# 问题

1 bgcolor（标签属性）和background（CSS）这两个有什么区别？

2 除了使用text-align可以设置剧中效果，还有啥可以？

学习某个东西：这个东西是什么？（概念）为什么要去使用这个东西。出现的背景是啥？

很详细的深入，推敲每个词每个字的意思。笔记后期自己要能看懂

# HTML5

超文本标记语言，文件实体为.html或.htm（.htm不怎么使用了）

HTML标签不区分大小写（最好遵循小写，遵循编码规范）。

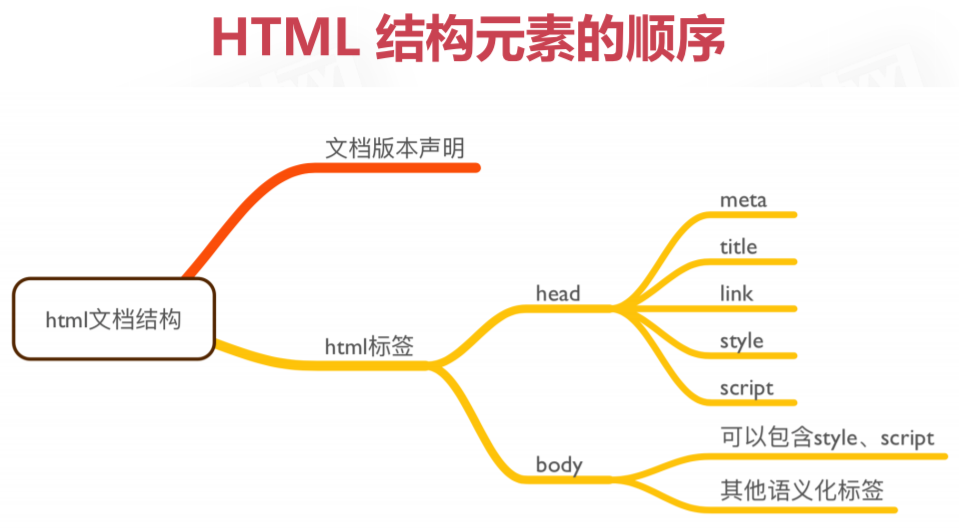
# 结构

文档结构组成

首行文档声明（<!DOCTYPE html>）+一包（<html></html>）+两段（<head></head>和<body></body>）

指明文档编码的标签<meta charset="UTF-8">，utf-8小写也可以

一般文档结构组成如下图所示（注意，在body中也可以放置style标签）：



HTML5标签结构

属性必须在开始标签中定义，不能在结束标签中定义。

一般：<开始标签 […属性]>标签内容<结束标签>，即一般都有结束标签。

例外的结构：

表单输入标签，<input type= “text” value=“” />

换行标签，</br>，很像一个结束标签

# 属性

HTML属性，属性的作用是给予标签基本标识或者给予标签基本样式、特征

通用属性/全局属性，

id，（网页中标签的唯一标识，不应该出现相同的标识）

class，用来指定标签的类名

style，用来设置内联样式

特有属性

lang属性，是对页面中的主要语言做说明，也可以用作对特定标签中的语言做说明，其中en属性值表示英文，zh属性值表示中文

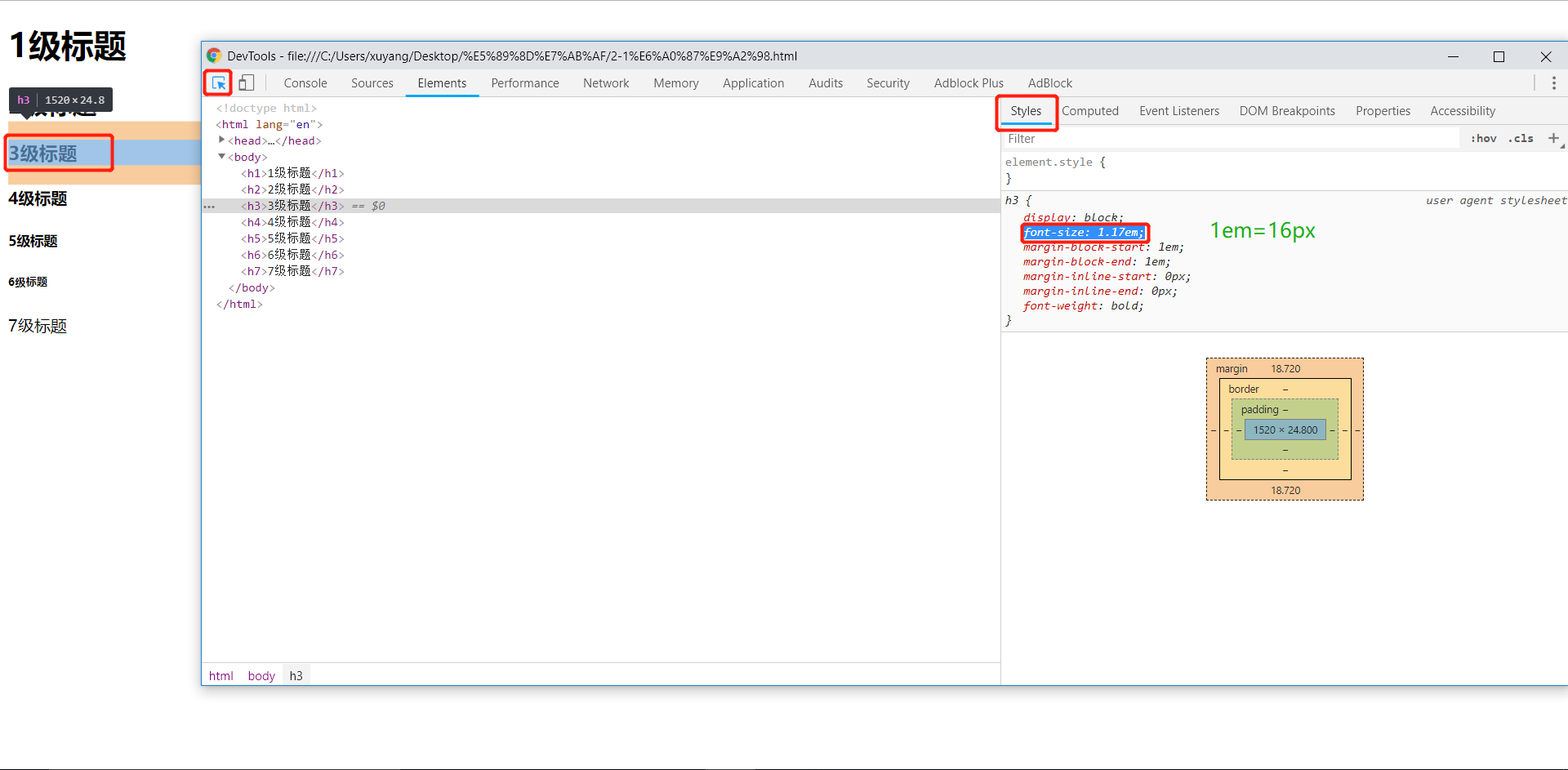
# <hx>标题标签

标题标签，HTML5中，标题标签在某些浏览器中是可以不闭合的，因此如果结束标签和开始标签不一致也会根据开始标签进行设定。

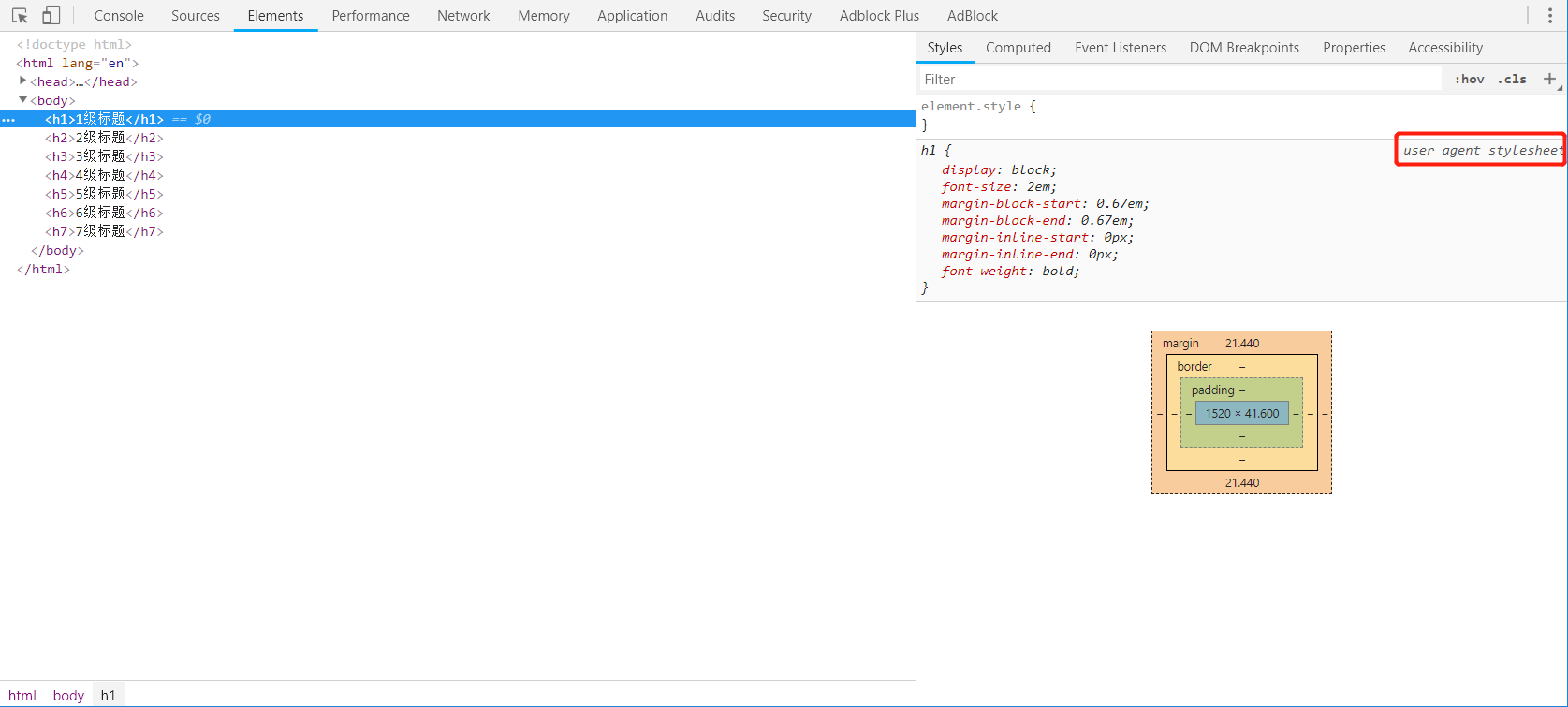
标题标签对SEO是有影响的。

<h1></h1>，这样的标题标签一共有六级

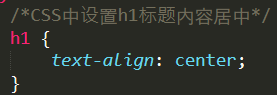
查看样式，使用Chrome浏览器的F12打开浏览器的开发环境，选择左上角的选择元素，点击页面中的3级标题，在样式中观察字体大小，其中em和px的换算关系是1em等于16px。



下图中的”user agent stylesheet”表示这个标签的样式是浏览器给的默认样式，而不是用户自己设定的样式。



如下所示，为设置标题的内容居中。

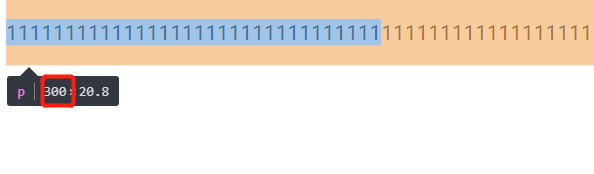


# 段落标签<p>（出现次数很高）

具有浏览器的默认样式，标签内如果没有内容也是占用空间的，所以如果要使用，要考虑覆盖<p>标签的默认样式的必要性。

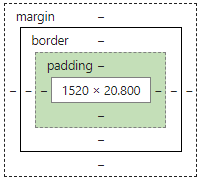
p标签内如果不放内容，都无法撑起p标签，给人的感觉就是p标签不存在一样。

p标签即使设置了宽度，如果内容超过宽度，也不会换行。如下图所示：

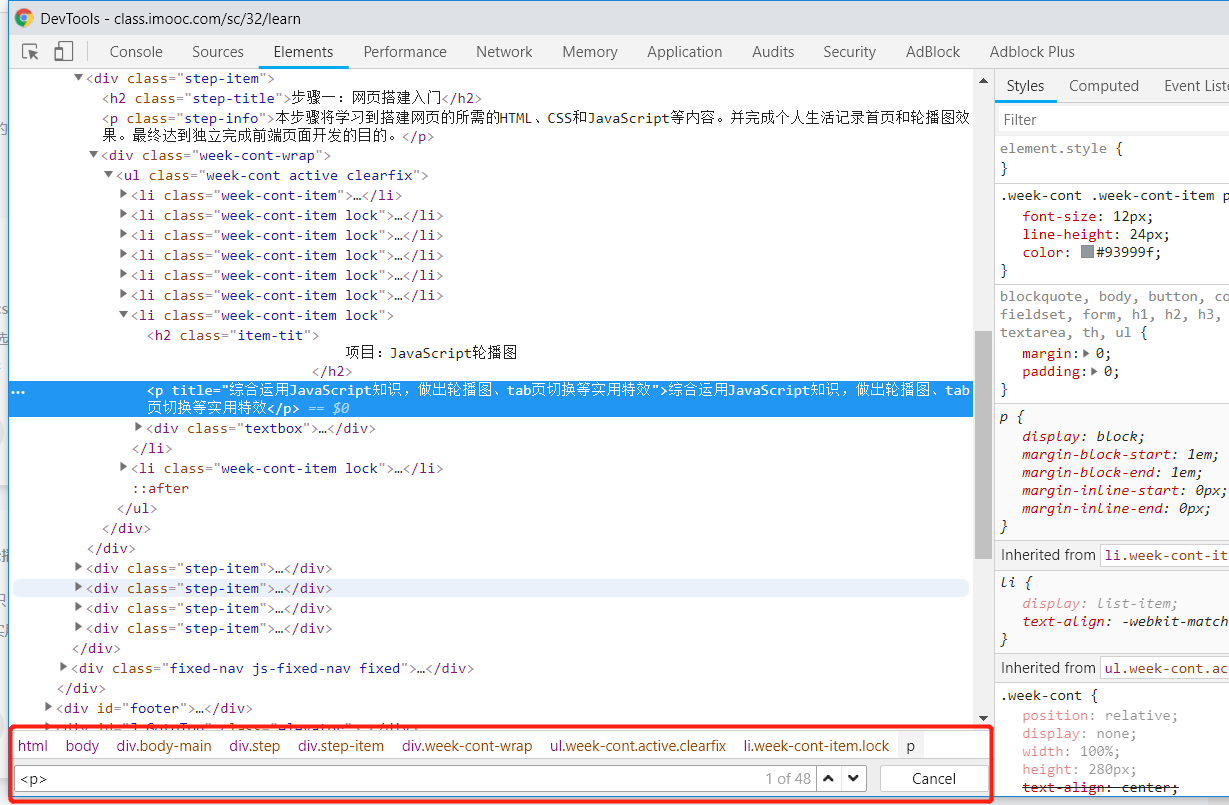


浏览器的默认样式是潜在的风险，会造成布局的不确定性。要考虑自己覆盖。

如下图所示，margin、border、padding位置关系一目了然。



如下图所示，打开后台，在Elements下，ctrl+f可以去寻找特定的标签或其他代码。



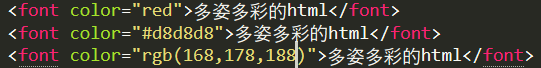
# <font>标签（不建议使用，可以用css进行设置）

<font>标签用来格式化文本。

size属性，设置文本的字体大小

face属性，设置文本的字体集

color属性，设置文本的颜色，支持三种表示方法：颜色字符串、16进制、rgb。

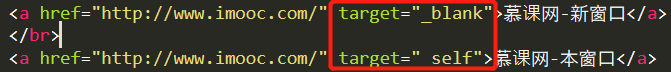


# 链接标签<a>

一，转跳到外部资源

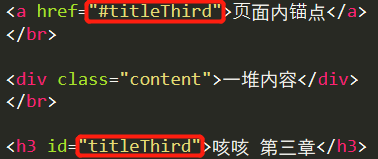
href（hypertext reference）属性，

target属性，如下图所示，\_blank表示在新窗口打开，\_self表示在当前窗口打开。目标地址都是是mooc这个网站，只是打开的方式不同。



二，设置页面内的锚点

如下图所示，页面会跳转到有id等于titleThird的标签的位置。



并且，在点击这个链接之后， 页面链接也发生了变化。

C:\Users\xuyang\AppData\Local\Temp\1550304071(1).png

技巧一，禁止跳转（死链）

在href中如此写，此a标签将没有任何作用。注意，绿色部分不写，也可以达到这样的效果。



技巧二，去掉a标签的下划线

注意：css设置颜色的时候，颜色值不需要写在””中

a:visited，为一个伪类，是CSS的知识。



a:visited中颜色的变化设置对死链不起作用（下图所示死链）。



a:hover，设置鼠标浮动在某个元素上时的样式

# <image>图像标签（超文本，意为不是文本）

和input标签一样，出现格式为，<image />

当你不给图片设置宽高的时候，则页面显示实际图片的宽高。

如下图所示，如果高度设置为””，图片依然可以显示，如果只设置一个属性，则图片等比缩放。

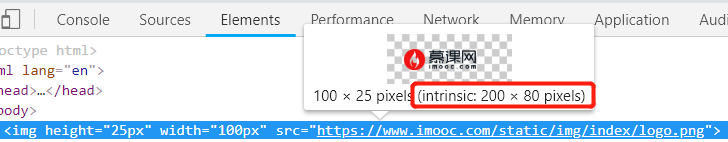


src属性，设置图片的路径，支持jpeg、png、gif、pdf（引入pdf必须是单页的）

如下图所示，

C:\Users\xuyang\AppData\Local\Temp\1550465659(1).png

如下图所示，在后台元素栏，鼠标悬停图片链接处，可以查询图片原来的大小。



alt属性，用于替换文本，当页面图片无法正常显示的时候，显示alt的值，有提示的意义。

技巧一，不通过<img />标签来插入图像

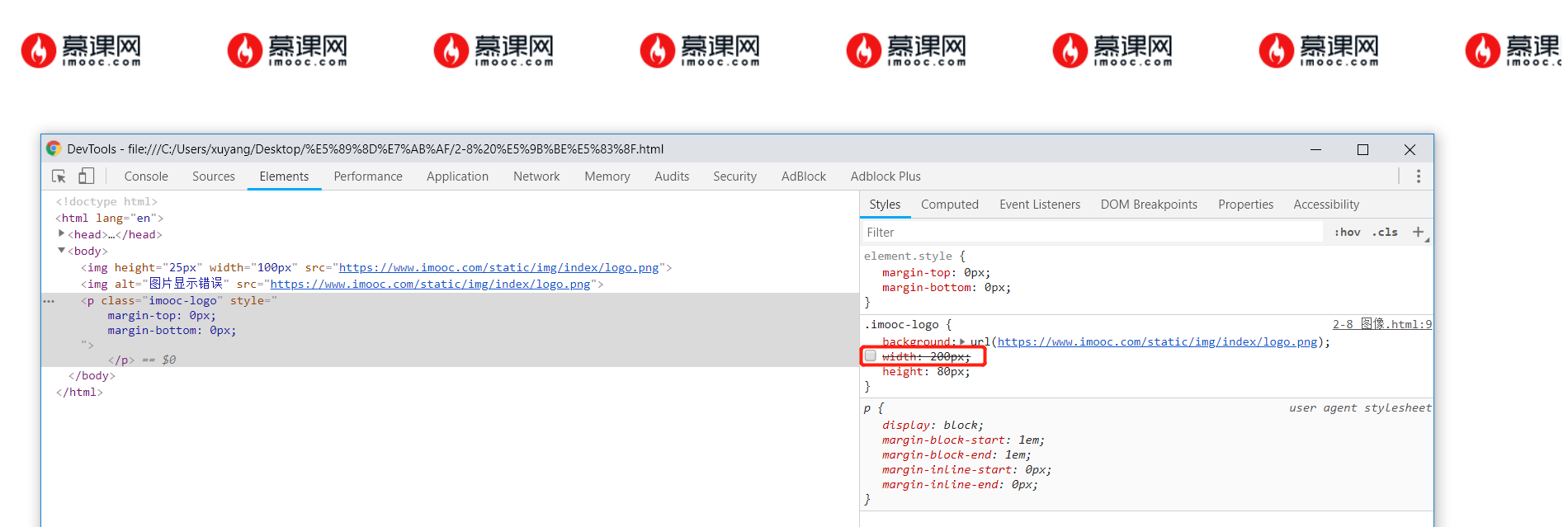
1，.mooc-logo是用来设置指定类名的样式的

2，如果没有设定类名，在style标签中也可以如第二个框对img标签的样式进行设定。同时body标签内对img标签样式的设定会被覆盖。

3，使用url()设定图像的路径



4，如果p标签不设置宽度，则效果如下所示。

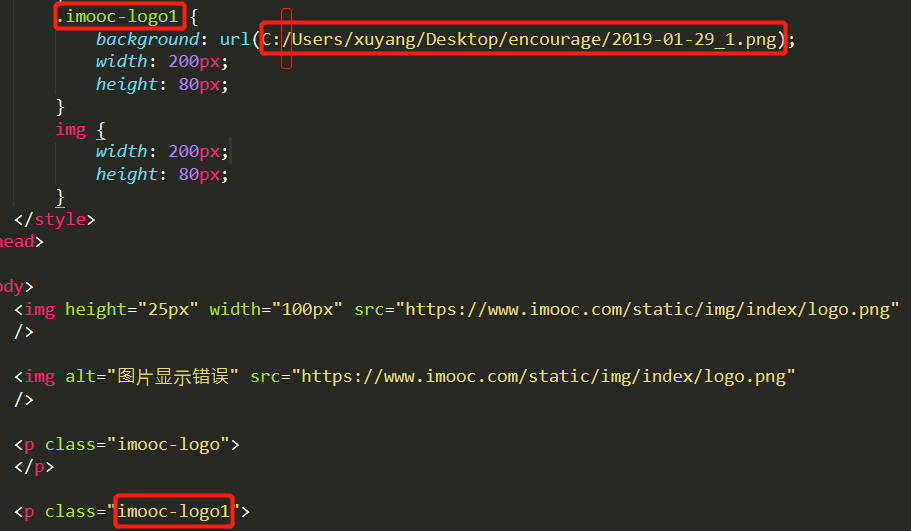


技巧二，插入本地图片

路径：描述资源的位置。

1，绝对路径

如下图所示，为应用绝对路径，注意：路径中使用/而不是\。

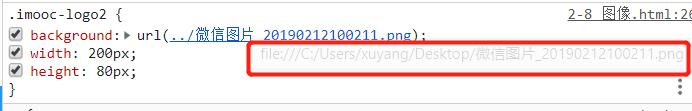


2，相对路径  
如下图所示，为使用相对路径

当前路径下索引文件使用./，回上一层路径../，回上上层路径../../



在后台，鼠标悬停样式中的路径上，可以看到所提示的绝对路径。

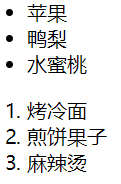
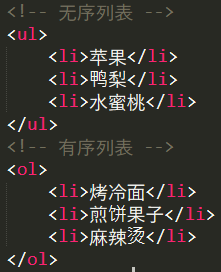


# 列表（建议只使用ul无序列表+样式）

无序列表

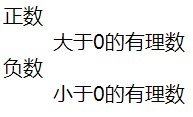
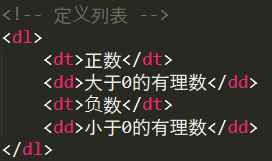
有序列表

代码和结果如下图所示



定义列表

代码和结果如下所示，<dt>中放概念的名称，<dd>中放概念的定义



type属性，用来设置列表前的符号（type属性设置的样式对有序和定义列表不起作用）（但是不建议使用，可以用CSS来进行代替）

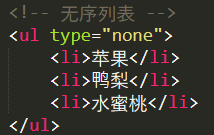
disc，实心圆

circle，空心圆

square，实心小正方

none，没有任何样式，但是会保留样式的位置

代码如下所示：



# <div>标签（万能标签，常用，重要）

这是一个没有语义的标签

<div></div>，用来进行布局

如果要显示的对象既不是图像也不是段落，可以考虑使用div

# 块级元素与行内元素

官方文档定义：https://www.w3.org/TR/CSS22/visuren.html#containing-block

块级元素，垂直上下方向挨着，上下块级元素的间距由margin属性决定。块级元素占满一行，多个块级元素在一起上下换行排列。

主要用来页面布局上，因此元素大小取决于开发者的设置。

注意：<p>元素是块级元素，是占满一行的。表格也是块级元素，内部的单元格却不是块级元素。

行内元素，通常被包含在一个块级元素中，（同一行）水平方向挨着。行内元素在一行内，一个挨着一个。

主要用来包裹内容，因此元素大小取决于内容。

如何记忆和识别这两类：

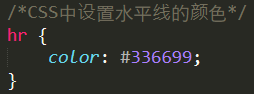
在没有CSS样式修饰的情况下，用开发者工具鼠标点击页面元素，如果占一行就是块级，如果不是则为行内。

# <span>标签（应用次数蛮多的）

用于在一个行内，把一段文本包起来。

# <hr>标签

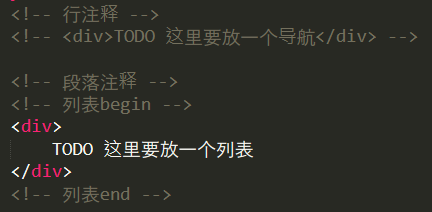
<hr> 标签在 HTML 页面中创建一条水平线。水平分隔线（horizontal rule）可以在视觉上将文档分隔成各个部分。如下为设置水平线的颜色。



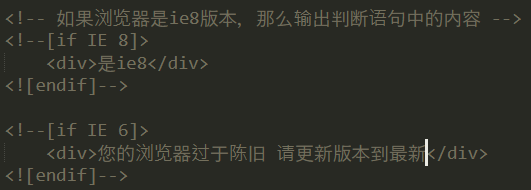
# HTML5注释

行注释，<!-- -->，必须闭合

段落注释，如下图所示，用于描述代码块的功能，如果要注释掉整段代码，可以把代码放到<!-- -->内。



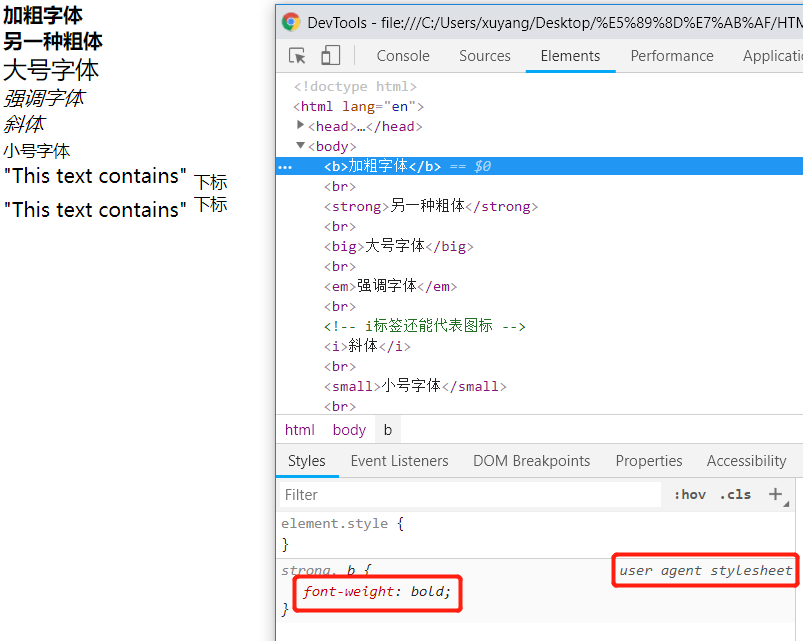
条件注释，兼容性检查必备（条件注释标签只有在IE10以下的浏览器下才能生效），是用于检查老旧浏览器的。提示用户更新浏览器或让老旧浏览器兼容当前代码或样式。如下图所示：



# 带格式作用的标签

1，文本格式化标签

你可以在Chrome开发者界面看到浏览器给予各个标签的默认样式，如下图所示。

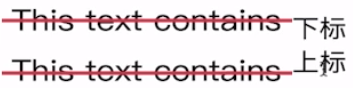


b和strong标签对应的是font-weight属性

big和small标签对应的是font-size属性

em和i标签对应的是font-style属性，注意：i标签还能代表图标

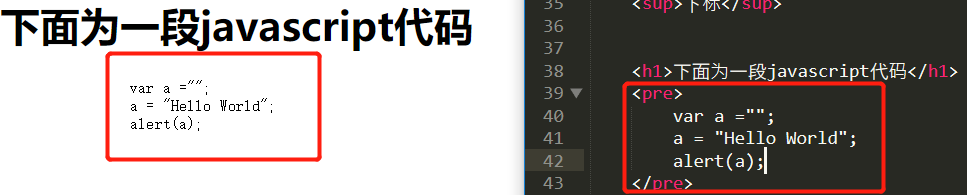
sub标签对应下标，sup标签对应上标，位置如下图所示。



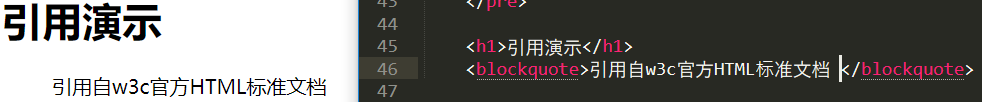
这些标签，我们一般是不用的，因为这些标签在各个浏览器下的表现可能不一致，我们一般用CSS样式控制字体的格式化。

2，预格式文本<pre>标签

<pre>是用来定义预格式化的文本。被包围在 pre 标签中的文本通常会保留空格和换行符。而文本也会呈现为等宽字体。经常在需要贴代码的网站中使用pre标签。开发者还会在CSS样式中对pre标签进行二次定制。



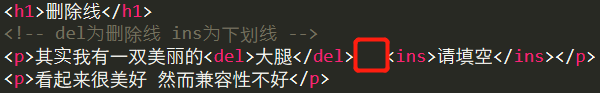
3，blockquote引用标签，告诉这段文字是引用过来的

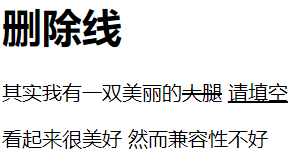


4，删除线<del>

注意，下图红色框内无论有多少空格，实际显示只有一个空格。如下下图所示。

删除线和下划线对应的都是text-decoration属性





由于兼容性的问题，一般还是会舍弃del或ins，转而使用CSS样式进行格式化。

# 表格标签（重要）

（应用广泛、且由多个标签组成）

标签格式，<table></table>

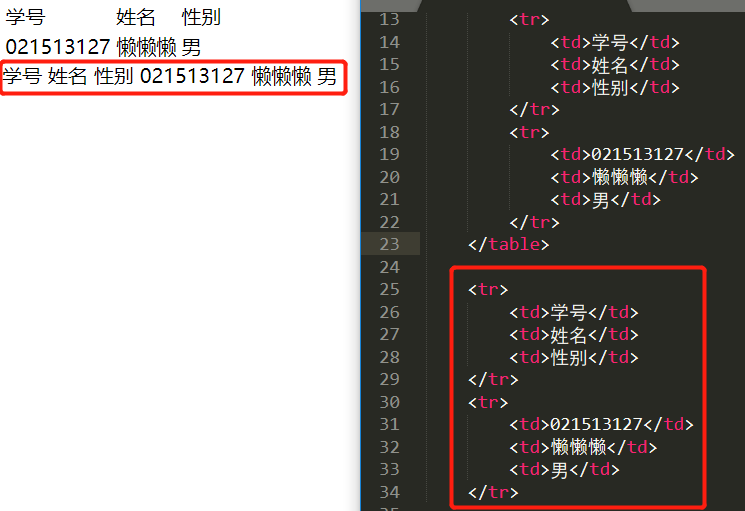
一般用来做一些布局，呈现需要表格布局的内容

表格是一个复合标签，由多个标签组成

行，tr

单元格，td，编辑td过程中，会自然地生成列（列不是由你定义的），相反tr（行）是需要你去定义的。

如下图所示，如果没有设定table，则如下结果排列。

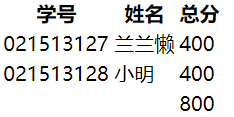
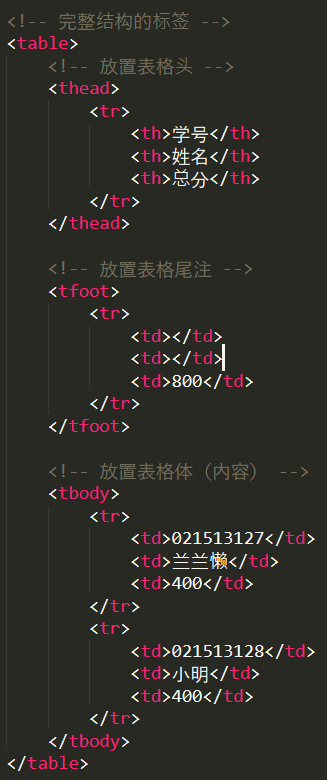


表格头，<th>，就是将第一行<tr>中<td>改成<th>，如下图所示。



除了带表格头和不带表格头的，还有一种完整结构的标签。即使在完整表格结构中，也是需要tr和td的。只是多了thead，tfoot以及tbody三个标签。

这种完整标签的好处是结构上控制得较为完整。并要求以thead，tfoot以及tbody的顺序进行编写。代码和结果如下图所示。

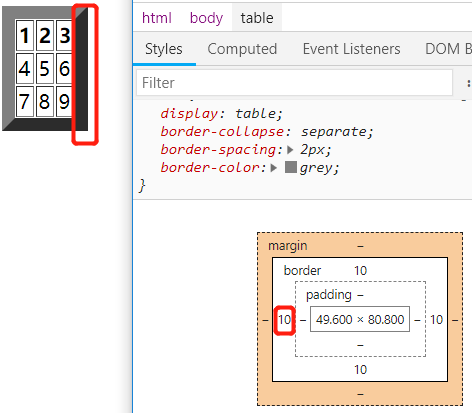


表格头和表格体只需了解即可。

相关属性：

border，用于设定表格外框的border，设置为0表示没有border，越大，表格的外框越大。如下图所示。开发中推荐使用CSS样式来管理边框。

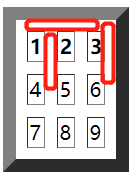
，直接利用表格属性，双引号（可加可不加（Chrome下））



cellspacing，单元格间距，设置为0，则没有间距。

C:\Users\xuyang\AppData\Local\Temp\1550750378(1).png

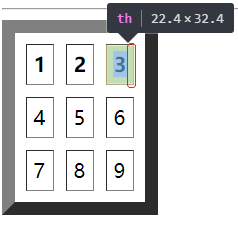
下图为设置为10的效果。



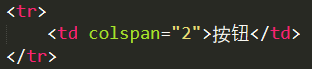
cellpadding，设置单元格的内边距，设置为0表示没有内边距。

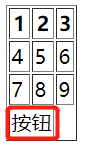


设置为5的效果如下所示

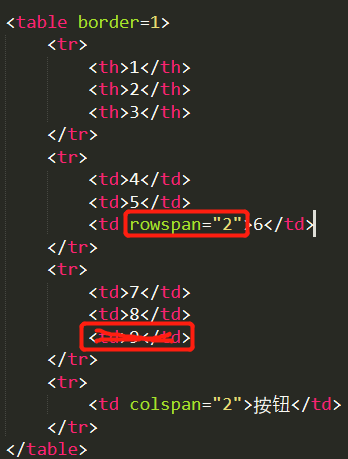


colspan，单元格跨列，想跨几列就设置为几。效果如下图所示。





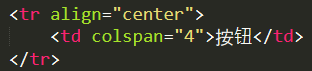
rowspan，与colspan一样设置和使用，要注意把下面的将要被合并的单元格删掉，不然会出现单元格被挤到外面的情况。

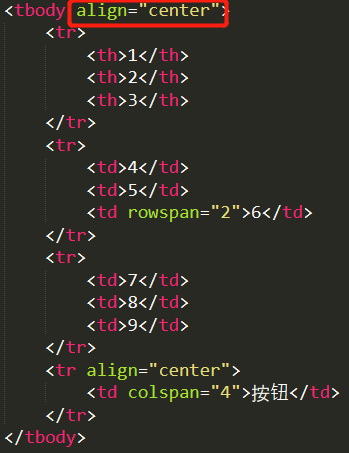
注意：rowspan是从上到下，colspan是从左到右进行合并

内容对齐，align，不是设置在单元格上的，而是在当前这一行上。（作用范围是这一行的所有内容）。

center是居中，right是右对齐，left是左对齐。

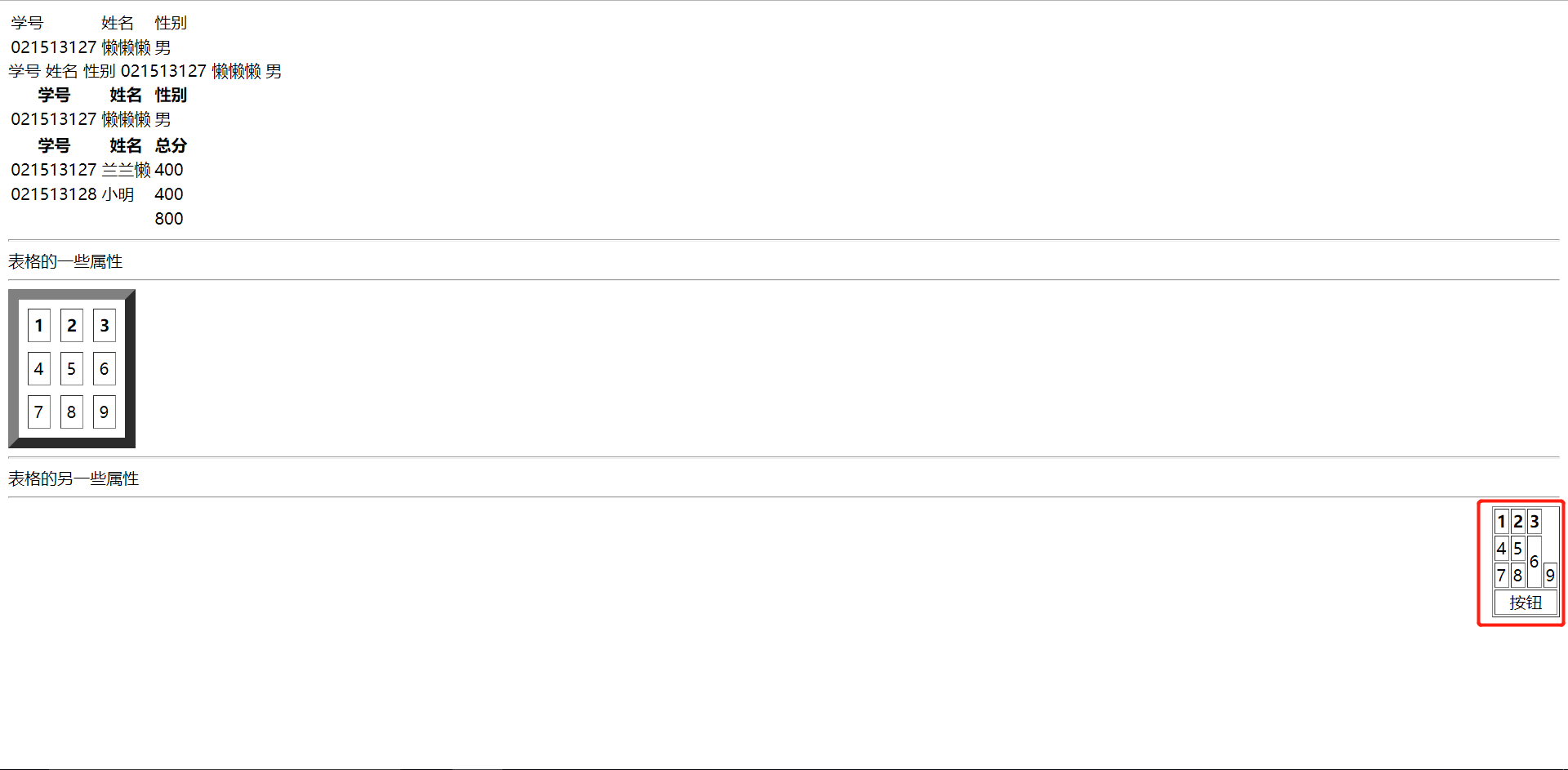


如果表格行数较多，可以将所有tr放在tbody中进行设置，对tbody设置align属性。如下图所示。因此，tbody可以用来便捷地管理整个单元格体。



如果将align设置在table上，则影响的是table在整个文档中的布局。





注意点：

表格标签是块级元素

全页布局的作用已经退出舞台了，（现在可以用div浮动）

专注于表格适合应用的领域，div也取代不了，div实现表格是有难度的

表格样式推荐使用CSS来定义，很少使用自身所带的样式属性。

如下图所示，可以设置表格的宽度：



如下图所示，可以设置表格的背景颜色：

背景颜色使用<table>标签的bgcolor属性进行设置，bgcolor为背景颜色属性，也可以在单元格或<body>标签等内容中使用。



如果只有thead和tbody标签，没有tfoot标签，也没有关系。（至少在Chrome下不会报错）



# <form>表单标签

表单标签是也是一个复合的标签，（和table标签一样），由多个标签组合来使用。表单元素必须放在<form></form>内，才能有效果。

作用是：早期是呈现给用户内容的，慢慢变成收集用户输入的内容。

form内可以放的表单元素有：

input，输入框（也可以认为是表单元素），类型有很多（通过type属性进行设定）：

1，文本（常用）text，

文本框的提示文字使用placeholder属性进行设置，如：<input type="text" placeholder="我是文本框">（当type为密码时，也可以这么设置）

文本框宽度设置使用size属性，如下图所示。



2，密码password

3，单选radio（需要自己加入name属性完成单选效果），value属性必须设置，设置选中之后的单选组合的值。  
4，多选checkbox，也需要自己设置value，来设置选中之后的值。checked，属性设置默认被选中（单选也有此属性）。

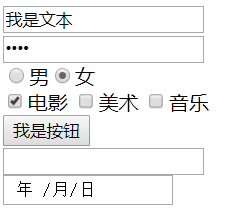
5，按钮button，它的value是按钮的显示文本；还有一个特殊类型的button，需要和form结合应用。

6，数字number，HTML5中新加入的类型

7，日期date，HTML5中新加入的类型，日期控件很少去使用，因为它在不同的浏览器中的表现不一致。一般借用第三方的库去使用。

1～7代码和效果如下所示





8，颜色color，点击会出现颜色选择器，由于浏览器兼容性的问题，所以也不是很常用。

9，范围range，

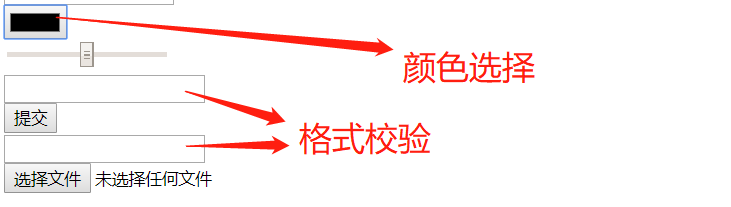
10，邮件email，和password一样，比较早就支持了。要和submit组合使用才能体现作用。这时，配合submit使用，浏览器会对邮件控件内的内容做格式校验。

11，url和email类似，也只有配合submit，才能体现作用（格式校验）。

12，文件file，multiple（可以用那选择多个文件）

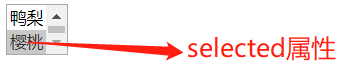
如下图所示，为相应的代码和效果。



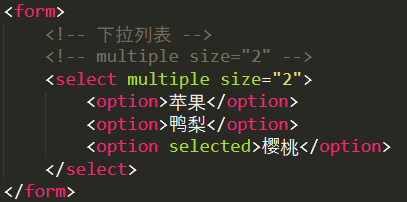


1，下拉列表select，select是需要闭合标签的。格式为<select></select>。select是一个组合标签，需要其他标签<option></option>协同产生效果。

option标签中的selected属性设定下拉列表的默认值，如下图效果和代码所示，设置樱桃为默认选项。



如果设置属size=”2”，则下拉列表会显示两个选项内容，如上图所示，multiple属性设置下拉列表为可多选。



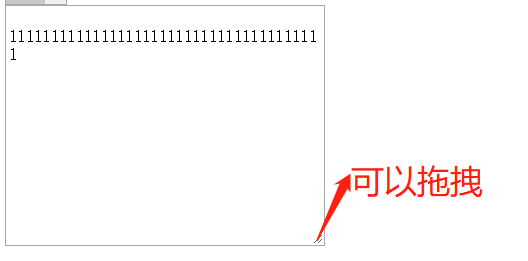
注意错题：

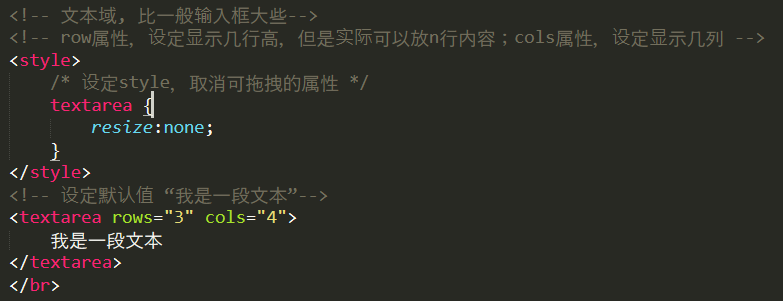
<select>标签中设置了multiple或size属性的话select是列表了，不是下拉菜单，不设置是下拉菜单。



2，文本域textarea，格式为<textarea></textarea>，可以多行显示，当你要显示多行信息时，可以选择文本域。

重点属性，rows和cols，设定rows和cols后，如果拖拽，则会破坏原来设定好的rows和cols。





3，按钮button（和input标签中的button类型是有区别的），格式为<button></button>。

重要属性，

type=”button/submit/reset”，

当type=”submit”时，如果<button>在form外面，此时根据button的form属性设定提交的表单是哪一个。button的form属性的值和要提交的form的id要一致。

当type=”reset”时，会重置button标签所在的form标签内的所有表单元素的值为默认值。



其中，文本，密码，单选，多选，按钮，范围，邮件，提交，文件，下拉列表最为重要

以上讲的都是form表单的构成标签们，下面是关于form表单的重要属性：

表单是如何提交的？提交到哪里？

1，action，提交到的服务端地址（提交数据给哪个路径去处理或者理解为前端的接口地址）

2，method，指定提交时用哪种HTTP方法（POST/GET），通常使用POST；当想告诉服务器，某个链接，用户每天点击多少次时（仅连通一下接口地址，本身并不传输什么数据）可以使用GET。

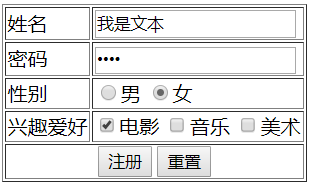
3，name，表单标识

4，autocomplete，浏览器是否可以自动填充（之前填过，下次填就可以自动填充（理解为记忆功能））

5，enctype，指定表单内容编码（encode type），默认为utf-8，这里的值其实是需要和服务端的开发人员预先协同好。

# 表格和表单的嵌套（表单和表格搭配最为常用，需要多做练习）

用表格来美化表单（用表格来做一个布局）。代码和效果如下图所示：



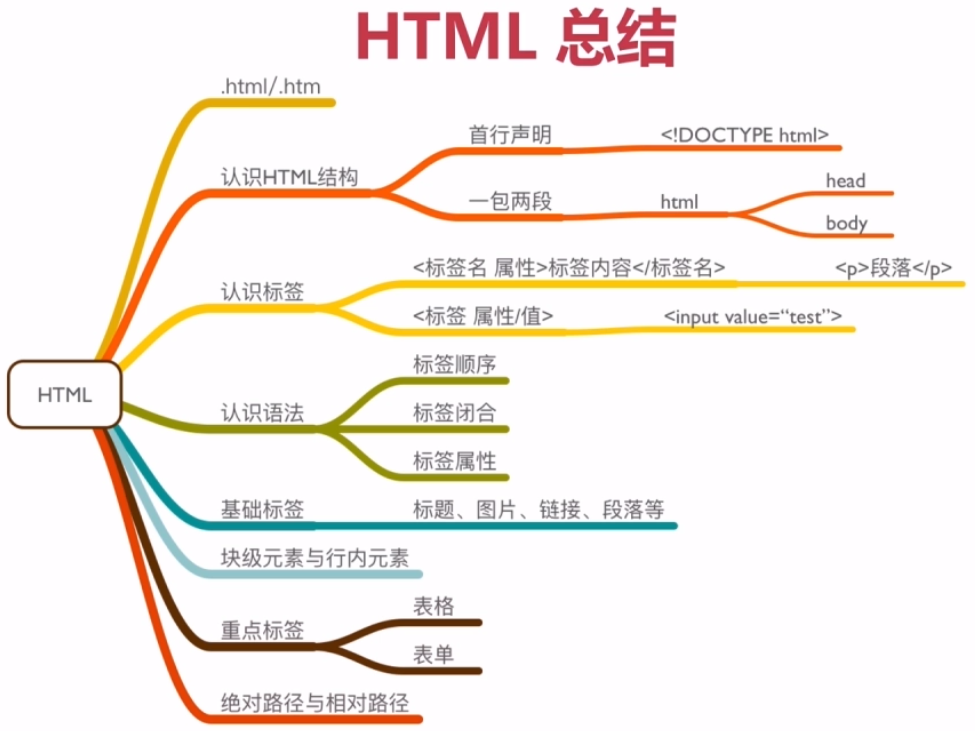


# 空格

HTML5中的空格表示：&nbsp;意思是no-breaking space

# 总结

可以根据如下思维导图来回顾HTML的学习。



# CSS

# 什么是CSS？

层叠样式表（用于制作炫酷的网页效果）Cascading Style Sheet。CSS应用于HTML页面中的元素在浏览器中的样式。样式通常被存储在样式表中（表明CSS本身还支持独立的文件，扩展名.css）。

# 为什么要使用CSS？

1，html文件在浏览器中运行时，浏览器对html页面中的标签进行解析，并将解析之后的样式呈现给客户。但是不同浏览器对同一标签的解析效果会有差异。为了解决这个问题，CSS出现了。它主要解决让同一页面（元素）在不同浏览器中呈现同一样式。

2，CSS样式定义如何显示html标签内容【可以理解为与第一点相同】

3，解决内容和表现分离的问题（低耦合）

内容：主要指html中的标签（运行一个页面时，用户看到的内容都能称之为内容）【HTML标签】

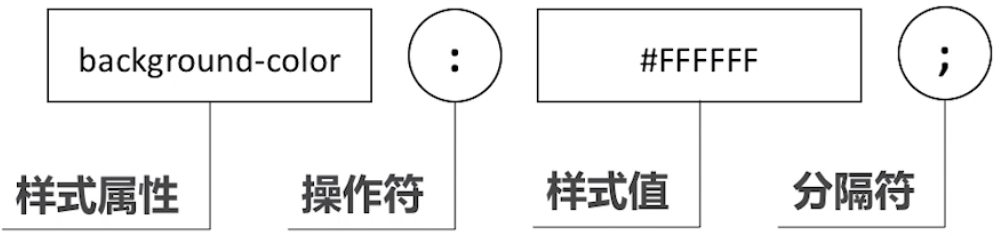
表现：相同的内容在不同浏览器中呈现出来的样式有可能是不一样的（表现主要指内容在浏览器中呈现的样式）【CSS】

CSS和HTML是低耦合的，它将内容和表现进行了分离。

# 基础语法

# CSS声明

声明由四个组成部分，样式属性、操作符、样式值、分割符。如下图所示：



样式属性：由CSS提供（固定的、不可变更、实际开发时，自己去选择合适的样式属性就好了）

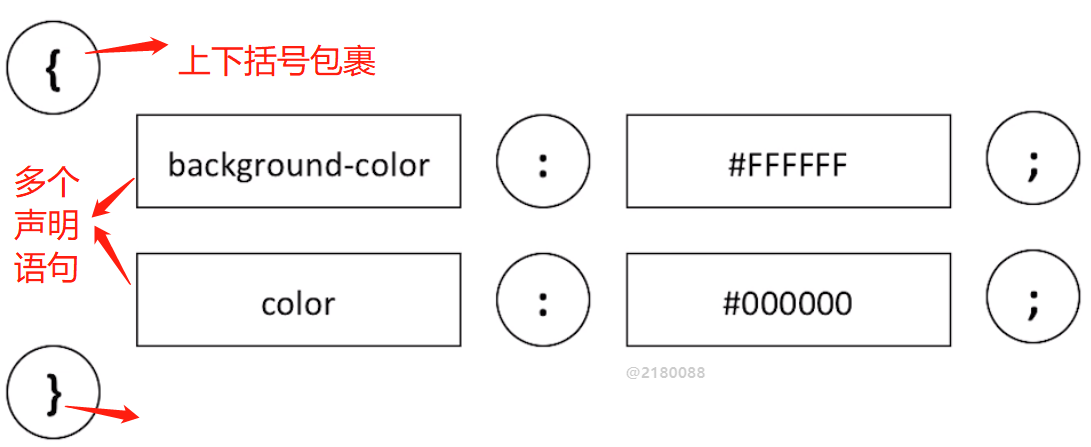
操作符：须为英文冒号

样式值：样式值针对样式属性，（不同样式属性对应的样式值有可能不同，例如，背景颜色属性对应的一定是和颜色相关的样式值）

分割符：须为英文冒号

为某一个或某一类标签定义样式，且使用到*多个*样式属性，这个时候怎么办？

引入CSS声明块（由多个CSS声明组成），如下所示：【相比较CSS声明来说，CSS声明块好处是可以设置多个样式属性】

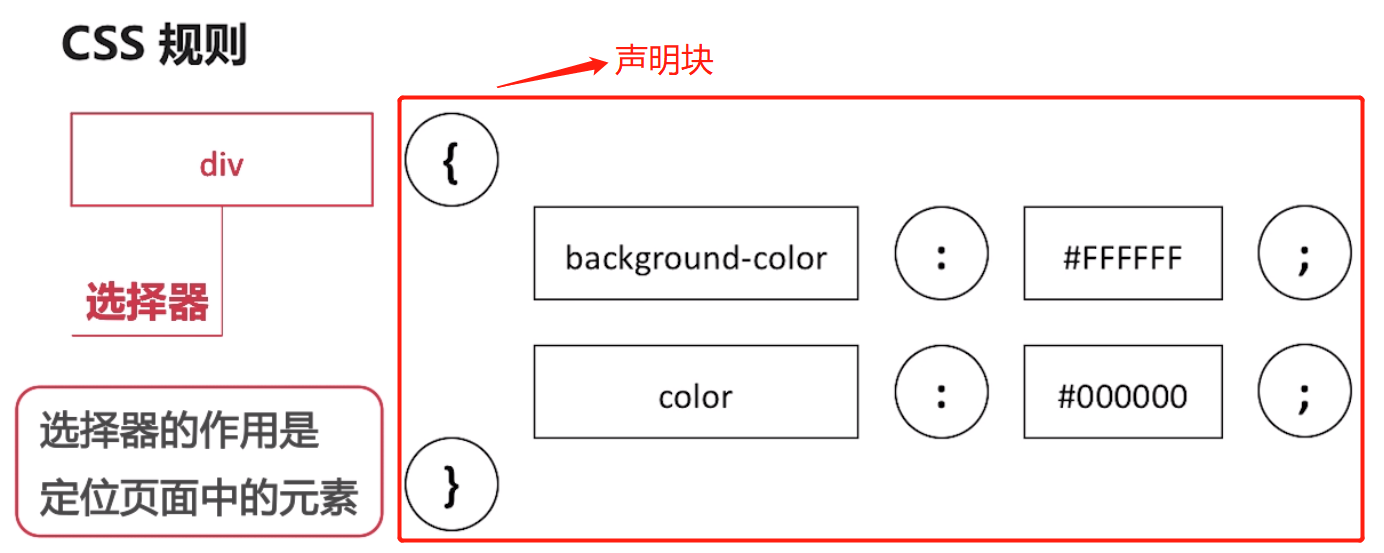


将CSS样式与HTML元素进行关联：

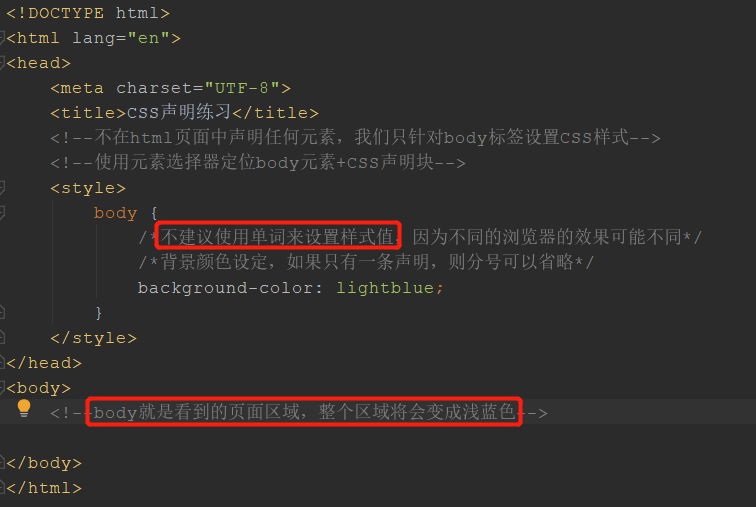
CSS规则，包含CSS选择器和CSS声明块。

CSS选择器，用来定位页面中的标签（一个标签或多个【也就是一类】标签）。CSS规则概览如下图所示。

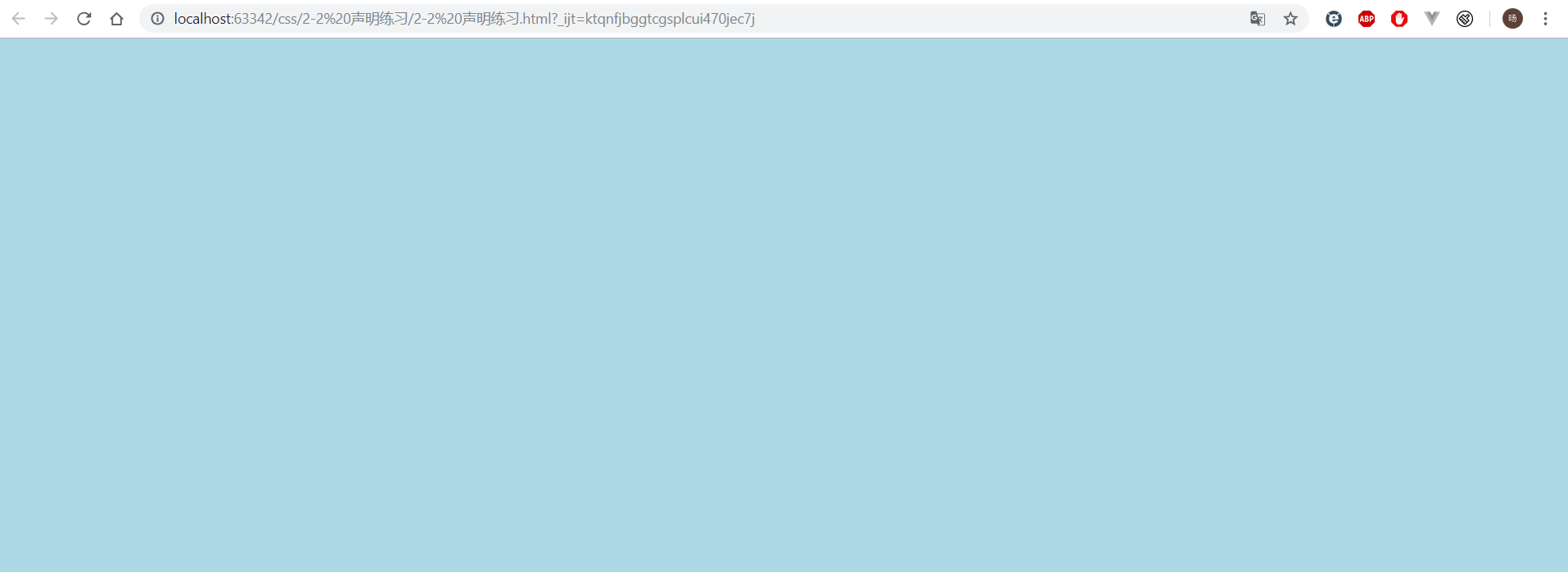
用CSS选择器进行元素定位（在这个说法里使用了元素选择器）



如下例子是对body的背景颜色进行CSS设置。下面分别是代码和效果。



由此也能看到body的作用范围。



# CSS内联样式

之前学习的是CSS声明，在这里则学习在HTML页面中，如何定义CSS。

为什么要定义CSS？（定义CSS可理解为使用CSS？）

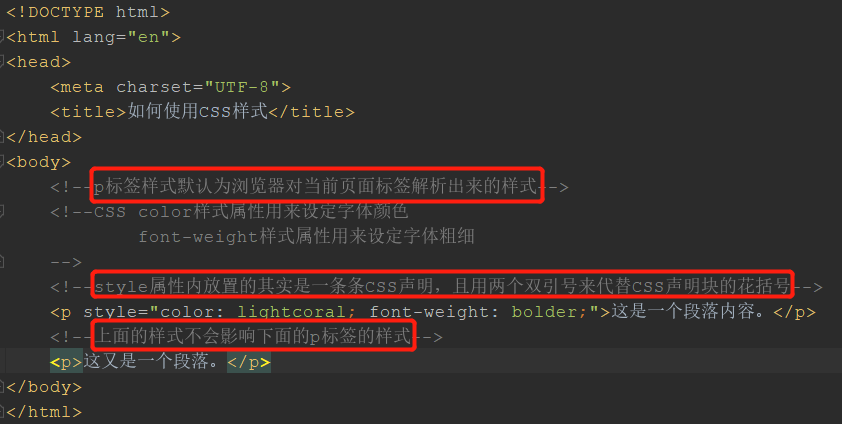
两种方式：

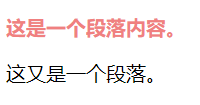
1，内联样式方式（定义：通过style属性实现）

2，外联样式方式

内联样式方式：利用HTML标签的通用属性style，用style属性对当前标签定义CSS样式。样式内容（CSS声明）就写在属性对应的值内。

代码和效果如下所示：





内联样式方式，会有如下问题：

1，作用域仅为当前的标签。如果要对一类元素设置相同样式，会带来代码的重复、冗余性。

2，内容（HTML）和表现（CSS）的代码耦合度过高（代码都缠在一起了，内容和表现并没有分离）。

# CSS外联样式

使用方式一：

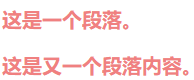
在<head>元素中定义新的元素<style></style>，在<style></style>内设置CSS。

表现和内容被有效分离了。

样式作用于所有相同标签。

代码和效果如下所示：





不足：<style>标签中的内容属于CSS内容，其他的都属于HTML内容，CSS样式内容被嵌入HTML页面中去了。样式和内容分离不够彻底。

使用方式二（优化）：

利用独立文件。

选择器

多种样式属性

盒子模型

行距，前一行底线到下一行顶线之间的距离

font-size，同一行中，顶线到底线之间的距离

line-height，前一行基线到下一行基线之间的距离

行距=line-height减去font-size

=====================================================================================

H5+CSS

H5中，<body>标签默认自带8px的外边距。

placeholder，设置输入框的默认值

div容器外边距的厚度不被计入div容器的高、宽中。

div容器内边距的厚度被计入div容器的高、宽中。

CSS外部引入使用<link>标签

样式颜色设置不对，可能导致字体实际存在，但是现实不出来。

注意点：给div设置宽度会导致宽度写死，会无法适应较小的屏幕，同样的，给任何元素设置高度（宽度）也不是好主意，要慎重。

ECMAScript

相关输出代码（JS语言最基本的）：

alert，页面告警（执行的优先级会变高）

document.write，往<body>中输出内容

console.log，控制台输出内容

注释（JS语言最基本的）：

多行注释，/\*\*/

单行注释，//

写在<script>标签中的注释可以通过网页查看源码阅读

代码执行顺序：从上往下

是否严格区分大小写：是

每条语句以；结尾（如果不放；，会消耗浏览器的资源，有时浏览器会加错分号）

字面量（常量）

变量

变量的声明，var，声明只需要一次就行。

变量的定义（赋值），var a = 10;

标识符规则（语言基础）

字母、数字、下划线、$

标识符不能以数字开头

标识符不能是关键字或保留字；

标识符推荐采用驼峰命名法

数据类型（语言基础）

（字面量类型）

String Number Boolean Null Undefined（前五个基本数据类型） Object（引用数据类型）

String，用引号””（单双引号都可以）引起来，但是引号不要混用，引号不要嵌套。单引号内不能放单引号，双引号内不能放双引号。

转义：\

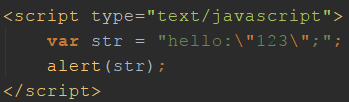
\”，表示”

\’，表示’

\n，表示换行

\t，表示制表符

\\，表示\



Number类型，包括整数和浮点数

JS可以表示的数字的最大值，Number.MAX\_VALUE，值为1.7976931348623157e+308，如果超过此值，则返回Infinity或-Infinity。

最小值为，Number.MIN\_VALUE，意义是大于0的最小值。

其中注意：Infinity为字面量。Infinity的类型是Number，可以用typeof检查。

NaN是一个特殊的数字，表示Not a Number，它的类型是Number

typeof，用来检查变量的类型

String类型，返回string

Number类型，返回number

如果使用JS使用浮点数运算，可能得到一个不精确结果。如下图所示：（故不建议使用JS进行精确度较高的运算）

进制：

0x表示16进制数

0表示8进制数

0b表示二进制

Boolean值，只有两个值，true和false，主要用来逻辑判断

Null类型，就是null。用来表示一个为空的对象。它的类型是对象。

Undefined，当声明一个变量但是没有赋值时，变量的值就是undefined。类型为undefined。

类型转换（*基础*）

转换String（两个方式，一个调用函数、一个调用方法）

方法一：调用被转换数据类型的toString()方法，但是这种方法不会影响到原变量。注意：Null和Undefined没有toString()方法。如果调用，会报错。

方法二：调用String函数，对于number和boolean也是调用toString()方法，但是对于null和undefined就不会调用toString()方法，而是直接将null和undefined直接转换为字符串。

转换Number

方法一：调用Number()函数，字符串转换为数字，如果是纯数字的字符串，则直接将其转换为数字。如果字符串中有非数字的内容，则转换为NaN。如果字符串是一个空串，或者是一个全是空格的字符串，则转为0。

true会转为1，false和null会转为0

undefined会转为NaN

方法二：parseInt()、parseFloat()，将字符串中有效数字解析出来。

parseFloat()，123.11px转为123.11

parseInt()，123px转为123

注意，如果对非String使用parseInt()或parseFloat()，它会先将其转换为String，然后再解析操作。

注意，parseInt(参数1，参数2)，参数2用来表示进制数。

语言的关键字和保留字（*基础*）

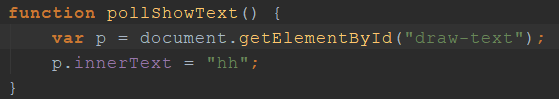
JS代码可以写到<button>标签的onclick属性中

JS代码可以写到<a>标签的href属性中

（但是最好不要将JS代码写到标签中，这样代码的耦合性会变高）

JS代码推荐使用<script>标签来外部引入，这样也顺便可以利用浏览器的缓存机制

如下两幅图所示，利用onclick事件来转跳执行函数内的代码，以实现设置p标签这一功能。





$(function(){});，JS代码中立即执行的函数

JS代码中可以写html代码，注意当html代码中有安排JS函数时，特别是这个函数有参数时，要注意html的代码的写法。

如下代码所示：<tr id=' + ret[index].name + ' onclick="showDetail(\''+name+'\')">



.submit()方法仅提交form标签中的所有内容

常用：

字符串查找，indexOf

字符串分割，.substring(start, stop)，内容是从 start 处到 stop-1 处的所有字符

返回页面中特定id的内容，.innerHTML

清空页面，$("#name").html("")

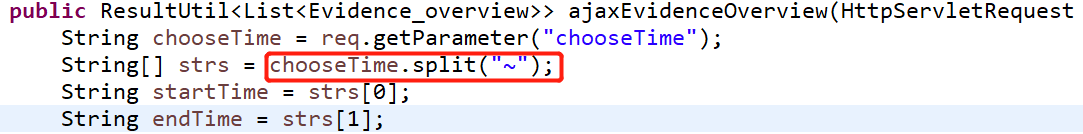
创建特定id的元素，document.createElement

元素中添加元素，.appendChild

Java

Java代码是运行在服务器端的，在服务器端代码里所描述的请求，就是页面对服务器的请求。

字符串分割



解决的问题：

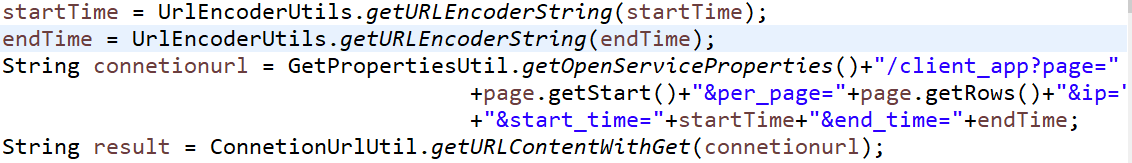
如下图所示：

startTime和endTime被编码为UTF-8格式后，请求才能成功。

e.g. startTime

原数据值为2019-01-26 00:00:00

转换之后的值为2019-01-26+00%3A00%3A00



nari-adf中的右侧导航栏的文字是如何显示的？

servlet.xml中设置了拦截器，如果是非法的url的话，那么会自动转到登录页面上。

前端部分的项目全部运行在服务器上，layout.jsp会获取从后台（LoginController.java和MainController.java）获取数据

nari-adf前端数据表格的数据数量如何传递给前端的？

后台Application\_recognitionController.java获取后台给到的json格式的数据，获取数据总数，随后将数据返回给前端。layui的框架自动处理了数据量，并显示出来。

# 如何添加一个页面？

newWeb.jsp

newWeb.js

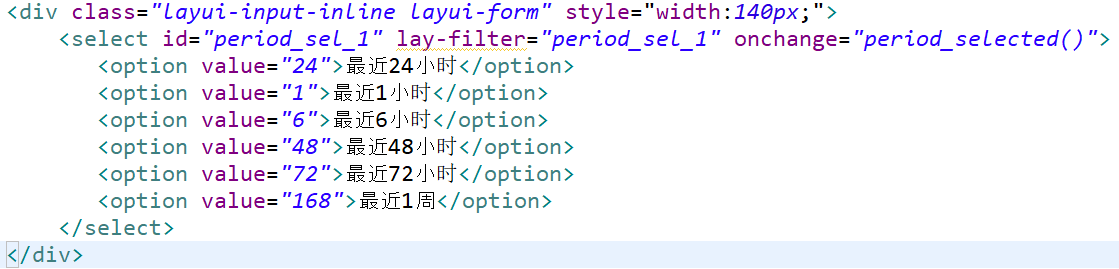
newWeb.class

修改数据库的三个表格menu、privilege、role\_privilege（role\_privilege可以根据role表的信息来添加）

注意：变量名较长，小心写错。

nari-adf添加时间选择下拉框的layui样式？

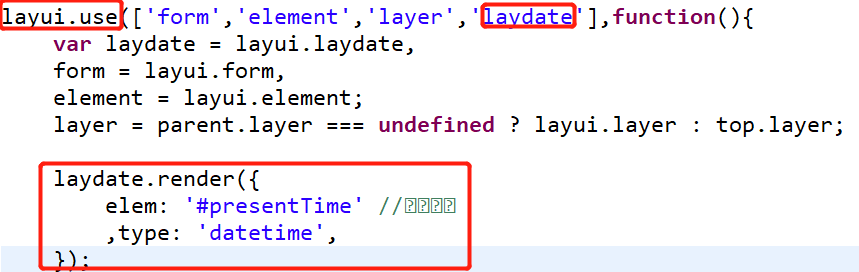
如下图所示，请加入layui-form这个CSS类。



JS修改之后，浏览器可能依然会保留着修改之前版本的缓存。请勤清楚浏览器缓存。

有一种报错：可以移除工程后，接着清除服务器。

使用layui的时间日期选择框，必须要按照如下步骤进行渲染



jsp页面间数据的传递方式：

通过后台传递：

index.js中把当前时间presentTime的值传给java后台



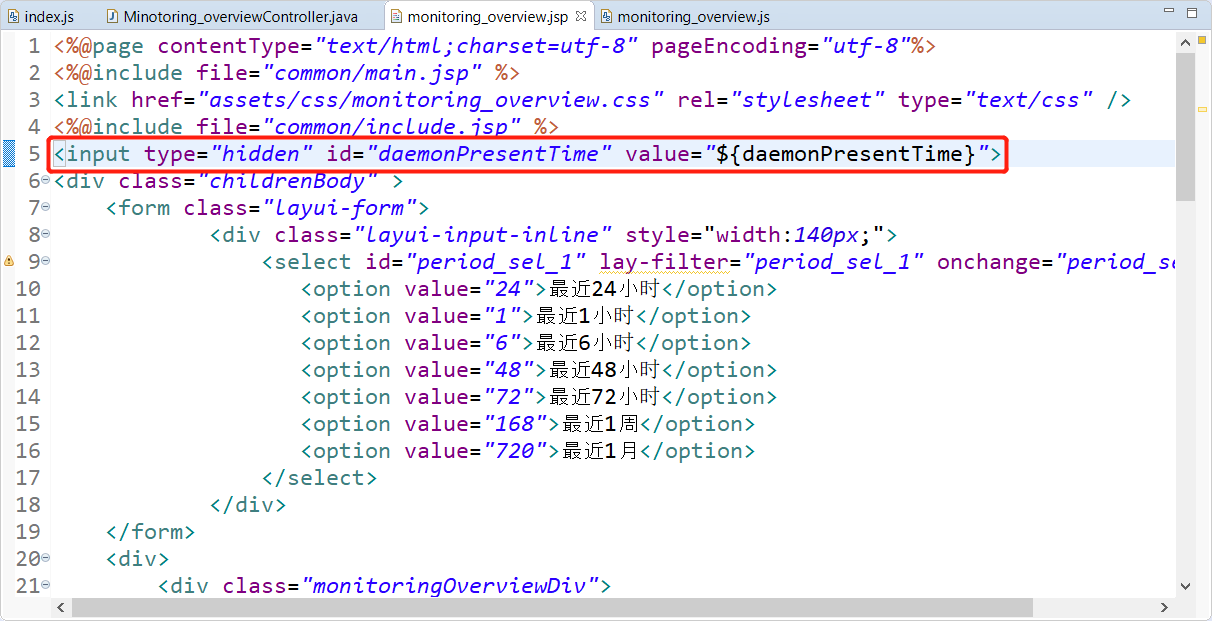
url的值如下图所示，这个整段对于java后台来说就是一个request



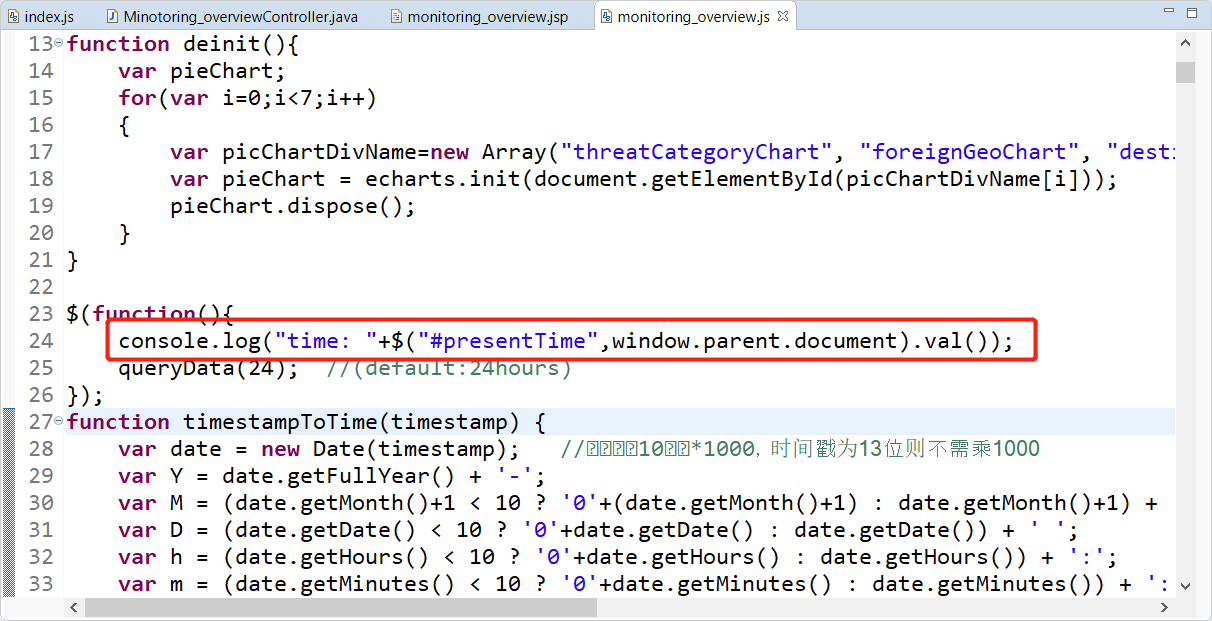
根据url的内容，找到符合路径的代码块，使用getParameter方法来获取presentTime这个参数。用setAttribute方法来设置jsp页面中EL表达式为daemonPresentTime的元素的值。（url中？后面的才是参数的开始，如果有更多的参数，则需要在前面加上&f）



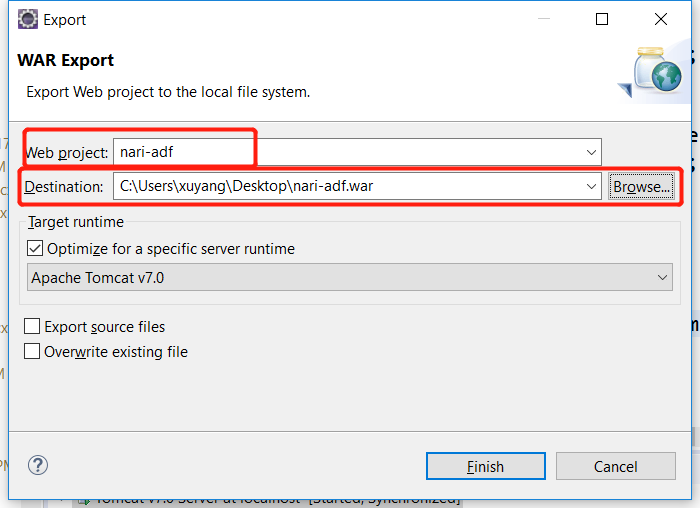
在jsp页面中，设置一个属性为hidden的input标签，然后通过java后台（上一步方法）设置这个标签的值。随后可以通过js来获取这个从其他页面传过来的值。



如果发生值传递的页面是父子关系，则可以用如下方法：

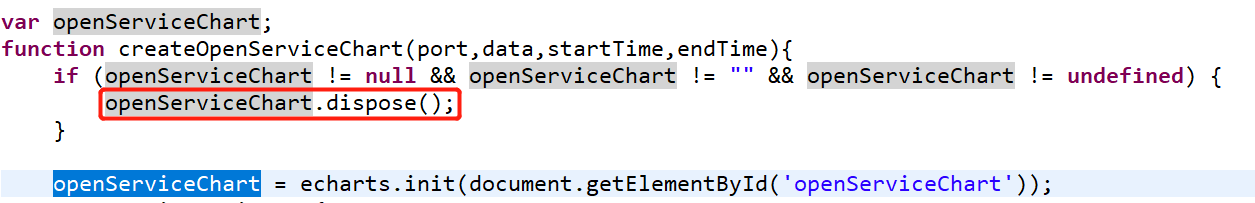


war包导出



layui内部渲染只能允许有一个form.on在代码中

echarts刷新，在初始化之前需要丢弃

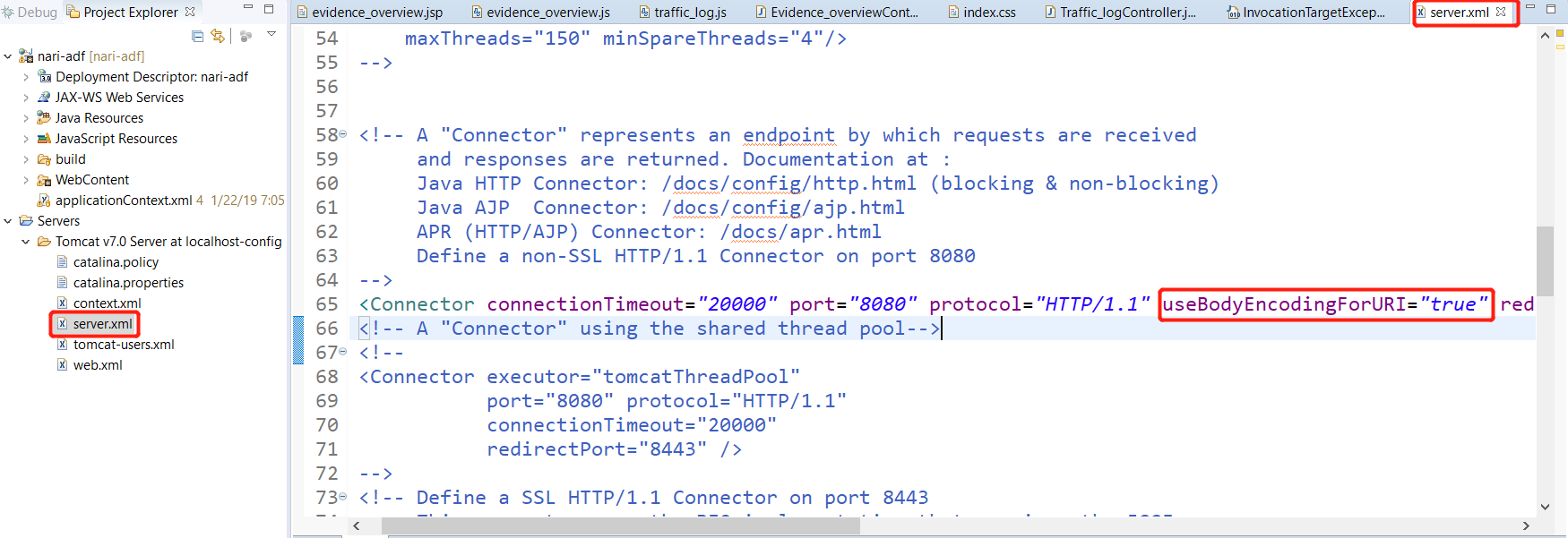


如下图所示，page用来存储分页相关的数据。



项目先从前端获取获取参数，然后再和后台（python那边）获取数据，随后再返回给前端。

配置服务器属性，修改页面乱码的方式



去除字符串中的空格

value = value.replace(/\s+/g,""); // exclude space

浏览器缓存和服务器缓存要经常清理。有一部分错误出错，很多时候因为没有去清理。