# HTML

## Html的运行原理

HTML: Hyper Text Markup Language 超文本标记语言

超文本: 比普通文本功能更加强大,可以添加各种样式

标记语言: 通过一组标签.来对内容进行描述. <关键字> , 是由浏览器来解释执行

1． 本地运行html文件使用浏览器（软件）

2． 远程访问运行 你的电脑（浏览器    远程服务器（hello.html）

## 最基本的HTML结构

<!--

1. **上面是一个文档声明**

2. 根标签 html

3. html文件主要包含两部分. 头部分和体部分

头部分 : 主要是用来放置一些页面信息

体部分 : 主要来放置我们的HTML页面内容

4. 通过标签来对内容进行描述,标签通常都是由开始标签和结束标签组成

5. 标签不区分大小写, 官方建议使用小写

-->

## 最常用的HTML标签

## 布局标签

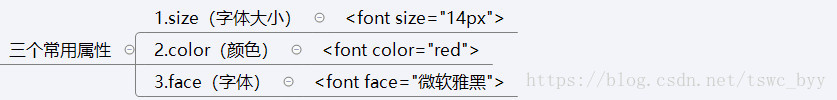
**分区显示标记**

<div></div>**区域**

<span>

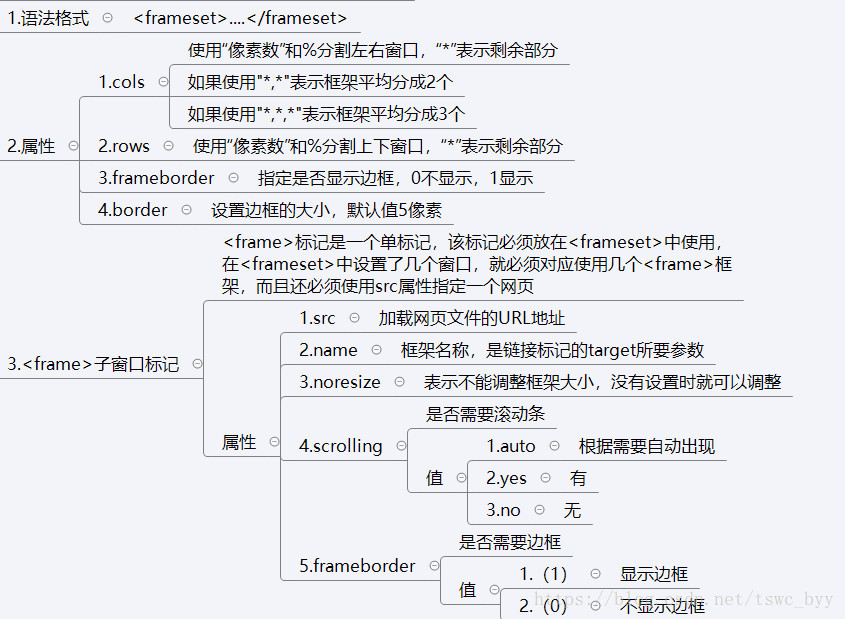
**字体设置标记<font></font>**

**设置字体的格式**

****

**<frameset>框架</frameset>**

**框架将浏览器划分成不同的部分，每一部分加载不同的网页，实现在同一浏览器窗口中加载多个页面的效果**



## 文本标签

p标签定义段落

b/strong标签加粗

都会让文字产生加粗效果

* <strong>用于强调文本，强度更深，表示重要文本--->用于SEO优化
* <b>只是视觉加粗效果--->单纯为了产生加粗

注释<!--注释的内容-->

<h1>----<h6>仅仅用于标题文本，不要为了产生粗体文本使用它们u标签下划线

斜体

* em : 斜体 包含语义em用于强调文本
* i只是视觉斜体效果

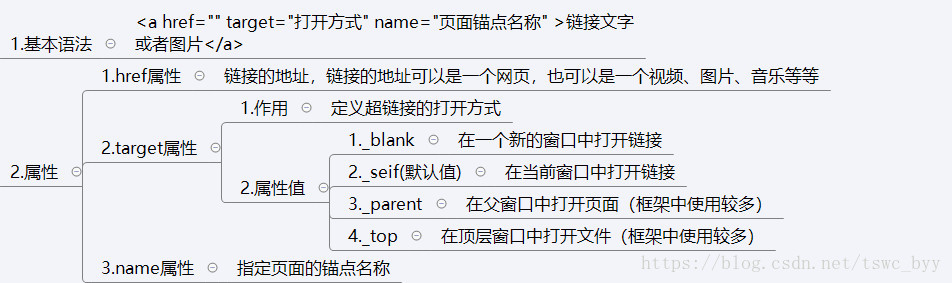
s标签删除线

br标签表示回车换行

hr标签表示水平线

## a标签定义超链接

**超链接的使用<a href=""></a>**



## 序列化标签

ul和li无序列表标签

<ul>

<li>HTML</li>

<li>JS</li>

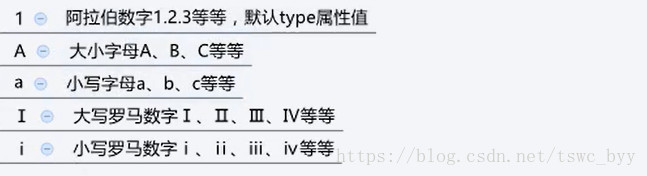
<li>PHP</li>

</ul>

type="circle" 空心圆 type=“disc” 实心圆 默认值 type="square" 方块符

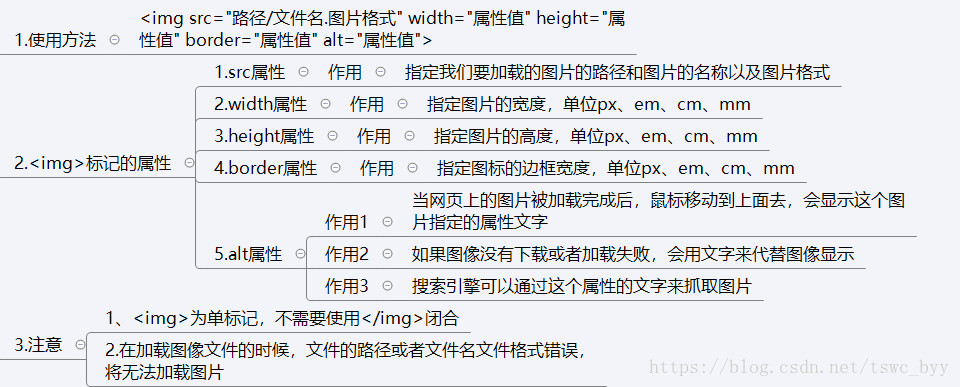
ol和li有序列表标签，可以使用type属性规定有序列表符号的类型。

<ol type="1">



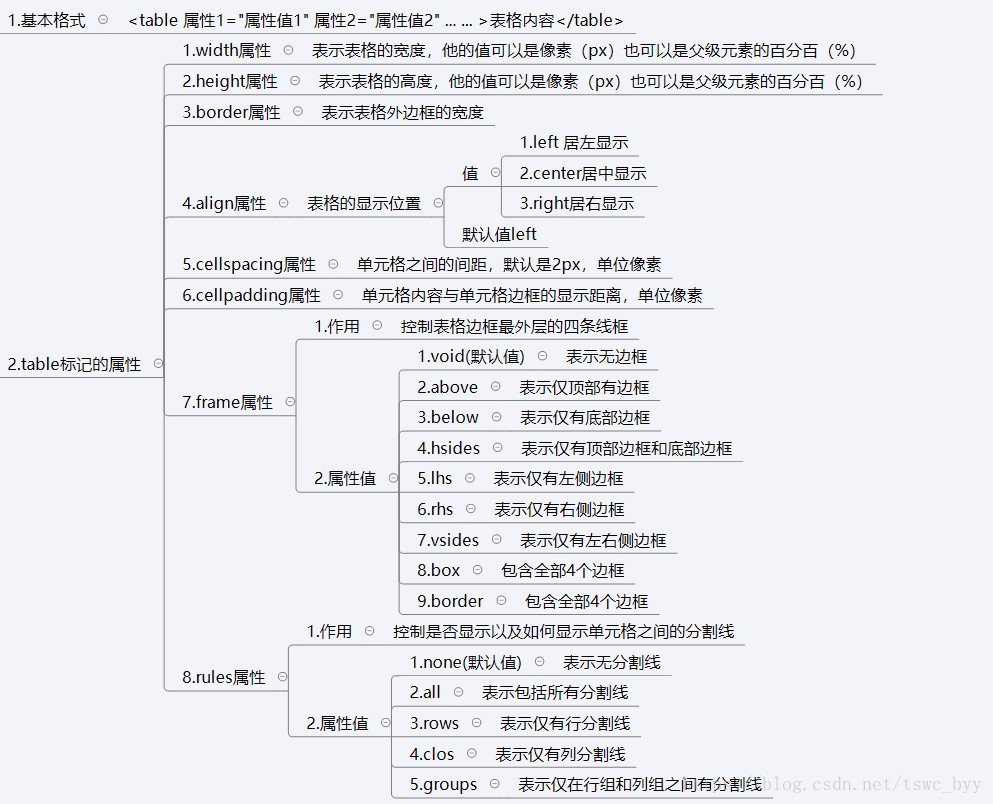
## 多媒体标签

<img>

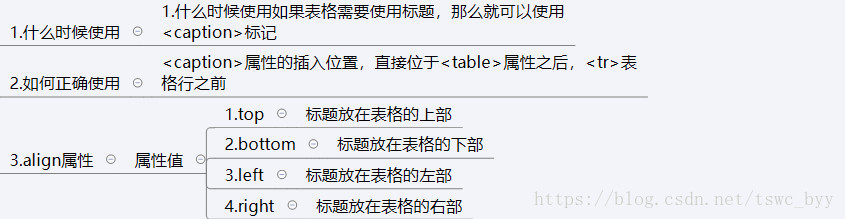


## 表格标签

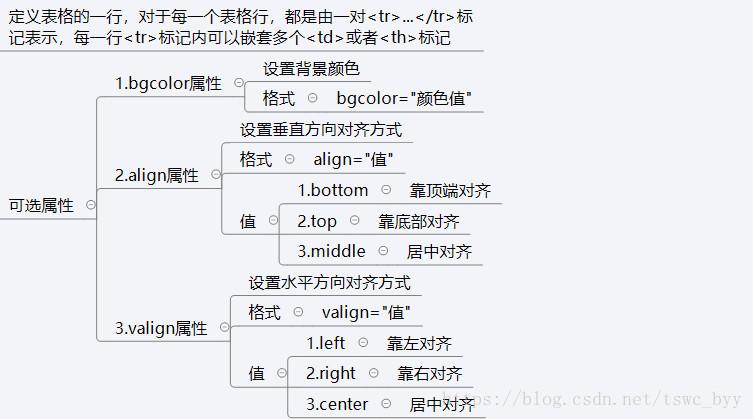
1.<table>标记



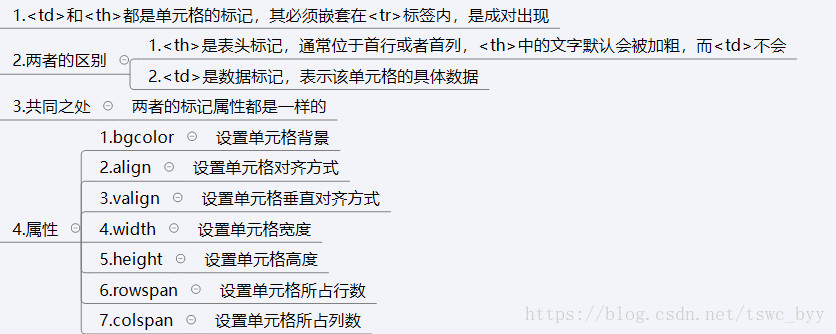
2.<caption>标记



3.<tr>标记

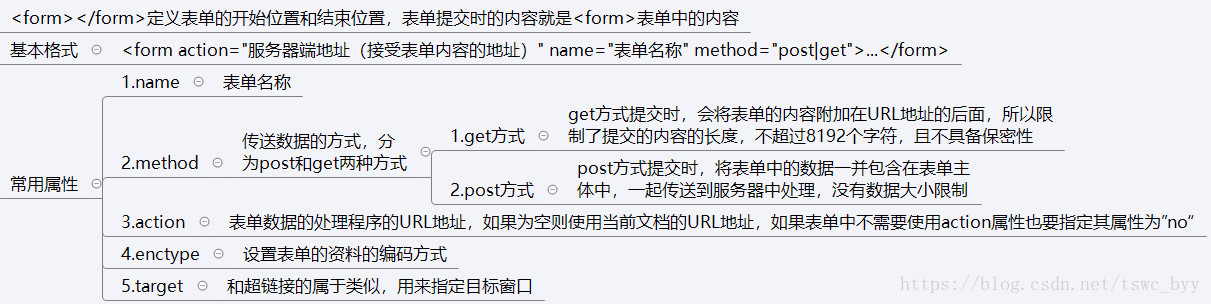


4.<td>和<th>

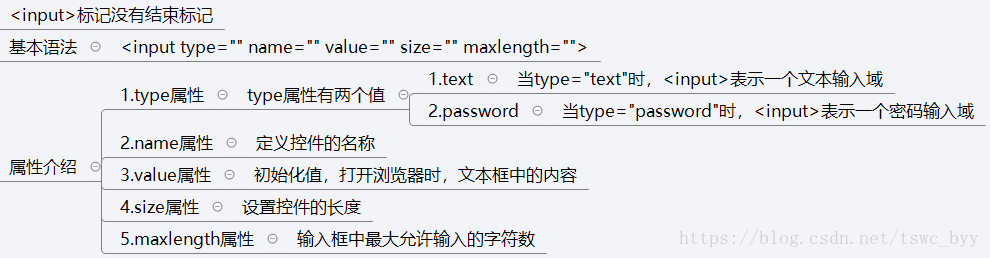


## 表单标签

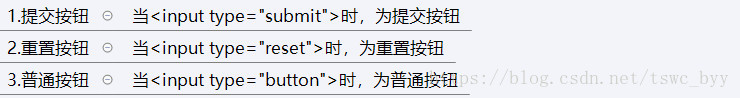
1.表单标记<form>...</form>



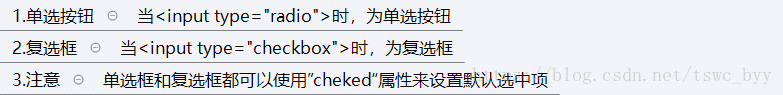
2.文本域和密码<input>标记



3.提交、重置、普通按钮



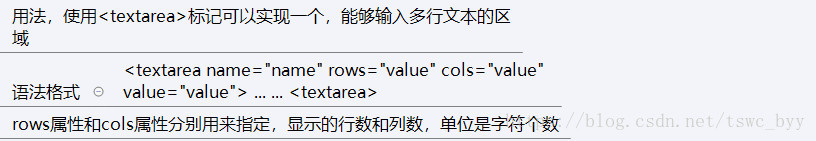
4、单选框和复选框

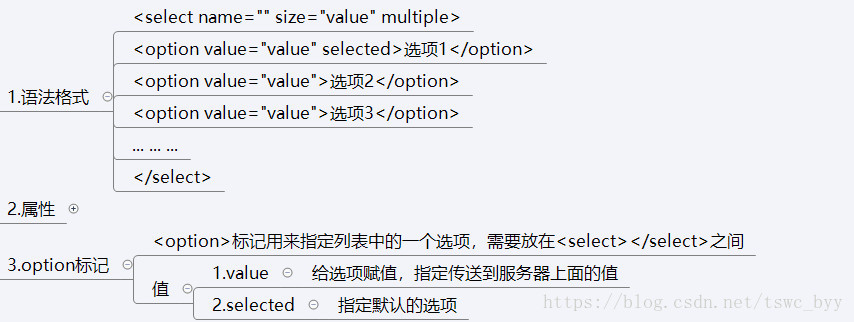


5、隐藏域

当<input type="hidden">时，为隐藏表单域

6、多行文本域

7、菜单下拉列表域<select></select>标记



8、**placeholder 属性的用法**

placeholder 属性规定可描述输入字段预期值的简短的提示信息（比如：一个样本值或者预期格式的短描述）。

该提示会在用户输入值之前显示在输入字段中。

# CSS

## CSS的引入方式

分类及位置：内部样式-head区域style标签里面

* 当单个页面需要设置样式时，就应该使用内部样式表。
* 使用 <style></style>标签在文档<head></head>里面定义内部样式表

从外部引入到样式分为两种：

* Link样式表式： <link rel=”stylesheet” type=”text/css” href=”my.css”(href表示路径)>
* Html式： <style type="text/css">@import url("css.css");></style>

行内样式: 直接在标签中添加一个style属性, 编写CSS样式

### 浮动

使元素脱离文档流，按照指定的方向（左或右发生移动），直到它的**外边缘**碰到包含框或另一个浮动框的边框为止。

说到脱离文档流要说一下什么是文档流，文档流是是文档中可显示对象在排列时所占用的位置/空间，而脱离文档流就是在页面中不占位置了。

在CSS中，通过float属性来定义浮动，其基本语法格式如下：

**选择器{float:属性值;}**

| **属性值** | **描述** |
| --- | --- |
| left | 元素向左浮动 |
| right | 元素向右浮动 |
| none | 元素不浮动（默认值） |

### 清除浮动

1.父级div定义 height, **父元素也设置浮动**

原理：父级div手动定义height，就解决了父级div无法自动获取到高度的问题。

优点：简单、代码少、容易掌握

缺点：只适合高度固定的布局，要给出精确的高度，如果高度和父级div不一样时，会产生问题

2**.父级div**定义 overflow:hidden

原理：必须定义width或zoom:1，同时不能定义height，使用overflow:hidden时，浏览器会自动检查浮动区域的高度

优点：简单、代码少、浏览器支持好

3. **父级div定义 伪类:after 和 zoom**

clearfix:after{content:””; display:block; clear:both;}

4. **结尾处加空div标签** clear:both

原理：添加一个空div，利用css提高的clear:both清除浮动，让父级div能自动获取到高度

优点：简单、代码少、浏览器支持好、不容易出现怪问题

### 显示隐藏

display:none是彻底消失，**不在文档流中占位，**浏览器也不会解析该元素；visibility:hidden是视觉上消失了，可以理解为透明度为0的效果，在文档流中占位，浏览器会解析该元素；

## 类选择器

CSS选择器: 帮助我们找到我们要修饰的标签或者元素

**元素选择**

元素的名称{  
属性名称:属性的值;  
属性名称:属性的值;  
}

**ID选择器**

以#号开头 ID在整个页面中必须是唯一的s  
#ID的名称{  
属性名称:属性的值;  
属性名称:属性的值;  
}

**类选择器**

.类的名称{  
  属性名称:属性的值;  
属性名称:属性的值;  
}

**多类名选择器**

我们可以给标签指定多个类名，从而达到更多的选择目的。

<div class="pink fontWeight font20">亚瑟</div>

**群组选择器**

* 选择多个元素,以逗号隔开 #main,.first,span,a,h1{color:red;}

**属性选择器**

* 选择包含某一属性的元素
* a[title]{color:red;} 选择包含title的a标签
* a[title][href]{color:red;} 选择包含title和href的a标签

**<a>伪类选择器**

* a:link {color:#FF0000;} /\* 未访问的链接 \*/ （只用于a标签）
* a:visited {color:#00FF00;} /\* 已访问的链接 \*/ （只用于a标签）
* **a:hover {color:#FF00FF;}/\* 鼠标移动到链接上**
* \*/（可和其他标签结合一起用）
* a:active {color:#0000FF;} /\* 选定的链接 \*/
* **注意**
  + 伪类选择器的排序很重要，a:link a:visited a:hover a:active，记作**lvha**

## CSS的优先级

按照选择器搜索精确度来编写:行内样式 > ID选择器 > 类选择器 > 元素选择器

就近原则: 哪个离得近,就选用哪个的样式

css specificity

!important声明specificity值优先级最高

**算法：**当遇到多个选择符同时出现时候   
按选择符得到的Specificity值逐位相加，

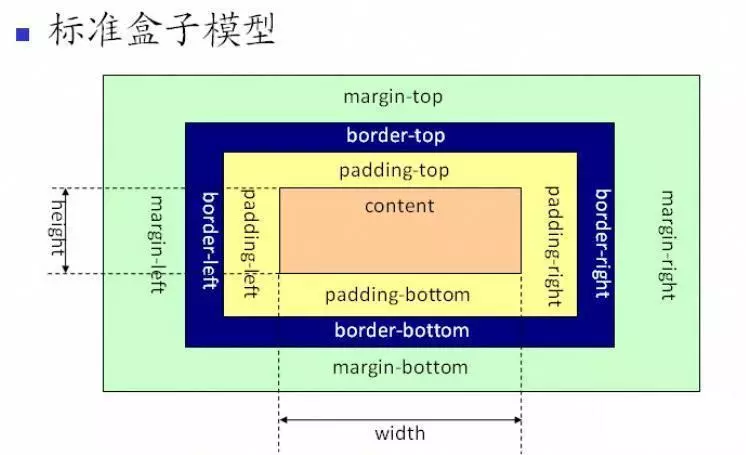
## 文字文本属性

* css文字文本属性：
  + **文字属性**
    - color:red; 文字颜色
    - font-size:12px; 文字大小
    - font-weight:“bold” 文字粗细(bold/normal)
    - font-family:“宋体” 文字字体
    - font-variant:small-caps小写字母以大写字母显示
* **文本属性**
  + text-align:center; 文本对齐(right/left/center)
  + line-height:10px; 行间距(可通过它实现文本的垂直居中)
  + text-indent:20px; 首行缩进
  + text-decoration:none;
    - 文本线(none/underline/overline/line-through)
  + letter-spacing: 字间距

## 盒子模型

* 盒子模型就是一个有高度和宽度的矩形区域
* 所有html标签都是盒子模型
* div标签自定义盒子模型

**所谓盒子模型**就是把HTML页面中的元素看作是一个矩形的盒子，也就是一个盛装内容的容器。每个矩形都由元素的内容、内边距（padding）、边框（border）和外边距（margin）组成。



**盒子边框（border）**

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

边框属性—设置边框样式（border-style）

边框样式用于定义页面中边框的风格，常用属性值如下：

none：没有边框即忽略所有边框的宽度（默认值）solid：边框为单实线(最为常用的)dashed：边框为虚线 dotted：边框为点线double：边框为双实线

table{ border-collapse:collapse; } collapse 单词是合并的意思

border-collapse:collapse; 表示边框合并在一起。

**内边距（padding）**

padding属性用于设置内边距。 **是指边框与内容之间的距离。**

padding-top:上内边距

padding-right:右内边距

padding-bottom:下内边距

padding-left:左内边距

注意： 后面跟几个数值表示的意思是不一样的。

值：像素/厘米等长度单位、百分比

* padding:10px; 上下左右
* padding:10px 10px; 上下 左右
* padding:10px 10px 10px; 上 左右 下
* padding:10px 10px 10px 10px; 上 右 下 左（设置4个点-->顺时针方向）

**外边距（margin）**

margin属性用于设置外边距。 设置外边距会在元素之间创建“空白”， 这段空白通常**不能放置其他内容.**

## 元素的定位属性

元素的定位属性主要包括定位模式和边偏移两部分。

1、边偏移

边偏移属性 描述

top 顶端偏移量，定义元素相对于其父元素上边线的距离

bottom 底部偏移量，定义元素相对于其父元素下边线的距离

left 左侧偏移量，定义元素相对于其父元素左边线的距离

right 右侧偏移量，定义元素相对于其父元素右边线的距离

也就说，以后定位要和这边偏移搭配使用了， 比如 top: 100px; left: 30px; 等等

2、定位模式(定位的分类)

在CSS中，position属性用于定义元素的定位模式，其基本语法格式如下：

选择器{position:属性值;}

position属性的常用值

值 描述

static 自动定位（默认定位方式）

relative 相对定位，相对于其原文档流的位置进行定位

absolute 绝对定位，相对于其上一个已经定位的父元素进行定位

fixed 固定定位，相对于浏览器窗口进行定位

# 前端应用

## 悬停显示文字，移走消失

**方法一：利用html特性，每个标签都有一个title属性。当鼠标hover在该标签内容上时，浏览器展示出该标签的title内容，让鼠标移走，内容消失。**

**方法二，利用css的伪类hover，以及显示隐藏属性display**