# HTML

## 背景以及基础了解

简介：Hyper Text Markup Language，超文本标记语言。

目前版本：HTML5+CSS3

优势：世界知名浏览器厂商对HTML5的支持。

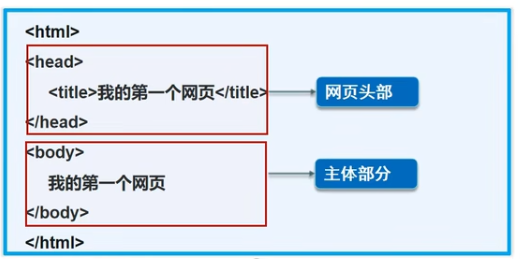
W3C标准：www是world wide web简称，w3c是world wide web consortium万维网联盟，这是web领域最具权威的国际中立性技术标准机构，<http://www.w3.org>

org结尾表示开源

W3C标准包括：

* 结构化标准语言，HTML,XML
* 表现标准语言CSS
* 行为标准DOM,ECMAScript

HTML基本结构



HTML引入外部CSS文件

<link rel="stylesheet" type="text/css" href="../css/p9.css" />

HTML引入外部JS文件，头部和尾部各一个，头部引入多个js文件只有第一个生效

一般头部引入Jquery,尾部引入该页面的JS文件！

<head>

<script src="../jquery/jquery.js"></script>  
</head>

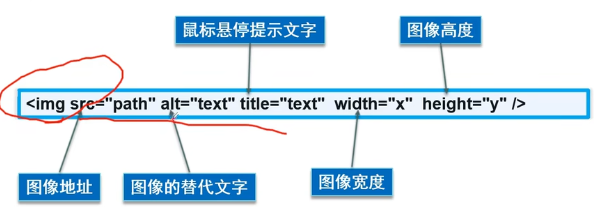
## HTML基本结构信息

<!DOCTYPE html>  
<html lang="en">  
<!--head代表网页头部-->  
<head>  
 <!--meta描述性标签，用来描述我们网站的一些信息 -->  
 <!-- meta用来做SEO -->  
 <meta charset="UTF-8">  
 <!--title网页标题 -->  
 <title>Title</title>  
</head>  
<!--body代表网页主题-->  
<body>  
  
</body>  
</html>

## 网页基本标签

* 标题
* 段落
* 换号
* 斜体和粗体
* 水平线
* 注释和特殊符号
* <!--标题标签-->  
  <h1>一级标签</h1>  
  <h2>二级标签</h2>  
  <h3>三级标签</h3>  
    
  <!--段落标签-->  
  <p>两只老虎 - 贝瓦儿歌</p>   
    
  <!--换行标签-->  
  <br/>  
  <!--水平线标签-->  
  <hr/>  
  <!--粗体和斜体-->  
  粗体：<strong>I LOVE YOU</strong>  
  斜体：<em>YOU HATE ME</em>  
    
  <!--特殊符号:&开头，;结尾-->  
  空格：&nbsp;  
  空&nbsp;&nbsp;&nbsp;&nbsp;格  
    
  &gt;大于号<br/>  
  &lt;小于号<br/>  
  &copy;版权sdm<br/>

### 图像标签



<!--src:图片地址  
 相对地址：  
 ./ ：表示当前目录  
 ../ ：表示上一级目录  
-->  
<!--src:资源路径,必填，alt：资源未找到则显示的文字，必填，title:鼠标的悬停文字-->  
<img src="./3.jpg" alt="Vennus图片" title="悬停文字" width="1920" height="1080">

### 链接标签



<!--  
 a：必填，表示要跳转的页面  
 target:表示新页面的打开位置  
 \_blank：在新页面中打开  
 \_self:在当前页面中打开  
-->  
  
<a href="https://www.baidu.com" target="\_self">百度</a>  
<p>  
<a href="图形标签.html">点我跳转图片标签</a>  
</p>  
<!--锚链接  
 第一步需要一个标记  
 第二步跳转到标记位置，  
 写法：#标记  
-->  
<a href="基本标签.html#ta">跳转</a>  
  
<!--功能性连接  
 邮箱连接：mailto:  
 QQ连接：  
-->  
<a href="mailto:2846456105@qq.com">点击联系我</a>

### 行内元素和块元素\*\*\*

* 块级元素block
  + 无论内容有多少，该元素独占一行，如<p><h1>
* 行内元素inline
  + 元素宽度和高度由内容决定，左右都是行内元素的可以排在一行
    - a,strong,em

所以块级元素可以包含块级元素和行内元素，而行内元素只能有行内元素！

### 列表

列表就是信息资源的展示形式，使信息结构化

列表的分类：

* 有序列表ol
* 无序列表ul
* 自定义列表dl
* <!--有序列表,应用范围试卷，问答-->  
   <ol>  
   <li>java</li>  
   <li>python</li>  
   <li>运维</li>  
   <li>前端</li>  
   </ol>  
  <hr>  
  <!--无序列表，导航，侧边栏-->  
   <ul>  
   <li>java</li>  
   <li>python</li>  
   <li>运维</li>  
   <li>前端</li>  
   </ul>  
  <hr>  
  <!--自定义列表  
   dl:标签  
   dt:列表名称  
   dd:列表内容  
  -->  
  <dl>  
   <dt>学科</dt>  
   <dd>python</dd>  
   <dd>java</dd>  
  </dl>

### 表格

基本元素：

* 行
* 列
* 跨行
* 跨列

单元格根据一行有几列生成

<table border="1px">  
 <!-- 以行开头，一行三列 -->  
 <tr>  
 <!--td属性跨列(一列独占几列)colspan -->  
 <!--td属性跨行(一行独占几行)rowspan -->  
 <td colspan="3">1-1</td>  
 </tr>  
 <tr>  
 <td>2-1</td>  
 <td>2-2</td>  
 <td>2-3</td>  
 </tr>  
 <tr>  
 <td>3-1</td>  
 <td>3-2</td>  
 <td>3-3</td>  
 </tr>

### 视频和音频

视频元素标签：video

<!--controls控制条，autopaly自动播放 -->  
<video src="" controls autoplay></video>

音频元素标签：audio，属性如上

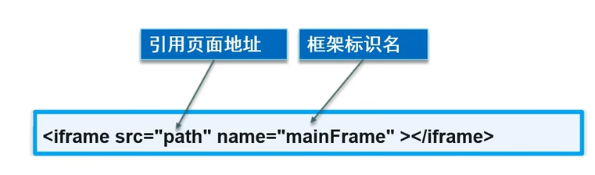
<audio src="" controls autoplay></audio>

### 页面结构分析



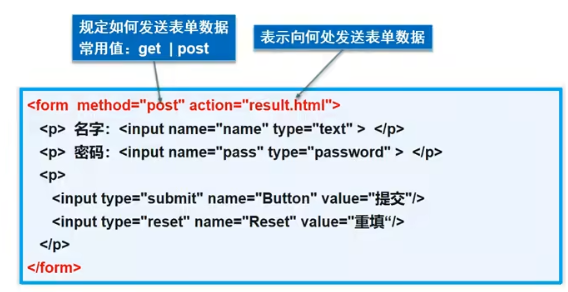
### iframe内联框架

网站中嵌入另一个网页



<!--src:内联地址，w:宽度，h:高度 -->  
<iframe src="https://www.baidu.com" frameborder="0" width="800px" height="500px"></iframe>

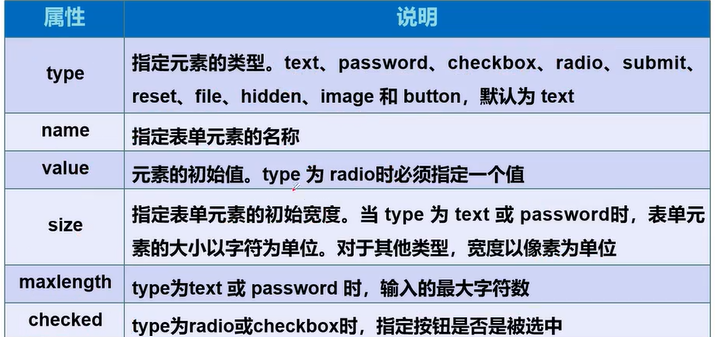
### 表单语法\*\*\*



input标签的多种类型

!--action:可以是网页，也可以是请求处理地址  
 method:post,get  
 get:地址栏显示提交信息，不安全但是高效  
 post:地址栏不显示信息，传输大文件  
-->  
<form action="表单.html" method="get">  
 <p>名字:<input type="text" name="username"></p>  
 <p>密码:<input type="password" name="pwd"></p>  
 <p>  
 <input type="submit">  
 <input type="reset">  
 </p>  
</form>

表单元素格式：input标签属性



button按钮的属性，onclick绑定JS函数，

disabled属性使文本框只读不可编辑且变灰色,且不会被表单提交

readonly属性表示当前input标签内容只读，会被表单提交！

<!--radio:单选框  
 name:表示组，单选框的不同值需要放在同一组，否则单选框也可以多选  
 value:单选框的值

-->  
<p><input type="radio" value="boy" name="sex">男<input type="radio" value="girl" name="sex">女</p>

### 表单的应用

* 隐藏域hidden
* 只读readonly
* 禁用disabled

以上都是input标签属性

### 表单的初级验证

保证的数据的有效性，安全性，降低后端压力

* placeholder:给出提示信息
* required:非空判断，不能为空
* pattern 正则表达式

<!--表单的初级验证-->  
<p>邮箱:<input type="email" name="email" pattern="^[a-zA-Z0-9\_-]+@[a-zA-Z0-9\_-]+(\.[a-zA-Z0-9\_-]+)+$"></p>

### 按钮

input标签的按钮属性

<!--radio:单选框  
 name:表示组，单选框的不同值需要放在同一组，否则单选框也可以多选  
 value:单选框的值  
-->  
<p><input type="radio" value="boy" name="sex">男<input type="radio" value="girl" name="sex">女</p>  
<!--  
 多选框:checkbox，jquery的change()方法可以判断复选框状态是否改变，复选框的属性checked只要被写出，默认就是被选中状态，无论其赋值是什么！

在jquery中，checked属性为true时，就是选中，为false就是不选中

同时使用.prop(‘checked’)方法获取是否被选中，选中为true,没选中则为false  
-->  
<p><input type="checkbox" value="sleep" name="hobby">睡觉  
 <input type="checkbox" value="playgames" name="hobby">打游戏  
 <input type="checkbox" value="writecodes" name="hobby">敲代码  
 <input type="checkbox" value="watchTV" name="hobby">看电视  
</p>  
<!--button:普通按钮  
 imgage:图像按钮  
 submit:提交按钮  
 reset:重置按钮  
-->  
<p><input type="button" name="bt1" value="按钮">  
 <input type="image" src="./3.jpg" onclick="p()">  
</p>  
<p>  
<input type="submit">  
<input type="reset">

### 下拉框

<!--下拉框-->  
<p> 国家:  
 <select name="列表名称">  
 <option value="China">中国</option>  
 <option value="American">美国</option>  
 <option value="Eth">瑞士</option>  
 <option value="India">印度</option>  
 </select>  
</p>

### 文件域和文本域

<!--文本域-->  
<p>反馈:  
 <!--name属性，col：文本域列数，rows 文本域行数 -->  
 <textarea name="反馈内容" cols="50" rows="10">反馈内容:</textarea>  
</p>  
<!--文件域  
 input标签file类型，凡是上传数据都是以key-value形式，所以上传文件需要添加name属性，后台就可以通过  
 name属性取value  
-->  
<p>  
 <input type="file" name="files">  
 <input type="button" value="上传" name="upload">  
</p>

### input补充

<!--邮件验证,以下这些具有一定的自动验证功能-->  
<p>  
 邮箱:<input type="email" name="email">  
 <!--URL-->  
 URL:<input type="url" name="url">  
</p>  
<!--数字-->  
<p>商品数量:  
 <input type="number" min="0" max="10" step="1" name="product">  
</p>  
<!--滑块-->  
<p>  
 音量:<input type="range" name="voice" min="0" max="10" step="2">  
</p>  
<!--搜索框-->  
<p>  
 <input type="search" name="search">  
</p>

### label标签

<!--增强鼠标可用性，点击后，id为mark的标签亮起-->  
<label for="mark">你点我</label>

# CSS

前端三要素Html+CSS+Javascript(结构+表现+交互)

快速入门：

* CSS什么
* CSS怎么样
* CSS选择器\*\*\*
* 美化网页
* 盒子模型
* 浮动
* 定位
* 网页动画

## CSS简介

Cascading style sheet层叠级联样式表

CSS：表现，美化网页

字体颜色边距，高度，宽度，网页定位

### CSS发展史

CSS1.0

CSS2.0 DIV+CSS, html和CSS结构分离的思想

CSS2.1 浮动和定位

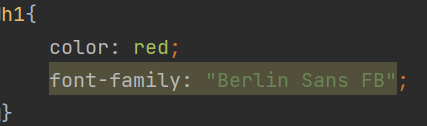
CSS3.0 圆角,阴影,动画,浏览器兼容性

## 快速入门

html，css分离式写法：



style.css如下：



CSS优势：

* 内容和表现分离
* 网页机构表现统一，可以复用
* 样式丰富
* 建议Html和CSS分离式使用
* 利用SEO，容易被搜索引擎收录

### CSS的三种引入方式

* 行内样式
* 内部样式
* 外部样式

<!--行内样式：在标签元素中，编写一个style属性即可 -->  
 <h1 style="color: green">我是标题</h1>

//内部样式

<!--CSS写在head头部，规范语法如下：每一个声明最好使用分号结尾  
 选择器{  
 声明:...;  
 ...  
 }  
-->  
<style>  
 h1{  
 color: red;  
 font-family: "Berlin Sans FB";  
 }  
</style>

外部样式链接式写法：(还有导入时式写法，不推荐)

<!—引入外部CSS文件-->

<link rel="stylesheet" href="../css/style.css">

外部CSS文件写法：

h1{  
 color: blue;  
 font-family: "Berlin Sans FB";  
}

三者优先级：

根据就近原则。行内样式优先级最好，内部样式的引入和外部样式谁和目标最近，谁生效！

## CSS选择器\*\*\*

作用：选中页面上的某一个元素

### 基本选择器

1. 标签选择器 标签名{}
2. 类选择器 .class{}
3. id选择器 #id{}

基本选择器优先级问题：不遵循就近原则

id选择器>class选择器>标签选择器

### 层次选择器

1. 后代选择器：在某个标签的所有子标签，语法空格
2. 子选择器：某个标签的一级子标签，语法>
3. 相邻兄弟选择器：向下第一个兄弟标签，语法+
4. 通用兄弟选择器：向下所有兄弟标签，语法~

/\*后代选择器\*/  
body p{/\*body 后代中所有的p\*/  
 color: red;  
}

/\*子选择器body下的一级p标签\*/  
body>p{  
 color: #a320f5;  
}

/\*相邻兄弟选择器，只能有一个，而且是向下指向\*/  
.active+ul{  
 color: green;  
}

/\*通用兄弟选择器，向下所有兄弟标签\*/  
.active~p{  
 background: red;  
}

### 结构伪类选择器

/\*ul的第一个子元素\*/  
ul li:first-child{  
 background: black;  
}  
/\*ul最后一个子元素\*/  
ul li:last-child{  
 background: red;  
}  
/\*选中p1:定位到父元素的第一个子元素  
选中当前p元素的父级元素，并且选中父元素的第一个子标签p  
\*/  
p:nth-child(1){/\*p标签的父元素的子标签的第一个p标签\*/  
 color: green;  
}  
p:nth-of-type(2){/\*选中父元素下的类型为p的标签的第二个\*/  
 color: red;  
}

### 属性选择器\*\*\*

常用，把id选择器和class选择器结合使用

/\*选中a标签有某个属性的元素：a[id/class/...]{}\*/  
 a[id]{  
 background: yellow;  
 }  
 a[class]{  
 background: green;  
 }  
 /\*class中有具体名称的元素\*/  
 a[class="4th"]{  
 background: darkmagenta;  
 }  
 /\*选中a标签中以http开头的  
 = :绝对等于  
 \*=:包含这个元素  
 ^=:以这个开头  
 $=:以这个结尾  
 \*/  
 a[href^=http]{  
 background: linen;  
 }  
 </style>  
</head>  
<body>  
<p class="demo">  
 <a href="http://www.sdm.com" id="first">1</a>  
 <a href="" class="2th" >2</a>  
 <a href="" id="third">3</a>  
 <a href="" class="4th" >4</a>  
 <a href="" class="5th" >5</a>  
 <a href="" id="sixth">6</a>  
 <a href="" class="7th" >7</a>  
 <a href="" class="8th" id="eighth">8</a>  
 <a href="" id="ninth">9</a>  
</p>

## 美化网页元素

美化网页加深用户印象，有效传递信息。凸显主题，提高用户体验！

### 美化元素的标签

span标签(块元素)

#### 字体元素

* font-size字体尺寸
* font-family字体样式，英文和中文有对应的字体样式！
* font-weight字体粗细
* color字体颜色
* font字体风格，粗细，大小统一设置

#### 文本元素

文本排版：

* text-align文本排版，居中居左居右
* text-indent段落首行缩进
* line-height行高
* text-decoration:underline/line-through/overline上中下划线
* vertical-align:middle文本图片水平对齐

#### 超链接伪类

链接：

* a:hover:鼠标悬浮状态
* a:active:鼠标按住状态
* text-shadow:阴影效果

/\*鼠标触碰悬浮时的状态，常用\*/  
a:hover{  
 color: orange;  
}  
/\*鼠标按住未释放的状态\*/  
a:active{  
 color: green;  
}  
/\*阴影颜色，水平偏移，垂直偏移，阴影半径\*/  
h1{  
 text-shadow: #3300fe 10px 0px 2px;  
}

### 列表元素

list-style:

* none:去掉原点
* circle:空心圆
* decimal:数字
* square:正方形

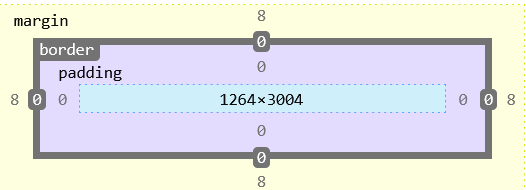
### 背景

background:

* color背景颜色
  + 渐变色工具网站：<https://www.grabient.com/>
* img背景图片
* position背景图片位置
* repeat背景平铺方式

div{  
 width: 1000px;  
 height: 1000px;  
 border: 1px solid red;  
 /\*图片背景默认是全部平铺的，即图片尺寸过小，将图片反复平铺至整个容器\*/  
 background-image: url("../img/6.jpg");  
}  
.div1{  
 /\*x向平铺\*/  
 background-repeat: repeat-x;  
}  
.div2{  
 /\*y向平铺\*/  
 background-repeat: repeat-y;  
}  
.div3{  
 /\*背景:颜色color，图片img，图片位置position，平铺方式repeat\*/  
 background:red url("../img/img.png") 0px 0px no-repeat;  
}  
body{  
 background-color: #ffffff;  
 background-image: linear-gradient(180deg, #ffffff 0%, #9f18f1 66%, #110101 100%);  
}

## 盒子模型



margin：外边距，更多其它写法详情查询文档<https://www.w3school.com.cn/css/index.asp>

/\*外边距上方\*/  
margin-top: 20px;  
/\*外边距下方\*/  
margin-bottom: 40px;  
/\*外边距左边\*/  
margin-left: 40px;  
/\*外边距右边\*/  
margin-right: 40px;

/\*外边距居中\*/  
margin:auto;

border：边框

/\*边框宽度，边框类型，边框颜色\*/  
border: 1px solid red;

padding：内边距，参照外边距，详情查询文档<https://www.w3school.com.cn/css/index.asp>

注意：

* html很多标签都有一个内外边距的默认值。如body标签外边距默认值是8
* 网页元素的大小=margin+border+padding+内容宽高

### 圆角边框

设置盒子模型的角为圆角！实现功能网页头像，具体参照文档

div{  
 height: 100px;  
 width: 100px;  
 border: 10px solid red;  
 /\*左上，右上，右下，左下，值最高为100，值越高，越趋近于圆\*/  
 border-radius: 50px 20px 10px 5px;  
}

### 盒子阴影

box-shadow: 10px 10px 100px yellow;

## 浮动float和dispaly

float:

* left:左浮动
* right:右浮动

浮动的问题：边框塌陷问题

边框塌陷问题的解决办法：

* clear：left左侧不允许有浮动
* clear：right右侧不允许有浮动
* clear：both两侧不允许有浮动
* 增加父元素的高度
* 增加一个空的div,清除浮动
* 在父级元素中增加overflow属性
  + hidden溢出部分自动隐藏
  + scroll溢出部分以滚动条方式显示
  + 父增加一个伪类:after
    - 

display:

* inline:行内元素
* block:块元素
* inline-block:具有二者共同的特性，inline在前，优先是行元素，然后又是块

块级元素：独占一行

* h1-h6,div,p,列表

行内元素

* span,a,img,

块级元素可以包含行内元素，行内元素不可以包含块级元素！

/\*dispaly:  
\*block块元素  
\*inline行内元素  
\*inline-block 保持行内元素和块元素共同的特性！  
\*none消失  
\*/  
div{  
 width: 100px;  
 height: 100px;  
 border:5px solid red;  
 display: inline;  
}  
span{  
 /\*span作为行内元素，即使设置了宽高，但实际所占空间依旧由内容决定\*/  
 width: 100px;  
 height: 100px;  
 border:5px solid red;  
 /\*将行内元素设置成块元素\*/  
 display: inline-block;  
}

### 二者对比

display:不可控制方向

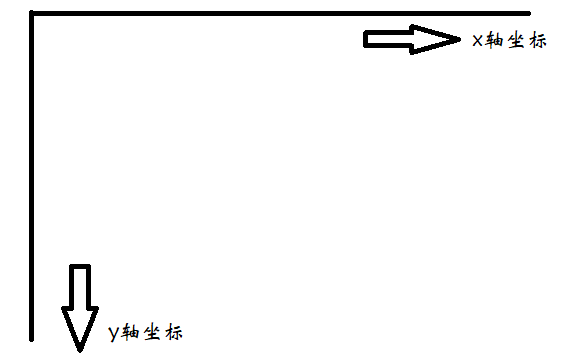
float:浮动起来会脱离标准文档流，所以要解决父级边框塌陷问题

## 定位问题

### 相对对位和绝对定位

* 相对定位：相对于原来的位置指定偏移，它仍然在标准文档流中，原来的位置会被保留
  + position:relative
    - top
    - bottom
    - left
    - right
* 绝对定位：没有父级元素定位的前提下，相对于浏览器定位，父级元素存在定位的情况下，相对于父级元素定位！不存在负值
  + position:absolute定位会随着浏览器页面动而动
    - right:10px 距离浏览器（或者父元素）距离10px
    - left
    - tom
    - bottom
  + position:fixed定位固定在屏幕的固定位置，全程不动
    - right
    - left
    - top
    - bottom
* z-index

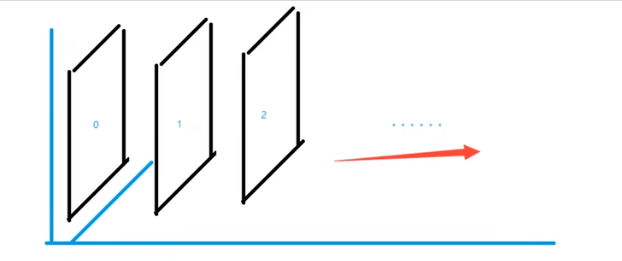
相对定位坐标如下：



## z-index

z-index:图层的层级关系，层级最低是0，低层级图层会被高层级图层遮掩！

只有使用了定位才可以堆叠，否则就是平铺！



透明度：opacity