 但是这种方式，样式只能在一个页面中使用，不能在多个页面中重复使用。
* **外部样式：**可以将所有的样式保存到一个外部的css文件中，然后通过<link>标签将样式表引入到文件中。

|  |
| --- |
| <link rel="stylesheet" type="text/css"href="style.css"> |

* 这种方式将样式表放入到了页面的外部，可以在多个页面中引入，同时浏览器加载文件时可以使用缓存，这是我们开发中使用的最多的方式。

# css选择器

选择器（selector）用于告诉浏览器：网页上的哪些元素需要设置什么样的样式。

## 4.1 选择器分类

### 元素选择器

* 元素选择器（标签选择器），可以根据标签的名字来从页面中选取指定的元素。
* 语法： 标签名 {}

### 类选择器

* 类选择器，可以根据元素的class属性值选取元素。
* 语法： .classname {}

### ID选择器

* ID选择器，可以根据元素的id属性值选取元素。注： id是唯一的。
* 语法 # id { }

交集选择器

* 可以同时使用多个选择器，这样可以选择同时满足多个选择器的元素。
* 语法： – 选择器1选择器2{}

### 并集选择器

* 可以同时使用多个选择器，多个选择器将被同时应用指定的样式。用逗号隔开。
* 语法： 选择器1,选择器2,选择器3 {}

### 后代选择器

* 可以根据标签的关系，为处在元素内部的代元素设置样式。用空格隔开。
* 语法： 祖先元素 后代元素 后代元素 { }

### 通用选择器

* 可以同时选中页面中的所有元素。
* 语法： \* { }

## 4.2 css特性

### 层叠性

样式的覆盖。 样式的层叠性与样式的调用顺序没有关系，与样式的定义顺序有关。

* 发生前提是：样式冲突。

### 继承性

继承性发生的前提是标签之间属于一种嵌套关系

* 文字颜色可以实现继承
* 文字大小可以实现继承
* 字体可以实现继承
* 行高可以实现继承
* 与文字有关的属性都可以 实现继承
* 特殊性：

<a href=”#”></a> 不能继承父元素中的文字颜色（层叠掉了）

<h1></h1> 标题标签不能继承父元素中的文字大小

### 选择器的权重

* 不同的选择器有不同的权重值：

– 内联样式：权重是 1000

– id选择器：权重是 100

– 类、属性、伪类选择器：权重是 10

– 元素选择器：权重是 1

– 通配符：权重是 0

* 优先级的特点：继承的权重为0
* 权重会叠加

## 4.3 css单位

### 单位： px

如果我们将一个图片放大的话，我们会发现一个图片是有一个一个的小色块构成的，这一个小色块就是一个像素，也就是1px，对于不同的显示器来说一个像素的大小是不同的。

### 百分比：

也可以使用一个百分数来表示一个大小，百分比是相对于父元素来说的。

### em

基于字体大小的倍数

### 颜色

* **预定义颜色：** aqua、black、blue、fuchsia、gray、green、lime、maroon、navy、olive、orange、purple、red、silver、teal、white、yellow。
* **十六进制颜色：**

— #FFFF00包含了三组十六进制的数字。每两位分别表示red、green、blue。

* **rgb**

— rgb(255,255,0) 每一个值对应表示十六进制的red、green、blue

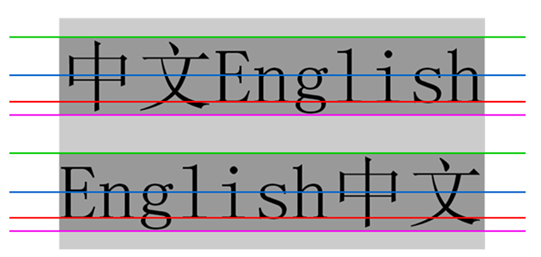
* **rgba**

— RGBA(255,255,0.5) 同rgb，最后一个值表示透明度，0~1

## 4.4 css常见属性

|  |  |  |
| --- | --- | --- |
| 属性 | 解释 | 书写规范 |
| width | 宽度 | width:300px; |
| height | 高度 | height:300px; |
| color | 文本颜色（前景色） | color:red; |
| background-color | 背景颜色 | background-color:green; |
| font-size | 文字大小 | font-size:34px; |
| Text-align | 内容的水平对齐方式 | text-align:left|center|right |
| Text-indent | 首行缩进 | Text-indent:2px; |

|  |  |
| --- | --- |
| font-size | 文字大小 |
| font-family | |  |  | | --- | --- | | 微软雅黑 | Microsoft YaHei | | 黑体 | SimHei | | 宋体 | SimSun | | 楷体 | KaiTi | | 新宋体 | NSimSun | | Arial | ABCDEFGHIJKLMNOPQRSTU | | Impact | ABCDEFGHIJKLMNOPQRSTUVW | |
| font-weight | 100-900.加粗700-900/ bolder lighter normal |
| font-style | Italic 斜体 / normal 正常 |
| Line-heigt | 行高，单位： px /倍数 / 百分比 |
| font | {font: font-style font-weight font-size/line-height font-family;} |

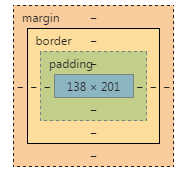


4.5 背景图片

|  |  |  |
| --- | --- | --- |
| background-color: | 背景颜色 |  |
| background-image: | 背景图片 | url(‘1.png’); |
| background-repeat: | 平铺方式 | repeat | no-repeat | repeat-x | repeat-y |
| background-position: | 图片位置 | left | right | top | bottom | center |
| background-attachment: | 背景滚动 | scroll|fixed |
| background | 简写 | background: green url(1.jpg) no-repeat center center fixed; |

# 盒子模型

CSS处理网页时，它认为每个元素都包含在一个不可见的盒子里。



一个盒子我们会分成几个部分：

– 内容区(content)

– 内边距(padding)

– 边框(border)

– 外边距(margin)

## 内容区域

* 内容区指的是盒子中放置内容的区域
* 如果没有为元素设置内边距和边框，则内容区大小默认和盒子大小是一致的。
* 通过width和height两个属性可以设置内容区的大小。
* width和height属性只适用于块元素。

## 内边距

内边距指的就是元素内容区与边框以内的空间

* padding:10px 20px 30px 40px

– 这样会设置元素的上、右、下、左四个方向的内边距。

* padding:10px 20px 30px;

– 分别指定上、左右、下四个方向的内边距

* padding:10px 20px;

– 分别指定上下、左右四个方向的内边距

* padding:10px;

– 同时指定上左右下四个方向的内边距

* 同时在css中还提供了padding-top、padding-right、padding-right、padding-bottom分别用来指定四个方向的内边距。

## 边框

可以在元素周围创建边框，边框是元素可见框的最外部。

* 可以使用border属性来设置盒子的边框：

– border:1px red solid 分别指定了边框的宽度、颜色和样式，样式为：

– none（没有边框）

– dotted（点线）

– dashed（虚线）

– solid（实线）

– double（双线）

* 也可以使用border-top/left/right/bottom分别指定上右下左四个方向的边框。
* 和padding一样，默认width和height并包括边框的宽度。

## 外边距

外边距是元素边框与周围元素相距的空间。

* 用法和padding类似，同样也提供了四个方向的margin-top/right/bottom/left。
* 当将左右外边距设置为auto时，浏览器会将左右外边距设置为相等，可以使元素居中。

### 垂直叠加（重叠）

* 当两个div发生垂直局部时，margin值没有发生累加，最终谁大取谁
* 对比的是margin-bottom 和margin-top的值

### 嵌套情况（坍塌）

* 当div发生嵌套 里面div的margin-top值 直接影响到了父类
* 解决方法：

1.overflow:hidden

2.padding

3.float

### 影响盒子宽度因素：

* 定义的内边距（padding）的宽度
* Border

### 盒子大小情况二

* 继承的盒子在父盒子宽度范围内，padding值不会影响该盒子大小。

## display

* 我们不能为行内元素设置width、height、margin-top和margin-bottom。
* 我们可以通过修改display来修改元素的性质。
* 值：

– block：设置元素为块元素

– inline：设置元素为行内元素

– inline-block：设置元素为行内块元素

– none：隐藏元素

## visibility

* visibility属性主要用于元素是否可见。
* 和display不同，使用visibility隐藏一个元素，隐藏后其在文档中所占的位置会依然保持，不会被其他元素覆盖。
* 值：

– visible：可见的

– hidden：隐藏的

## overflow

当相关标签里面的内容超出了样式的宽度和高度时如何处理

• 可选值：

– visible：默认值

– scroll：添加滚动条

– auto：根据需要添加滚动条

– hidden：隐藏超出盒子的内容

# 文档流

* 块元素独占一行
* 行内元素在一行上显示，如果排不下换行
* 自上而下

# 浮动

## 7.1 浮动的概念

浮动指的是使元素脱离原来的文本流，在父元素中浮动起来。

• 浮动使用float属性。

• 可选值：

– none：不浮动

– left：向左浮动

– right：向右浮动

• 块级元素和行内元素都可以浮动，当一个行内元素浮动以后将会自动变为一

个块级元素。

• 当一个块级元素浮动以后，宽度会默认被内容撑开，所以当漂浮一个块级元

素时我们都会为其指定一个宽度。

## 7.2 浮动的表现形式

• 当一个元素浮动以后，其下方的元素会上移。元素中的内容将会围绕

在元素的周围。

• 浮动会使元素完全脱离文本流，也就是不再在文档中在占用位置。

• 元素设置浮动以后，会一直向上漂浮直到遇到父元素的边界或者其他

浮动元素。

• 元素浮动以后即完全脱离文档流，这时不会再影响父元素的高度。也

就是浮动元素不会撑开父元素。

• 浮动元素默认会变为块元素，即使设置display:inline以后其依然是个

块元素。

## 7.3 清除浮动

* clear属性可以用于清除元素周围的浮动对元素的影响。
* 可选值：

– left：忽略左侧浮动

– right：忽略右侧浮动

– both：忽略全部浮动

– none：不忽略浮动，默认值

# 定位

position属性可以把一个元素放置到网页中的任何位置。

• 可选值：

– static

– relative

– absolute

– fixed

## 相对定位（relative）

每个元素在页面的文档流中都有一个自然位置。相对于这个位置对元素进行移动就称为相对定位。周

围的元素完全不受此影响。

* 当开启了相对定位以后，可以使用top、right、bottom、left四个属性对元素进行定位。
* 如果不设置元素的偏移量，元素位置不会发生改变。
* 相对定位不会使元素脱离文本流。元素在文本流中的位置不会改变。
* 相对定位不会改变元素原来的特性。
* 相对定位会使元素的层级提升，使元素可以覆盖文本流中的元素。

## 绝对定位（absolute）

绝对定位指使元素相对于html元素或离他最近的祖先定位元素进行定位。

* 当开启了绝对定位以后，可以使用top、right、bottom、left四个属性对元素进行定位。
* 绝对定位会使元素完全脱离文本流。
* 绝对定位的块元素的宽度会被其内容撑开。
* 绝对定位会使行内元素变成块元素。
* 一般使用绝对定位时会同时为其父元素指定一个相对定位，以确保元素可以相对于父元素进行定位。

## 固定定位（fixed）

固定定位的元素会被锁定在屏幕的某个位置上，当访问者滚动网页时，固定元素会在屏幕上保持不动。

* 固定定位不占据原来的位置，会脱标。
* 给元素设置固定定位之后，元素位置从浏览器左上角出发。
* 可以将行内元素转换为行内块元素。

## z-index

当元素开启定位以后就可以设置z-index这个属性。

• z-index可以指定一个整数作为参数，值越大元素显示的优先级越高，也就是z-index值较大的元素会显示在网页的最上层。

## 规避脱标流

1. 能用标准流（没有脱标）解决就不用浮动
2. 解决不了就考虑有浮动（页面布局类型，“不完全脱标”）
3. 浮动解决不了用定位（小图标，完全脱标，不影响内容）