# HTML

## 第一课：HTML介绍

* 前端技术：

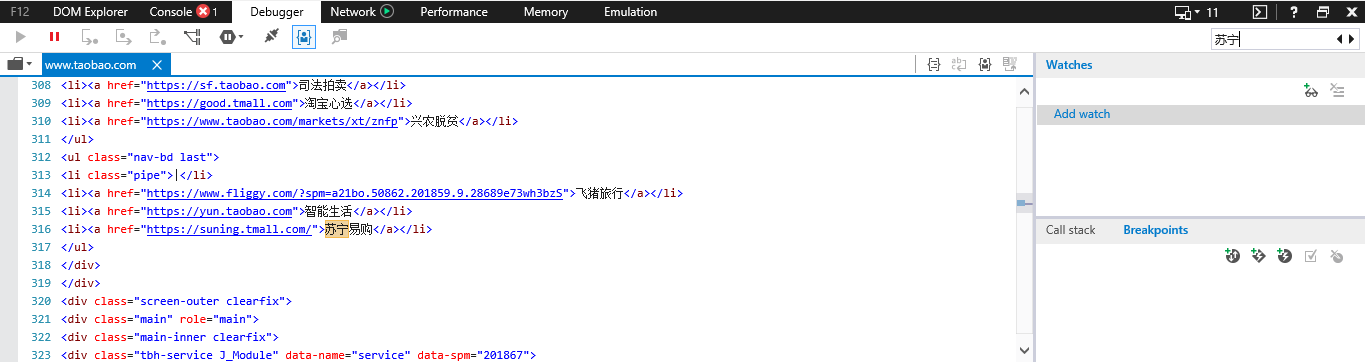
Html： 负责网页楼梯**结构**。网页文本内容

CSS：**样式**。网页图片内容

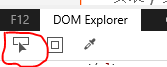
Javascript：**行为**。动态内容，互动内容

要求三者相分离。

* Html是用于网页前端编程语言，全称是 HypeText Markup Language 超文本标记语言；
* Html语言是由一对对具有特殊功能，语法含义的标签组成，标签组成： 头标签<html> 尾标签</html>——根标签，所有的标签都是在根标签上长出来的，一个html只有一个html标签。只能在这里面编程。
* 每个标签都有属性，属性必须写在头标签中。标签的标志两个尖括号<>。
* 头尾标签是用来修饰其内部包裹的文本内容的，需要成对出现。
* 某些可以自己代表功能，不需要成对出现，如 <meta> <br>*回车*，<hr>*水平线* 不需要包裹或者修饰啥。——单标签



* IE 浏览器中，右键 View source或者F12可查看网页源代码，右侧的find栏和select可找到对应Html代码,



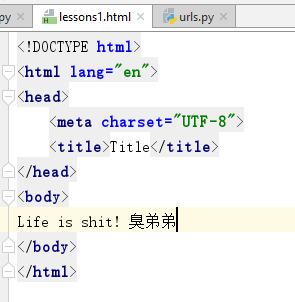
//取色器可以查看所指位置的配色

* <div>\*2 = 2对 <div> </div>

## 第二课：基础标签

* <html>…</html> 标签里必须有两个结构化标签：<head> , <body>
  + <head> </head> 头标签： 编辑浏览器特性的，与浏览器沟通的，人看不到的。

如 编码字符集，便于爬虫搜索的特征标签等



**Ⅰ** 如给浏览器设置字典：

<meta charset=”UTF-8”> :

1. meta 意为源代码，此标签可以单独存在，属性写在<meta 后>，写法：<标签名 *属性名 = “属性值“* >

Eg: <html *lang*=”en”> ，直译是”语言=英文”，告诉搜索引擎爬虫，这个网页是干嘛的，更方便抓取网站。与百度排名搜索结果有关，使网页排名更靠前。

2. 属性chaset 意为编码字符集，相当于系统的字典。常见的编码字符集：gbk, utf-8

3. gb2312 – 国家标签2312条，包含所有**亚裔**字符集（除了繁体字符集） 后来出台了包括了繁体字的扩展版本：Gbk。

4. Unicode——万国码，所有国家语言都包括。UTF-8是Unicode 8bit的升级版，最通用的所有国家的字典。

**Ⅱ** <title> </title> 页面角标设置：<title>淘宝网，淘！你妹</title>



* + <body></body> 表皮标签：写到主页面上可以被人看到的。

**常见标签：**

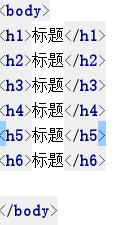
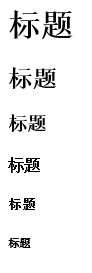
**1.** <p> </p>标签：段落标签，让标签内的内容呈段展示，paragraph

**2.** <h1> ~ <h6>标签：标题标签，6级标题。功能特点：

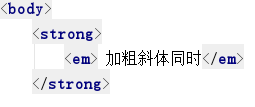
1. 独占一行独成段

2. 更改字体大写

3. 加粗字体

**3.** <strong> </strong> 加粗展示标签内部；<em> </em> 斜体展示

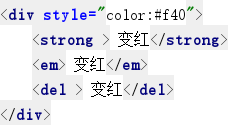
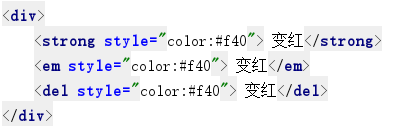
**4.** <del> </del>中划线标签



**5.** <address> </address>地址标签; 功能特点：斜体、成段展示独占一行

**6.** <div> </div> <span> </span>容器标签，分块功能，给页面分块分解的，使页面**结构化**；div功能特点：成段展示独占一行。——使用最多的

6.1 可以对容器进行打**绑定化**操作，其中的内容只用对容器进行操作。



打包操作

6.2 <div>属性 id=””,<div id=”demo1”> text</div>，指定唯一命名，便于确定该div模块的位置

## 第三课：高级标签

**前言：**

* 文本中的空格、回车无法在html中显示，无论多少个空格只显示一个。

1. 这是因为空格、回车在html是文本分隔符，空格前后告诉系统是不同的单词

2. 中文的每个字都是自己的含义，不需要空格。如果给定一个范围，汉字可以自动适应范围，但英文需要空格告诉系统何处分行。

解决办法：

* Html编码： **& ；**

**&**nbsp**;** = 空格文本

**&**lt**；** = < （less than）  **&**gt**;** = > (great than)

可以用来区别标签记号<>

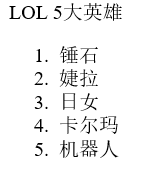
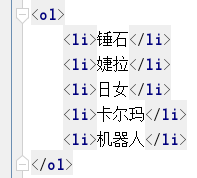
* 回车可以用标签代表： <br>, 可以单独使用，其本身就具有功能含义。很多br模仿页面滚动条。
* 某些可以自己代表功能，不需要成对出现，如 <meta> <br>*回车*，<hr>*水平线* 不需要包裹或者修饰啥。——**单标签**

**高级标签：**

* <ol> <li></li>…<li></li> </ol>，有序列表标签，用来按序号显示文本内容。Order list

<ol>是列表架子，类似橱柜

<li>是列表项，类似抽屉



* <ol type=”A”>头标签中可以定义排序显示类型的属性，总共5类：

1. 1，默认，列表项按照12345来排序

2. a，列表项按照abcd排序，27进制，后aa，ab，ac…

3. A，列表项按照ABCD排序

4. i，罗马字母，列表项按照Ⅰ，Ⅱ，Ⅲ排序

5. I，大写罗马字母

* <ol type=”1” reversed=”reversed”> 倒叙排列属性
* <ol type=”1” start=“121”> 如要从121开始排序，则头标签中定义属性。

<ol> </ol>用处基本不大。用的最多的是<ul></ul>，无序列表。

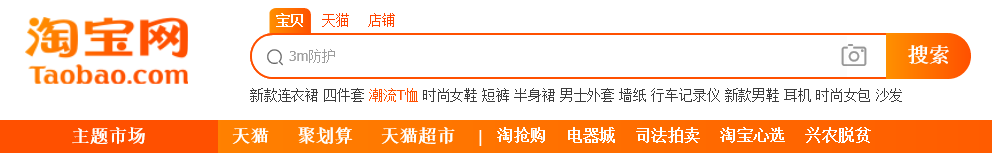
* + <ul> <li></li>…<li></li> </ul>，无序列表标签，只有一个属性可以更改 type。*Unoder list*
* <ul type=”disc”> 默认实心圆，discircle

1. type=“square” 方块

2. type=“circle” 圈

// 如果用css选择器： list-style:none ; 消除排序圆

* <ul>标签专门用于网页中存在父子结构的功能导航栏（天猫，聚划算…）

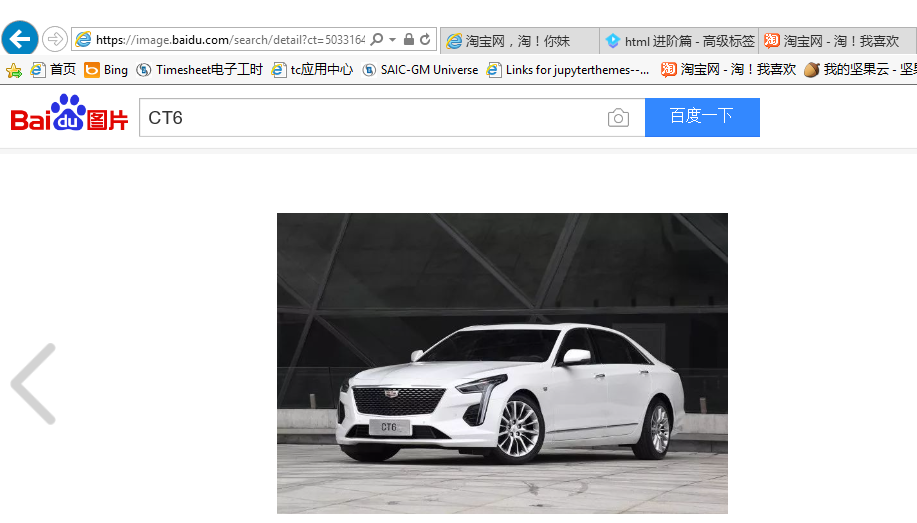
 

* + - 网页图片导入功能标签：单标签 <img src=““ alt=”“>

属性 src=““ ，source图片资源

1. **网上url**，把网上的图片引入，操作方式：

1.1 在百度搜图片，这个地址不是图片地址，是百度图片的图片地址



1.2 右键图片，属性。复制URL地址，放入 src=““中 （IE浏览器）

1.2 其他浏览器，右键 ‘在新标签中打开’

**2. 本地的相对路径**

2.1 相对路径是指，目标图片与本Html文件在同一文件夹下，可用相对路径

2.2 <img src=“pic.jpg” *style=“width:200px;“*> 直接输入图片名称即可

//img中直接输入图片尺寸，不加px单位也可。

**3. 本地的绝对路径**

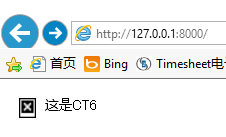
3.1 当图片文件和html文件不在同一目录下时，需要指定路径

3.2 <img src=“D/…”>

属性 alt=““，当图片地址传输或者写错时，会展示不出来。Alt-图片占位符

1. <img src=“pic.jpg” alt=“这是CT6”>

2. 当图片地址出错时，会显示alt里面的内容，告诉读者或者开发员信息

Alt内容

属性 title=““，图片提示符，图片指针放上去会显示图片内容。

属性 style=“height:100px; height:100px;” 设置图片格式大小

* + <a> </a>超链接
* 网页跳转链接标签 <a herf= “text1”> text2 </a>，可用包裹任何东西，文本+图片 *anchor——锚*

1. text1 内部放网页目标的地址链接

2. text2 放网页显示用户的内容，给用户看的

3. herf = hyperText reference 超文本引用



// 如果不带https://，直接写www…则直接在当前url后

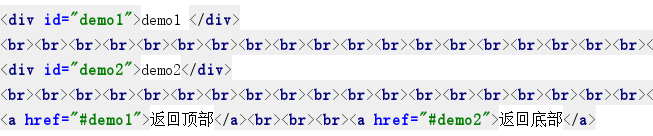


4. text2变为<a>标签，放图片作为链接:



5. 新标签页打开网站，<a herf= “text1” target=“\_blank”> text2 </a>，放在<a>标签的属性中，不是img中

* <a> </a>标签最原始的功能，是锚点功能，作为锚点记录位置使用的，同一个页面下的定点位置。跨越到不同页面，就是后来的超链接了。
* 锚点写法： herf=“#ddd”，使用#



应用： 1. 置顶

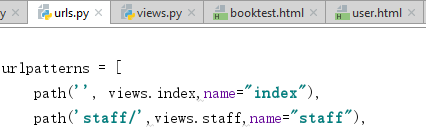
2. 朋友圈双击回顶部

3. 在一页的页面中，左边目录，右边章节，点一下目录到对应章节

4. 指定页面跳转URL：

Href = “{% url ‘ ***yoururlname*** ‘% }”





* <a> </a>标签还可以打电话，发邮件（移动端，或电脑与手机在同wifi下）

用法：<a herf=“Tel:18261975262“> 给wzh打电话 </a>



应用： 饿了吗打电话

* <a> </a>标签 协议限定符，调用 JavaScript语句，实现嘿客功能

<a herf:“JavaScript：while(1) {alter(‘你瞅啥’)}”> 你过来啊！ </a>

应用：书签，黑客

总结 - <a>标签4大功能:

1. 超链接

2. 锚点

3. 打电话发邮件

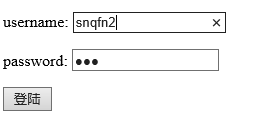
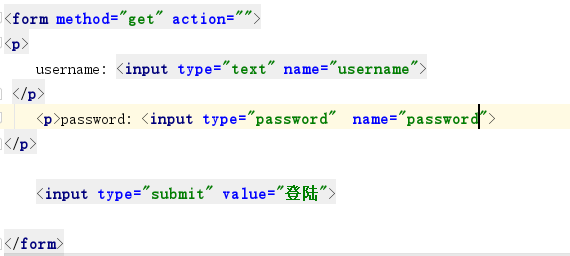
4. 协议限定符

* + <form> </form>表单标签，是前端获取数据传递至后端的方法标签（方法+地址）
  + <form method=“get“ action=""> </form>，action=““是后端接收的地址，将<form>内包裹的信息传递至后端。方法是”get“
  + Form标签内部常用信息获取方法：<input>单标签

1. 提供输入框功能，一般包含 **名name+ 值value**，输入的内容是文本。

也就是选完后，Url中会出现 “?name = value” 你的选择

<input type=”text” name=“”>。



2. 也可特定指密码框，type=“password“即可。默认暗文显示。

// 调整文本输入框的长度：属性style="width:500px;"

内容是文本框内输入的东西，起名字是“name“属性提供，给内容贴标签，告知后台这是什么数据。

框前可定义：username,password，使用<p>将用户名和密码放到两行；

3. 提交信息，<input type=”submit” value=”登陆”>，默认值是‘提交’。“value“属性是显示的内容，区别name,name是加标记。

4. 判断数据是否发送成功，点提交后，查看Url-?后是否有输入的内容~



Done

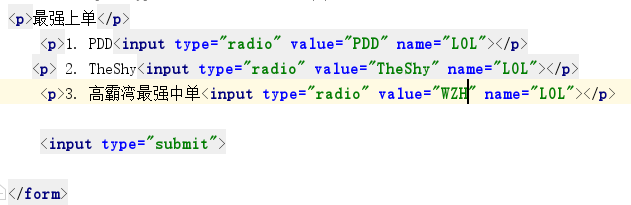
5. <input >的**单选框**，type=”radio”属性(区别于text文本框)。

<input type=”radio” name=“” value=““> —— 显示小圈

其中：

Name= 属性，是告诉系统这个单选是属于哪道题的。不然这些选 项就不具有排他性，变成多选了。

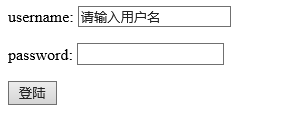
Value= 属性，是告诉系统选的是哪个内容，有名有值才能提交，显示 url?后



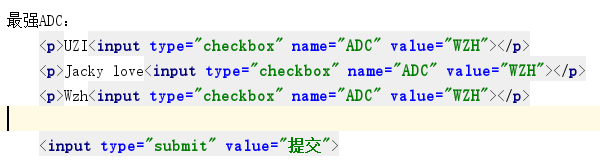


6. input 文本框，给 value= 值属性，会在文本框默认显示value.





7. 复选框，<input type=“checkbox“ name= “” value=““>



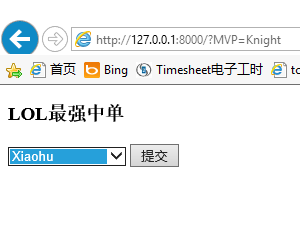
其中： name属性告诉浏览器这是同一题的选项。Value告诉浏览器选的是啥。

8. 为了更高效的开发，一般会给予**默认选项**。<input >头标签中输入checked=“checked”即可设置为默认选项。

9. input标签调整大小，使用style行间样式选择器设置



* + <select> </select>标签，下拉菜单。
* <select>标签成组出现，相当于架子，中间包裹若干选项标签<option></option>
* 和ul，div，span类似，只用在架子<select>上给定name属性，则全都有。
* 首要以value属性为主，可作弊。
* 题目可用搭配用<h>标签

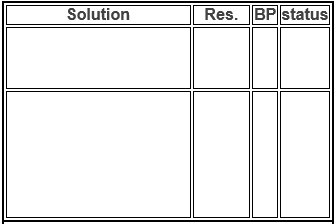
* Table标签：

<table id=” ”> 调用表格标签

<td> table d 指定行数，td内有几行代码，则表格有几列

<th> table head 表头数据，具有特殊的字体和格式

<tr> table row 指定行数，tr内有几行代码，则表格有几行



## 第四课：CSS 选择器

* 注释：系统不会执行，2个功能。

1. 备忘录，解释

2. 用于调试代码，将可疑项注释，观察结果。

* 注释方法： <!-- --> 快捷键： CTRL + / css代码注释方式 /\* … \*/
* CSS（ cascading style sheet ）层叠样式表
  + 引入CSS（三种方法）
* 行间样式，利用标签属性style：<div style=”

Width:10px;

Height:10px;

Background-color:green;

”> </div>

* 页面级 CSS，在<head>中编写CSS代码，此时使用一对<style>标签：

<head>

<style type=”text/css”> //*默认出现的，可写可不写，不可写错*

Div{

Width:10px;

Height : 10px;

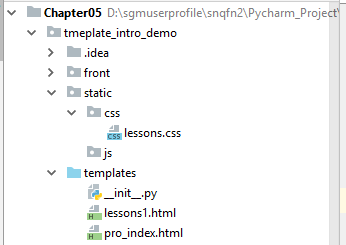
Background-color: green;}

</style>

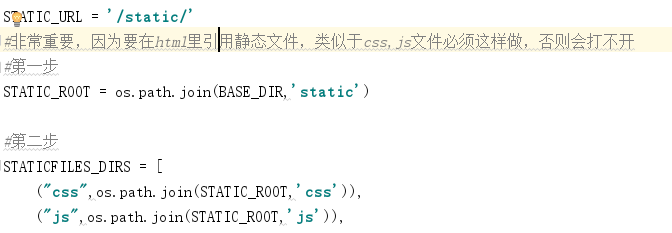
<head>

* **最常用实用高大上-**外部CSS文件，创建 .CSS文件，然后引用;
* 外部引用CSS文件设置：

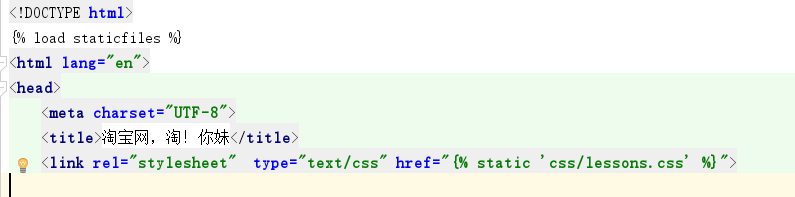
1. 在项目下创建一个static文件，创建css和js文件夹存放相应文件



2. setting中设置static引入的相关路径：



3. 新建.css文件，引入static设置 {% load staticfiles %}；并在<head>标签 中设置 <link rel="stylesheet" type="text/css" href="{% static 'css/lessons.css' %}">



* Css中选择器
  + Id选择器：在<div>头标签中加入 id属性，<div id=”only”> 123 </div>
  + 在css文件中，通过id名选择该div，关键字：**#+名字** ，语法：

#only{

Background-color : red;

}

* + Id选择器特点：每个标签**只能有一个**id，与css中渲染效果是一一对应的。
  + Class选择器：在<div>头标签中加入class属性，<div class=”demo”> 123 </div>

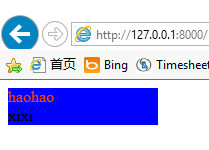
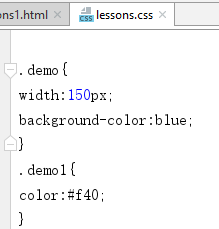
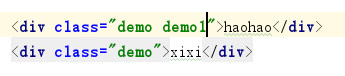
在css文件中，通过class选择该div，关键字：**.+名字**，语法：

**.**demo{

Background-color : red;

}

* + 1个class**同时作用多个渲染效果**： <div class=”demo demo1”> text<div>



* + Class选择器特点：class选择器可以在多个标签中，即可一次同时渲染多个同名class标签；而且，一个标签可以同时使用多种效果渲染，如同时设置背景色和字体色。
  + 属性选择器： 对所有存在该属性的标签进行渲染。

关键字： [ *属性名* ] or [属性名=“值“]，

[id = “only”] [class=””]

{

color:#f40;

}

* 属性选择器特点： 1. 主要是用来改变标签的原有属性，自定义标签。

2. **优先级和class选择器相同**，后面覆盖前面的，可用于单选和多选。

* + 标签选择器： 对**所有选中类型**的标签进行渲染。不管标签被套多少层都可以选到。

用法： 在css文件中直接输入该标签<div>/<span>等，并编写css语句。

div{

color:#f40;

}

* 通配符选择器：选择html页面中所有标签，包括<head><html>等进行渲染。
* 关键字 **\***，用法：

**\***{

color:#f40;

}

* + ! important 选择器： 优先级最高的渲染效果。在某选择器的属性后加上该语句，则该选择器效果最优先。

{background-color：green !important} —— 最高优先级

* Css选择器优先级：

！**important> All**

**style行间样式> ID选择器> class选择器=属性选择器> 标签选择器 >通配符选择器**

理解： 纹身 > 身份证 > 马甲 > 名字 > 乱选

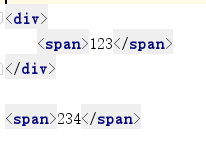
优先级相同时，后者渲染效果覆盖前者

## 第五课：CSS 权重&盒子模型

* Css选择器权重：各选择器之间优先级不同，因为其权重不一样

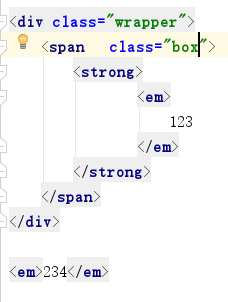
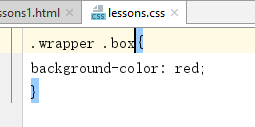
|  |  |  |
| --- | --- | --- |
| 选择器 | 优先级 | 权重 |
| ！important | 1 | Infinity |
| Style行间样式 | 2 | 1000 |
| Id选择器 | 3 | 100 |
| Class / 属性 / 伪类 | 4 | 10 |
| 标签 / 伪元素 | 5 | 1 |
| 通配符 | 6 | 0 |
| **注：**权重的数值不是10进制，而是256进制，即0和1之间差256！ | | |

* 父子选择器（派生选择器）：选择**父子结构**标签的选择器，可用标签选择器单独作用指定的内容。
  + 利用标签选择器的父子关系作为限定修饰，语法：父亲标签 目标子标签 { CSS }

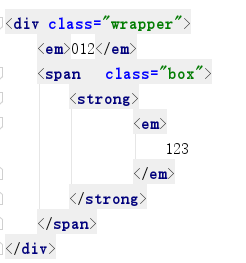
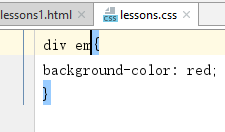
  

如果有多层父子，只用选能清楚定义的父子层就够了无需写全。如： div span em {}

* + 利用其他选择器的父子关系也可以，如class选择器父子选择器等，不在乎什么类型的选择器，只要是父子！

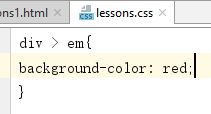
* + 父子选择器不一定必须要是直接的父子，子孙等都可以。

Div下有两个en，直接父子和间接的。都起作用

* + 直接子元素选择器：如果只想选父子，其他的子孙不选，只选直接一级的儿子。

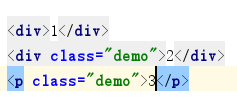
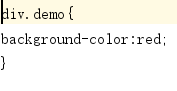
关键字 **>** ，用于指定继承：



* + 选择器排查顺序： 当遇到多层选择器时，浏览器**从右向左**排查各父级标签。
  + 并列选择器：当多个标签类型、class值重复，想要单独标记2时，挑出其独特性。

用法： 直接两者写一起，div 和 .demo ，注意不要写在 **.** 后，否则会误解。

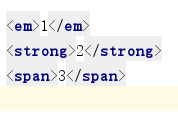
//这种情况也可以加 id，但是麻烦

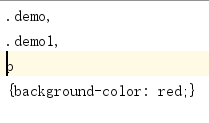
 

并列选择器涉及到优先级的问：当并列的内容过多时，只要写在同一行的选择器，判断优先级根据**权重值相加**判断。且都有！important时，再比较其他的权重。

* 分组选择器，将所有选中的类型选择器都进行渲染，一般常写成竖列。

用法： 使用“逗号”将各标签隔开，也可几个类选择器分组渲染

用法： 分组选择器专门用来简化代码的

* Css代码：属性名+ ： + 属性值 + ；字典网站 [www.css88.com](http://www.css88.com)

字体&框

* Font-size: 16px 字体大小，默认是16，设置的是文字高度
* Font-weight： lighter/ normal / bold /bolder 加粗：细-最粗；也可数字表示 100-900（但最粗需要看这个字体包里最粗的是多少）
* Font-stlyle: normal / italic; 斜体作用，<em>是由这实现的；
* Font-family：Arial 字体。互联网默认Arial通用，乔布斯发明的。 Cursive-楷体
* Color： 字体颜色，3种方式。

1. 单词（red,green,yellow…），开发时不能写这种

2. 颜色代码（RGB-三原色红绿蓝，#ff0000，每个位置16进制，00-ff,如果相同可以写3个，#f40淘宝红。

3. 颜色函数，rgb(0-255,0-255,0-255)；

* 颜色渐变：

1. linear-gradient：； 线性渐变

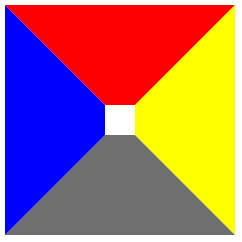
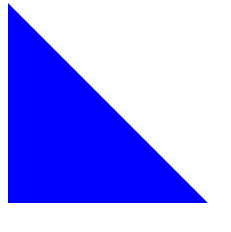
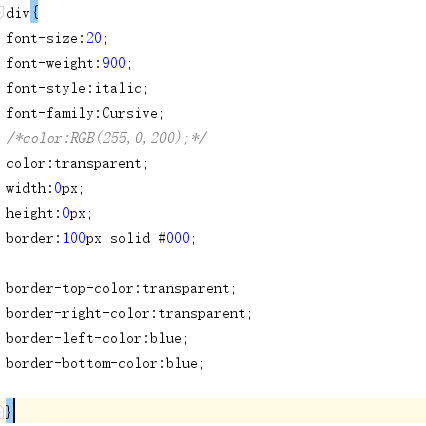
* Color: #f40 /#fff/ #000 / transparent 淘宝红/白色(3个颜色满了)/黑色**/透明色**
  + Border: 10px solid black; 给标签加外边框，第一个属性是粗细， 第二个是实心，第三个是颜色，配合 width , height

Border-width:10px (实际上是4个复合值，顺时针设计，也可单独设计)

Border-style: solid/dotted 点虚线/dashed 条状虚线

Border-color: 颜色

* + Border-left/top/: (width style color) 可以单独设置每一条边。用来画三角形。

* Background-color 背景颜色：red/green…，标签内必须有内容才生效
* Border-radius :10/ 50% 边界圆角,50%时为圆
* Opacity 透明度，取值 0-1之间

// ctrl + D 复制当前代码

* Width: 100px 标签框宽度：100 像素
* Height：100px 标签框高度：100像素

//当没有设定宽和高时，变为自适应区域，高度随着内容变化，宽度无限制。

文本

* 文本对齐方式： *text-align*: left / center / right
* 文本行间距： *line-height*：

文本垂直居中方法： line-height: 16px; 和文字高度相同，即0行间距。可将文本框的高度height = line-height。**单行文本适用**

**{**

**Height ： 30px;**

**Line-height : 30px;**

**}**

Height = 3\* line-height >>> 则文本框只能展示3行

* 文本缩进：*text-indent* : 2em; “em”是相对文本单位，字体宽度1 font-size =1em

**用法**：用于设计文字文本行高，line-height: 1.2em 相当于1.2行间距。

* 文本划线标签： *text-decoration* : line-through 中划线 / none 若用在<del>标签种，则把划线取消了 / underline 下划线，相当于<a>标签 / overline 上划线，没用（Del是禁止使用的标签）
* 鼠标光标定位：cusor： pointer 鼠标移上去变小手 / help 变问号 / resize
* 伪类标签
* :hover{ } 当光标移到标签范围时产生变化，光标移除效果消失前部写各类选择器都可以，标签/属性/class等等

//注意权重问题，a{ } 权重 < a:hover{ } < style{ }

* 行级元素/内联元素（inline）：内容决定元素所占位置，不可以通过css改变宽高

Span strong em a del (文本类元素)

* 块级元素（block）:独占一行，可以通过css改变宽高

Div p ul ol li form address

* 行-块级元素(inline-block)：内容决定大小，可以改变宽高

Img

// 可以更改标签的固有属性display，inline-block 可以自定义。

Span{ display:inline/block }

* 块级元素和行级元素区别：

1. 块级元素一般用来搭建**网站架构、布局**、承载内容……像这些大体力活都属于块级元素的

2. 行级元素一般用在网站内容之中的某些**细节**或部位，用以“强调、区分样式、上标、下标、锚点”等等

* 元素间内嵌关系：

1. 块级元素可以包含行级元素或某些块级元素，但行级元素不能包含块级元素

//如 块li可以包含块div

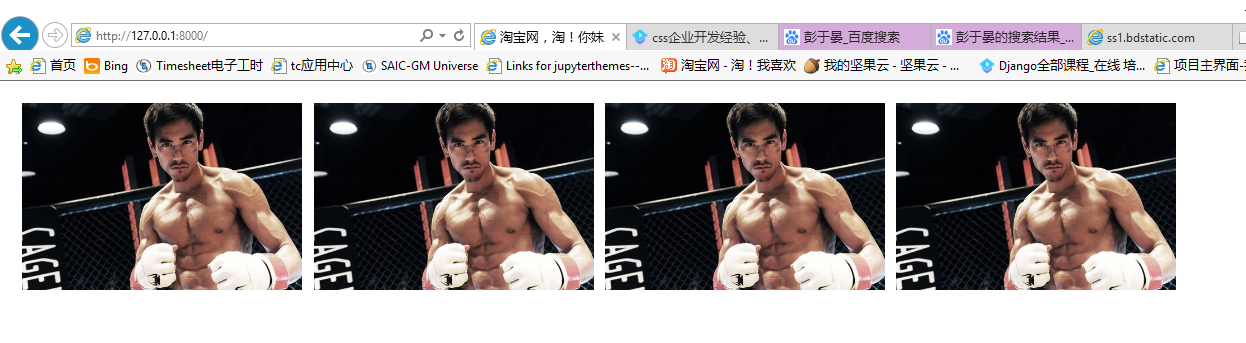
2. 这些块级元素<p>，<h1~h6>里面内不能放块级元素，只能包含行级元素

3.块级元素和块级元素并列，行级和行级并列

* **经典BUG：**导入图片时会产生图片间的空隙

//凡是带有inline的元素，都带有文字特性，就该被分割，即带有文字分隔符，所以图片本身就带有4像素的文字分隔符造成的边距。





解决方法：

1. 在html代码种，将图片间的空格和回车删除，使得img代码首尾相连；

2. 在css中利用margin属性，将图片-4px。Img{ margin-left: -4px;}

//图片的宽高只用设置一个，另一个会等比例缩放。

* 开发习惯
* 先写css，css模块化

用法：公司会有很多现有的css文件模板，直接调用就可以，模块化轻量化

.green{

Color=green;

} 把这个放到对象的class属性种即可。 < class=”green”>

* 自定义标签，属性选择器，使得原始的标签一出来就不一样。

Css文件开始就写 a{ }, em{ }…

* 通配符标签\*妙用：专门用于设置页面的所有标签的内外边距：

1. \*{ } 标签选定的范围广，页面所有的标签都会被选中

2. 权重为0，任何时候用户定义都可以生效。

**用法**：元素出现时与页面会有一个默认的边距，这是由于<Body>默认带有**8**px的margin，使得元素出现时与页面边距有些距离。

一般使用通配符设置：

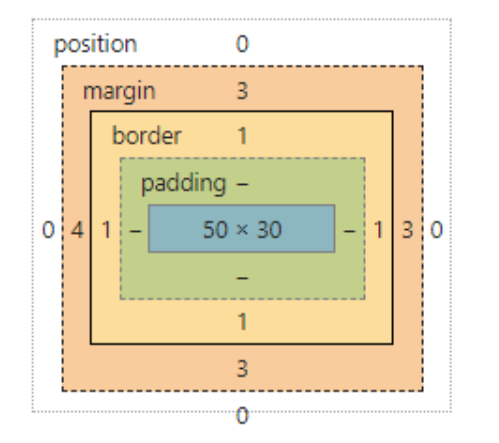
\*{

Margin：0；

Padding：0；

}

* 盒模型：
* 盒子四组成部分 盒子壁（border），内边距（padding），盒子内容（width ,height），外边距（margin）



* Padding：100px 100px 100px 100px 内边距，边界和内容的填充物

设计方法:（4个参数时-按照**上右下左**的顺时针填充方法）

（3个参数时-按照**上 左右 下**的顺时针填充方法）此时盒子左右等距

（2个参数时-按照上下 左右的顺时针填充方法）

// border-width类似

Padding-top: 单独设置上

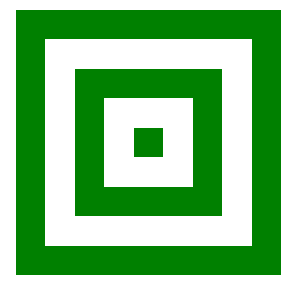
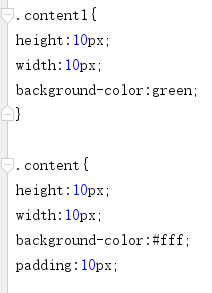
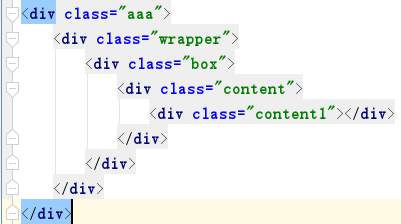
Padding-left:

* Margin 外边距

设计方法：与padding类似，4个复合值。

应用：控制元素间的间隙 margin-left:10px；

* 盒子模型计算：略~
* 远视图：在同一标签下，前元素框作为后一元素框的内容呈现。在两框之间填充恰当的padding尺寸。

//**盒子居中方法**：1. 使两个盒子一样大，两层中间填充padding即可。

2. margin： 0 auto; //上下无空余，盒子左右居中

* 层模型
  + 定位技术： position

1. absolute定位：脱离原来位置进行定位，原来占位让出来。*浮动*

用法: Position: absolute; 此时可被定位，一定要有这个声明

Left/right:10px； 元素框左侧边线距离左侧，一个即可

top:10px; 一般不设置底部

* 特点：此种定位方式，是将设定为absolute的定位元素提高一个图层，和其他元素 不在一个层，互不干扰。
* 特点：相对于最近的有定位的父级进行定位，如果没有，则相对于文档定位。

//**z-index：0/1/2…** 页面的层级，元素默认在0层，数值越大越在上层

2. relative定位：保留原来位置进行定位，原来占位也不给出来。灵魂出窍。

用法: Position: relative; 此时可被定位，一定要有这个声明

Left/right:10px； 元素框左侧边线距离左侧，一个即可

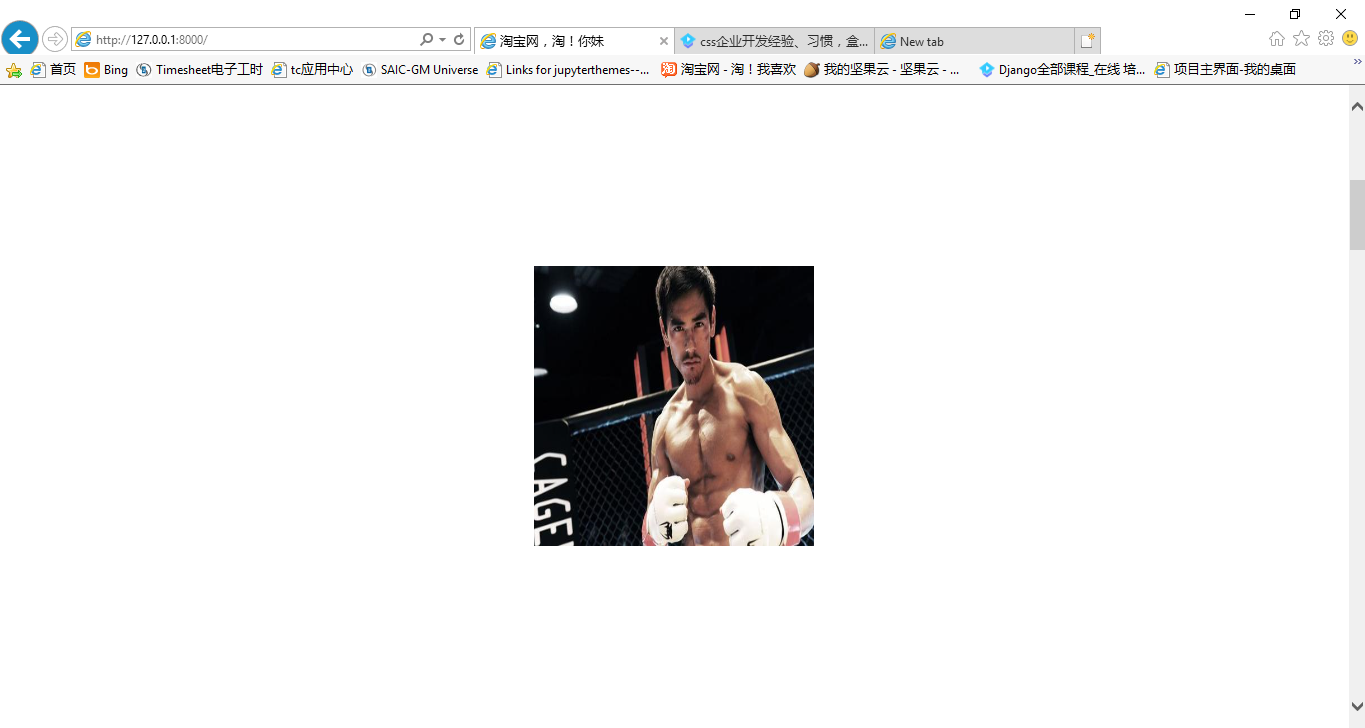
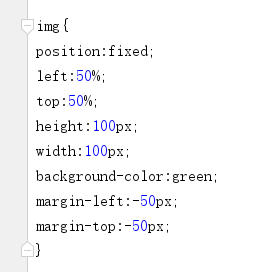
top:10px; 一般不设置底部

* 特点：此种定位方式，是将设定为absolute的定位元素提高一个图层，和其他元素不在一个层，互不干扰。
* 特点：相对于自己原来的位置进行定位的。
* 定位用法：用relative作为父级参照物，用absolute进行定位。减少对后续元素的破坏。如果用absolute作为参照，则其他元素全部窜上去了。
* 3. 广告定位：fixed

//不随着页面滚动而动。设置的位置是元素左顶点位置。

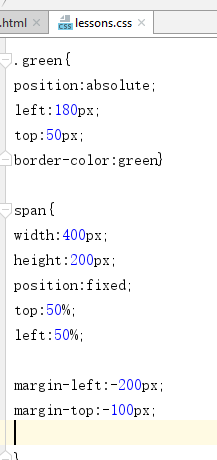
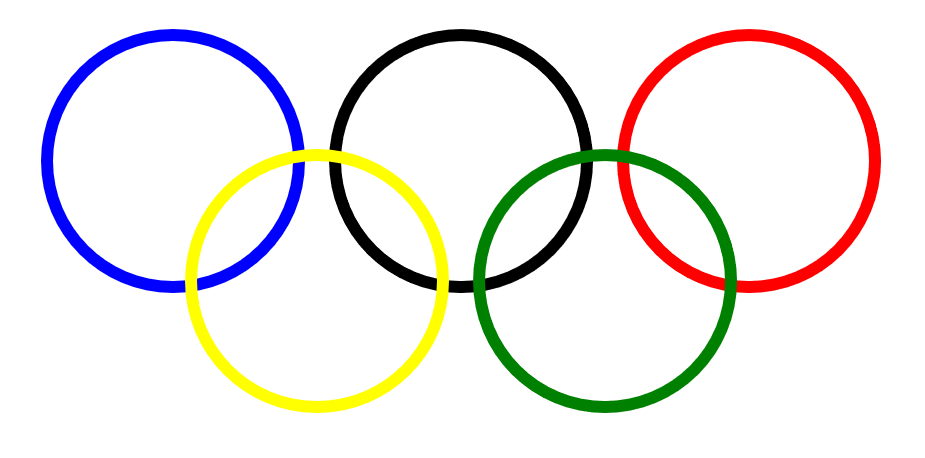
广告居中悬浮定位，不随页面滚动，且可以随着窗口更改始终在中间：

* 1. Position的位置可以采用百分比，50%表示居中
* 左顶点在中心，但是整个图片不在中心，需要利用margin移动图片。



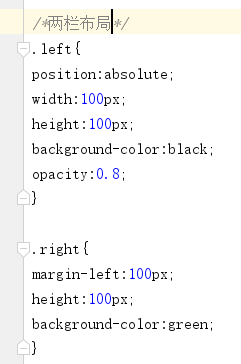
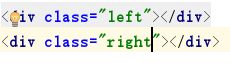
作业：画居中五环？

// 五环设置position后，五环就不在原始图层，使用border则无内容。

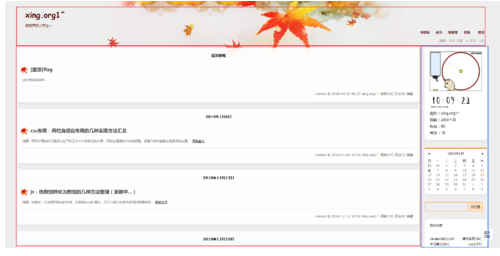


## 第六课：CSS 两栏布局 & 经典BUG & BFC

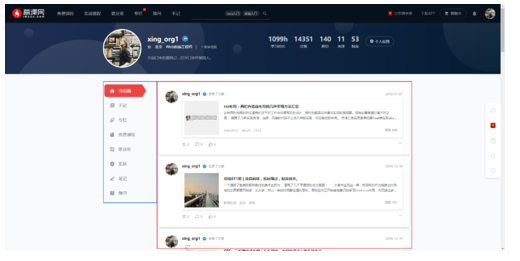
* 两栏布局
* 网页经典布局方式，很多有名网站都采取这种类型布局。将主要内容和介绍/导航等内容分开写。
* **常规用position：absolute，也可用浮动。浮动元素先写**，否则不会向上占位。
* 其中一栏不定义宽度（非浮动栏，若浮动栏则宽度固定随内容走），为避免两栏盒子区域重复，注意使用margin元素。

应用：



博客



* **Bug1**- Maigin塌陷
* 1. 父子元素在垂直方向上的margin-top，取两者最大的值一起动。子元素的margin-top不以父级元素的顶棚作为边框移动。好像父级的顶棚消失。
* 弥补办法 – BFC
  + BFC（block format context）：css默认将每个html元素作为一个盒子来渲染，且每个盒子的渲染语法是一样的。而bfc可以单独定义某个元素块的语法规则。使用BFC专门用来针对margin塌陷问题。
  + 通过改变父级的渲染规则，使父级/兄弟级变为BFC，解决子元素的塌陷问题。
  + 每个方法都只能弥补问题，可能会引入新的问题，需要根据实际需求选择。
  + 如何触发盒子的BFC:

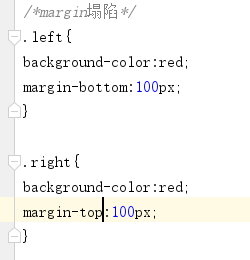
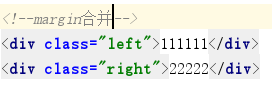
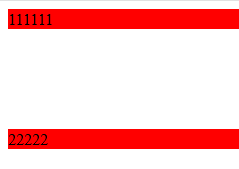
1. position: absolue;

2. display: inline-block

3. float: left / right;

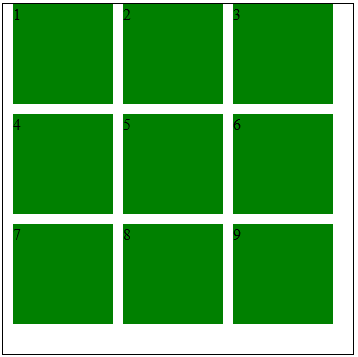
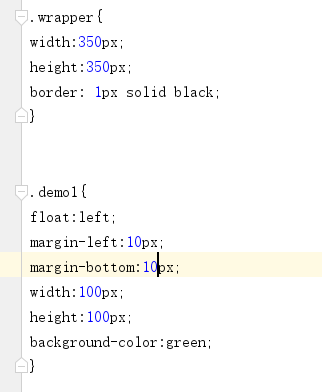
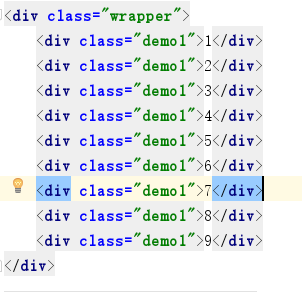
4. overflow: hidden;

* Overflow: hidden; 溢出盒子的部分隐藏展示。在父级css中，超过父级框的就隐藏。而且可以避免Margin塌陷了。子元素可以移动了。
* **Bug2**- Margin合并
  + 1. 并列块元素在垂直方向上的margin-top，不累加，且会合并。

本应该是200，但只有100px

* + 笨B解决方案: 如果不是父级的关系，是并列的兄弟级关系，如两个<div>，此时可以在html中创造父级，外部创建一个或者两者的父级，再触发BFC就可以解决。
  + 这个bug不解决，原因：绝对不能随意改动html的结构。结构最重要，不能因为一个bug带来结构的改变。可以根据像素计算自行处理，直接一个里面写200px。
* 浮动模型（三大模型之一，盒/层/浮动） float: left/right
  + Float: 在子元素中，使得子元素按照从左往右/从右往左的方式进行站队排列，每行站队的个数取决于父级的边界。
  + 把竖着的<li>，<div>这些独占一行的block元素变成可以并排站立。

* + 在排列过程中还可以使用margin功能，-left或者-bottom等

应用：



* + 浮动元素产生浮动流：

1. 所有产生了浮动流的元素，块级元素看不到他们，会直接重叠过去使用他们的位置

2. 产生了bfc的元素和文本类元素（inline）的元素及文本能看到浮动元素

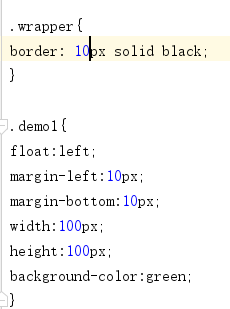
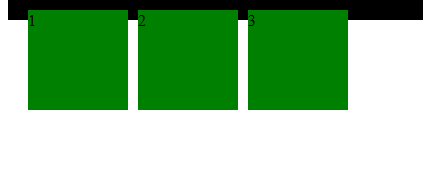
* + 浮动流理解： 看得到看不到是指这两类元素的占位，看不到时，会直接重叠上去仿佛这里没有元素。看得到是指inline元素会站在浮动元素后面。
  + 和“层“的概念有些像，但是针对不同的元素，层级不确定。
  + 想要使浮动元素被看到:

1. 将块级元素通过 float也变为bfc元素。就不会重叠，就会被看见。

2. 解决浮动流。

* 解决浮动流
  + Clear：left/right/both 清除周边的浮动流，**只适用于块级元素**
  + 用途： 在后期不断增加子集个数时，想要父级边界一直包住自己元素，使用<p> clear：both属性，清除p标签附近的浮动流。
  + 问题引入： 自适应包裹浮动元素

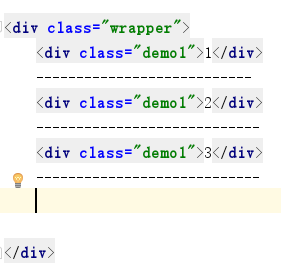
想要父级一直包裹子级元素，但子级元素的浮动流导致**父级块元素看不到**。会自动缩回去。如何设置使得父级边界能自适应包裹子级元素。



问题表现

* 问题分析：

由于浮动float的存在，会在浮动元素的下端产生浮动流，导致父级的下边界看不到内容的下沿。

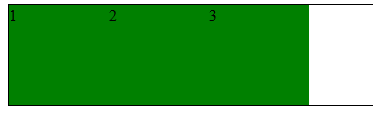
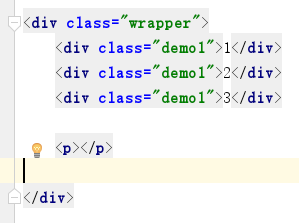
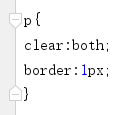


只要在所有浮动元素的最下端设置一个下边界能看到的内容即可。

* 解决方法：

1. 将父级元素也出发BFC，其下边界即可看到浮动的子元素。（详见下章）

2. 最下边界处，引入块级元素<p>，为了p不受浮动流的影响浮上去导致父级底线也上浮，需要清除<p>附近的浮动流。



// 此时，父级边框感应到的是不受浮动影响处于最底层的<p>标签内容。

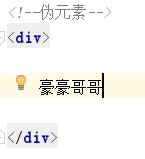
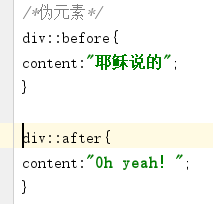
//这种方法同样是犯错了，相当于改变了html格式，这是不允许的。

3. 使用伪元素。见下章。

## 第七课：包裹浮动元素

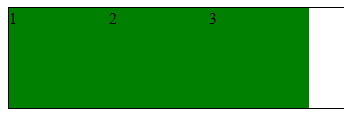
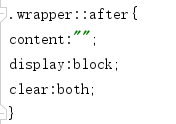
* 伪元素
  + 伪元素：天生存在于每个标签元素中，存在于标签逻辑的最前和最后两个位置，可以通过CSS选择器选中进行编辑样式和结构。这种可以当元素来操作，但是没有写在html元素结构，的叫伪元素。
  + 伪元素关键字： **：：before{ content：“”； }** / **：：after{ content:”” }**

双冒号，标签逻辑前/后伪元素，内容更改

* + 伪元素选择器的权重 = class选择器 = 属性选择器 = 10
  + 伪元素是行级元素（inline），如果想要设计宽高，需要display: inline-block设置
  + 利用伪元素清除浮动：在**块级**标签逻辑的最后加上伪元素，并使用clear消除浮动。

1. 伪元素内必须要有 content：“” ，否则伪元素不生效。内部置空也可；

2. clear只针对于块级元素生效，伪元素不是块级元素，需要内部转换 display**:**block; 

标准解法——包裹浮动 **三件套**

{

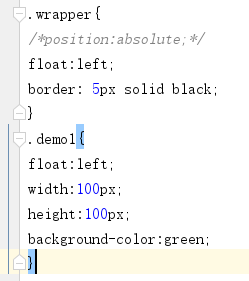
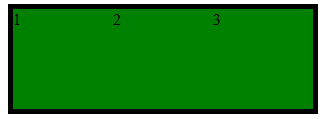
Content：“”；

Display: block

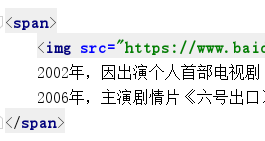
Clear : both;

}

* 自适应包裹浮动元素-**方法一**
  + 上述可知，块级元素看不见浮动元素导致无法包裹子级元素。方法一是将父级元素触发BFC，使其能看到浮动的子元素即可。
  + 使用 position：absolute； float：left/right； 触发BFC：确实包住了底部。

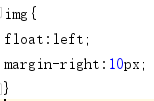
 

* **引发新问题**： 变为横向紧紧包裹住子级元素，而不是随页面宽度。Why？
* 新知识点： 当使用position：absolute； float：left触发BFC时，会使元素变为inline-block属性，此时，元素的边框随内容决定。造成了这样子。
* 同时，position：absolute； float：left/right 也可以起到 display的作用将inline元素变为具有block属性。
* Float原始用途是使文字环绕图片。
  + 问题：若在图片后直接加上文字，会出现底边对齐情况。无法达到报纸效果。



文字在图片下边沿出现

* 解决： 给 < img >标签增加float。

* 浮动编辑标准导航栏
  + Ul-li标签为标准
  + 开篇\*{ }设置边距，padding，字体
  + 文本垂直居中使用 height=line-height
  + 使用属性标签时 > 指定子标签

## 第八课：文本溢出&背景图片设置

* 文本溢出容器
  + 单行文本溢出处理： 三件套

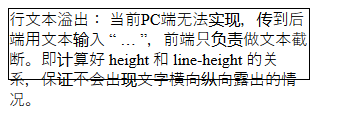
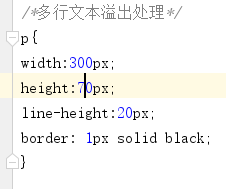
White-space : nowrap; // 取消换行功能

Overflow : hidden; // 溢出模型部分隐藏

Text-overflow : ellipsis; // ellipsis 椭圆的复数，意思是文本溢出部分“ … ”



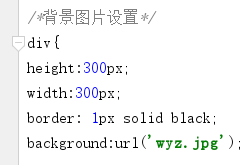
* + 多行文本溢出： 当前PC端无法实现，传到后端用文本输入 “ … ”，前端只负责做文本截断。即计算好 height 和 line-height 的关系，保证不会出现文字横向纵向露出的情况。

纵向露出

原因： 没有设置好height 和line-height 的倍数关系。导致溢出

* 背景图片设置—— **url / size / repeat / position**
  + 在图框中设置背景图片： background：url (“ 123.jpg”)



* + 图片未完全显示：

设置图片背景颜色和框尺寸大小一致即可，background-size : ;

均可居中填满显示。否则图片会自动重复填充。

// background-size: 50% 50%; 第一个设置宽度，一个设置高度，占全高宽的。



* + 背景图片

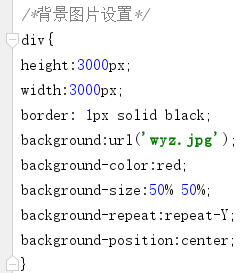
Background-repeat：no-repeat; //不重复，只会显示一张

: Repeat-X/Y // X，Y向重复平铺，

Background-position：50% 50%; //图片居中显示

：center //居中显示

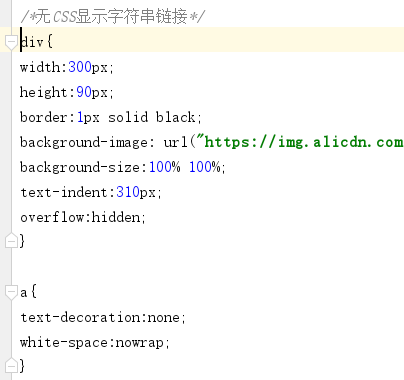
：Right Top; //图片在右上角显示

* 浏览器默认加载策略
  + 大网站的加载策略

一旦网速降低至20kb/s 时，还是要保证网站能正常运行打开，这时候浏览器会屏蔽所有css和js文件，只剩下html文件。即正常时图片链接正常，网速过低时图片消失，只剩文字链接。

* + 方法一： 将淘宝网图片设置为框架背景图片，随后通过text-indent: width; 和 white-space: nowrap;将超链接<a>文字放到框架外部，最后使用overflow：hidden；将超出图片的链接文字隐藏。

//这样一来，有CSS时候显示图片，没有超链接时候，overflow功能小时，显示链接文本。

* + 方法二：将淘宝背景图片放到元素框的 padding 内容中。利用padding-top将文字挤出显示框内容，结合overflow：hidden；隐藏文字。

**Height：0px；** // 文本框内容高度0，文字就缩到框外面去了

Padding-top: 行高90px； // 文本原始行高，用padding把框撑起来

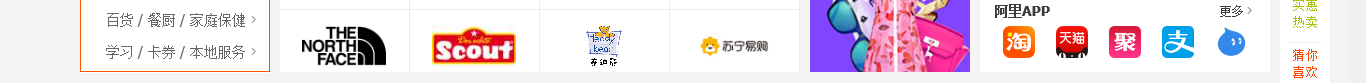
// padding内是可以放置背景图片的

# 实战：淘宝页面

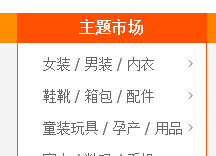
## 第0课：做淘宝页面问题搜集

* 练习中的问题汇总





* + 1. 如何开始一个网页，把一个完整的项目分解为哪些框架和区域？框架确定后，再在上层输入框架则会全乱。页面放大后，排版都乱了。
* 2. 在<div>方块中，有<a>标签，指针在div方块上时，a标签内容产生变化如何设置?



* + 3. 如何显示默认形式，: hover的拓展功能?



* + 4. 指向<a>标签时，外部<div>框架显示图片，:hover的拓展功能?



* + 5. 导航栏中的 | 是如何设置的？





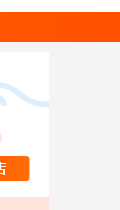
* 6.1 搜索栏<form>标签的text和submit属性如何无缝连接，改变样式？
* 6.2 搜索栏<form>标签的text标签如何设置默认显示文字，放大镜，照片？

Placeholder

* 6.3 搜索栏<form>标签的text标签如何设置默认显示文字，放大镜，照片？
* 7. 如何淘宝网和搜索栏fixed浮动显示？



.8. 页面的两侧留白是使用的何种方式实现？

## 第1课：前期准备&基础结构样式

### 淘宝首页页面（第一屏幕）



### 页面整体布局策略：

* + 划块而治

划分结构原则： 从上至下，从左到友，一行作为一个单位；

可以按照内容；结构块；间距；边框；颜色；功能，等作为划分依据。

* 划块方式：<div> , <span>
* 按照个人习惯,在div中配置class或id属性选择器,一般来说,id用于独立的块或者最外层的块

#id .class

### 前期准备：

* 创建img文件夹
* 创建一个CSS文件，作为样式重置文件，即全部放默认网页样式的代码

### 通用配置设计——**样式重置**/样式模块化/预定义样式

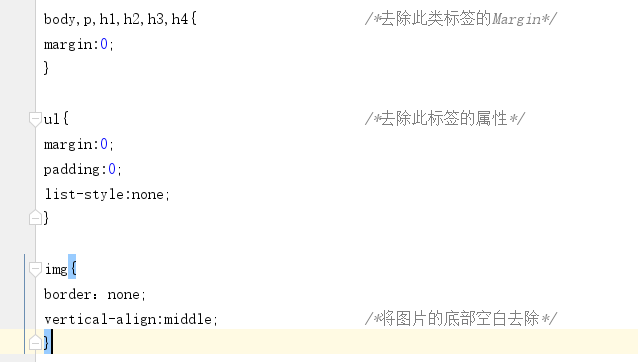
先写CSS文件，定义通用标签的默认样式，写结构化HTML时，会碰到很多频繁引用的标签样式，提前预定义样式，在引用时，直接标签中引用预定义的Class即可。

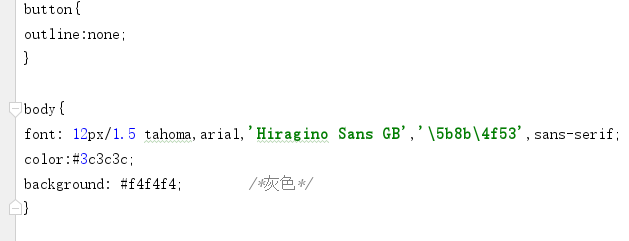
* + 部分标签的原始属性移除，一般是某些标签的固定margin和padding等

1. 将需要使用到且需要修改的标签拎出来，使用分组选择器进行设置：

Body, p, h1, h2, h3, h4 {

}



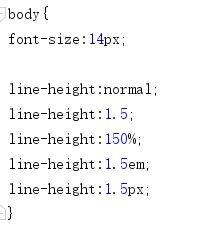
* + 字体，颜色，行间距等网页默认设置：

Font的各种属性可以集中设置，一般放在body的标签中：

Body { font: 12px/1.5 tahoma,arial,'Hiragino Sans GB','\5b8b\4f53',sans-serif;}

body{ font: Size/line-height font-family }

1. line-height 行高设计：



行高设计有多种方式，但有一种行高设计与众不同，**使用数字行高**，即line-height:1.5;这种方式可以让子模块中的行高随着子模块中的字体变化。

如果使用其他表达方法，则子模块中只会继承父模块中的固定行高，而不会随子模块中变化。

2. font-family字体设计：

前后端数据交互时，代码中尽量避免出现中文。在编辑字体时尽量使用英文或代码表示方法，**'\5b8b\4f53'** 是‘宋体’的意思，使用 ‘fehelper’ web网页开发者工具，对字符串加/解密的工具，将不同的字体转为unique字符串，供网页识别。

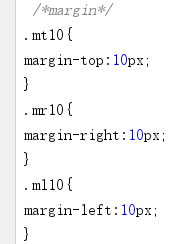
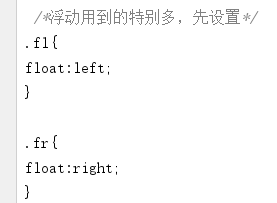
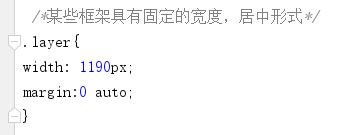
* 字体设计方法（3种）：

Font-family: '\5b8b\4f53';

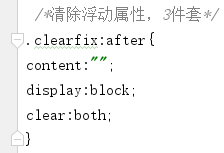
Font-family: '宋体';

Font-family: Simsun; //宋体的英文单词，微软雅黑 ：Microsoft YaHei;

* 常用样式属性预定义： 频繁使用的属性预先定义，如字体颜色，浮动，边距，模块居中等

* 页面中出现浮动后务必要**清除浮动**，子级是浮动，则在父级中引用.clear ，若父级也需要浮动，继续向外找父级清除。

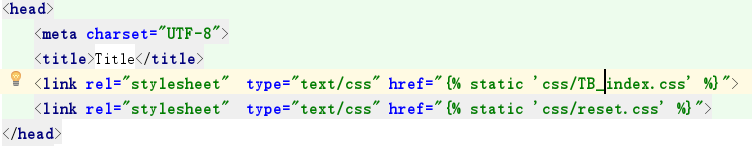


### 样式.css文件模块化

编写网页时，会产生各类样式文件，首页/重置/等等，

* + 样式模块引入方法：各类样式文件作为一个单独的模块，共3种引入方法

1.头标签<head>中重复填写导入路径



2.通过模块化引入，@规则



### Css文件@规则

Css语法中的一种，可以让编码变得高级些。常规有以下4种：

* + @charset //设置样式表的编码，@charset：‘utf-8’;
  + @import //导入其他样式文件，实现样式模块化，在css文件中引入其他样式



// 各类网站，样式多种，但不是每个样式都需要重置重新编辑，可以通过模块化引入，统一样式类型，减少重复工作，提高效率。

//如淘宝中的搜索栏模块样式，在天猫中搜索栏模块也类似，那么直接@引入即可。

* @media //媒体查询，可以查询到当前访问的设备信息，用于移动端。可以访问用户屏幕尺寸，设备类型，是手机电脑还是电视打开的。
* @font-face //自定义字体，用于网页中的各类icon，或者使用特定的字体（电脑系统中会自带部分字体，但想用其他字体，需要自己设定。后续结合iconfont详解。

## 第2课：头部信息结构

### 淘宝页面小图标 / 网页收藏夹中的页面图标

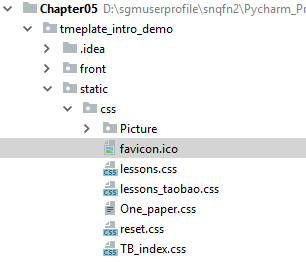


* 把图片生成.ico图标工具：<http://www.bitbug.net/>

1. 自己做，把准备好的图片通过网上的 “ico在线制作工具”转为ico图标

2. 拿现成，把已有网站的ico找到后复制保存下来——www.iconfont.cn

* 图标icon的位置： 命名为默认的favicon.ico，放在static下



* 图标代码： 使用<link>标签，指定rel类型和链接地址herf

 **or**



### Base标签：定义文档中相对url的所有根目录

* <a herf: > <form action=‘’> 都会有url的跳转链接。如果文档中频繁出现，则可以使用base标签，统一设置好根目录



//每个页面中仅仅能有一个**Base**标签，注意要有http://



//链接中使用‘#’是直接到base标签，如果去base标签下的url，直接‘写路径’即可。

// herf=”123”, 则会跳转到 base / 123 , <http://www.baidu.com/123>

* Base标签中还可增加设置，所有的a标签页面target另开网页显示：



**BUG：设置base的 herf后，ico小图标不再显示了**

**// 人员回复：ico图标要放在网页所在处，不能保存在本地的某个路径**

* + Html/css中相对路径表示方法：”../”

../../font/test.css 表示上上级目录

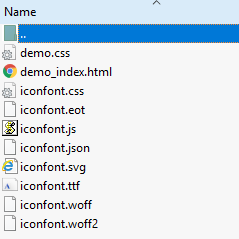
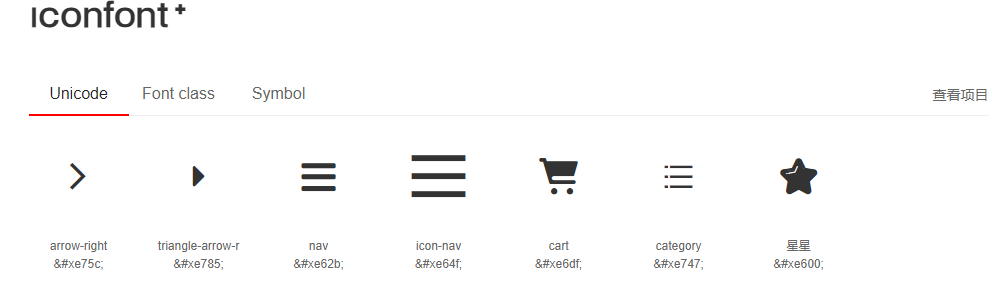
### 特殊字符标签： **将特殊字符视为特殊字体（浏览器的非默认字体，方正琥珀）**

* + 进入icofont阿里图标网站：[www.iconfont.cn](http://www.iconfont.cn)
  + 一次性将所需要的图标寻找到，加入购物车，添加至项目中（如果需要新添加图标，需要更新5文件）
  + 选择 Download Code，下载代码



* 下载好的文件包中包含各类文件，把5个文件copy到css所在文件夹，点击进入.html文件，有详细的引用方法（5个文件是在不同浏览器内核的渲染）

* 以Unicode引用方法（其他两方法font class,symbol自行学习）为例：

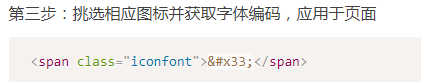
1. 拷贝下面的代码在css文件中



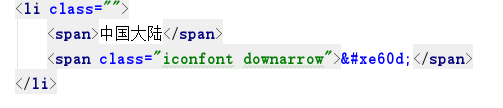
2. 拷贝第二步的代码在css中，并修改5个文件的引用路径。如果直接5文件在css文件中，则不用修改，如果在其他目录，则注意使用相对路径引用。

3. 在html代码中，copy下列代码，引入你想要的图标。Span标签中间就是代码。注意，分号；不可丢。



4. 通过css修改图标大小，颜色即可。





### 实战训练



* 搭建思路

1. 本栏模块使用DIV标签单独成行

2. 分为左侧和右侧内容，两侧的排列内容使用 ul-li标签

3. 左右侧ul标签使用左右浮动，将两栏内容并列分为两侧

4. 两侧内容输入，**凡是可以点击的，使用li内的a标签**；同属于一个li的，但是不能链接的，用span标签并列

5. 特殊图标视为文字处理，先留白，后面再处理

## 第3课：前期准备&基础结构样式

## 第4课：

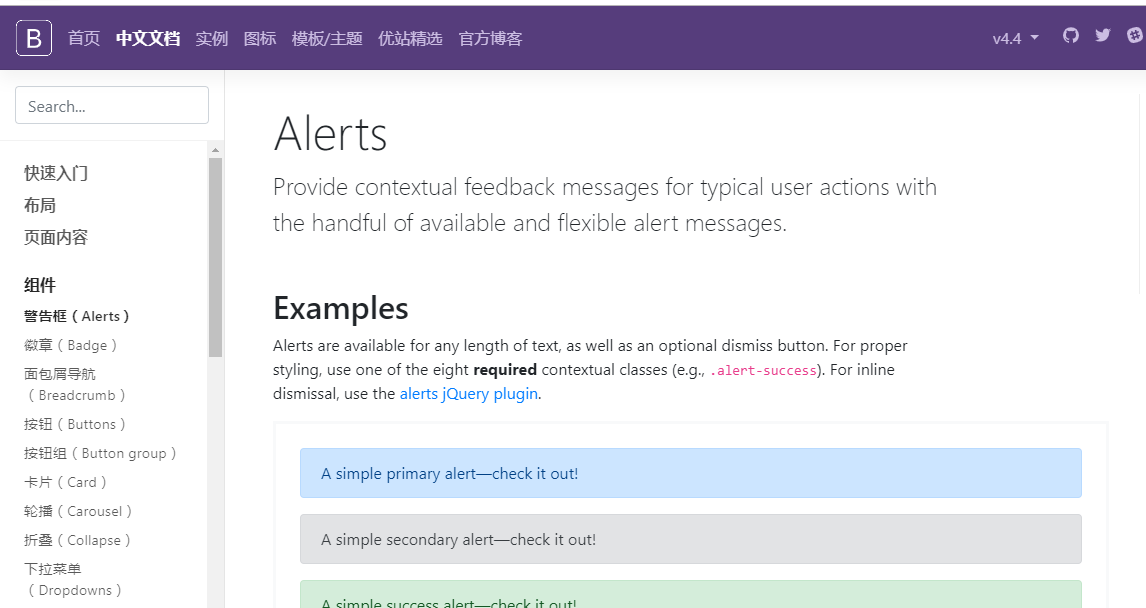
# 实战：多种页面布局

* 现有优秀前端页面布局展示&下载&引用
* Bootstrap网站中，有许多现存优秀前端网页，可以直接引用或修改，引用方法：

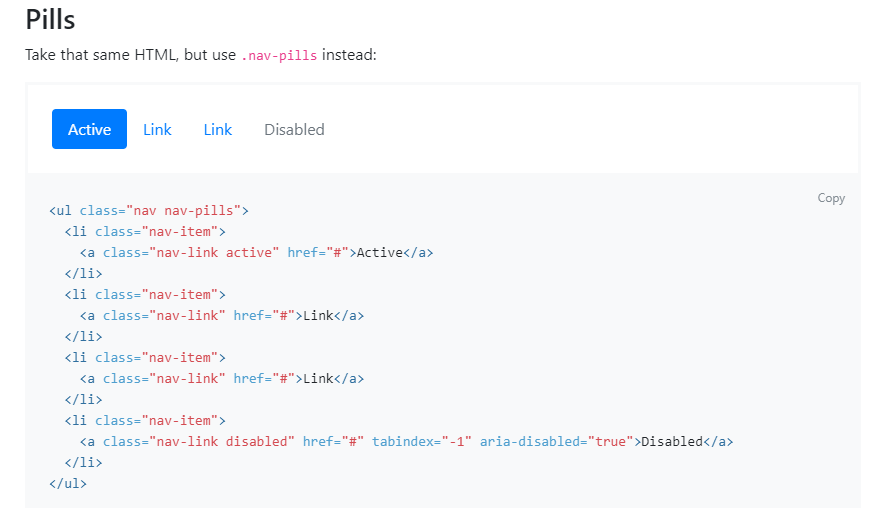
1. 进入网站，点击Bootstrap4中文文档



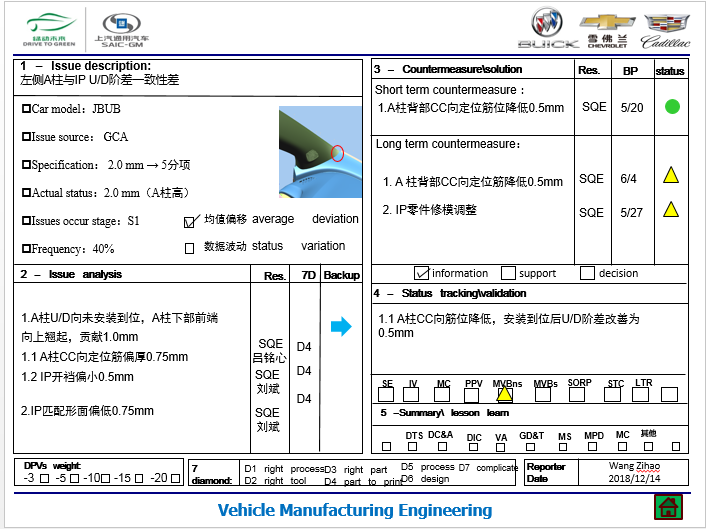
2. 文档左侧有许多页面现有现有标签模块，右侧是对应的案例和示例代码



3. 将示例代码copy后放入Django的Html文件中即可，设置好对应的Views,Url运行



# 实战：一页纸报告



# ERROR

* 项目urls.py运行时报错

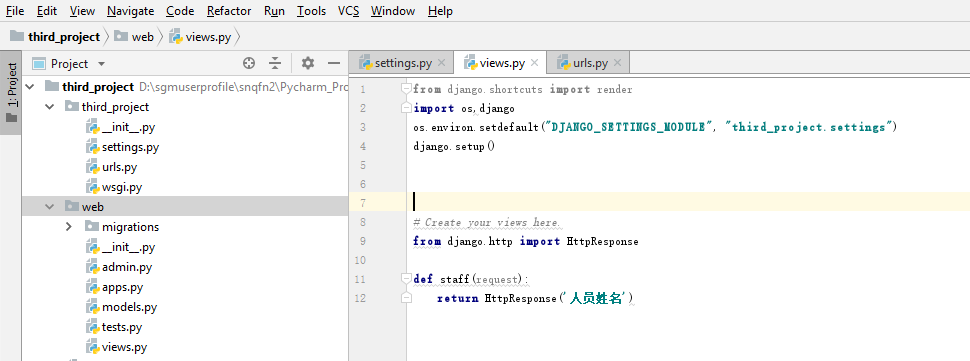
Error：django.core.exceptions.ImproperlyConfigured: Requested setting INSTALLED\_APPS, but settings are not configured. You must either define the environment variable DJANGO\_SETTINGS\_MODULE or call settings.configure()

【不合理的配置，必须定义环境变量DJANGO\_SETTINGS\_MODULE，或者调用settings.configure()】

ERROR：Apps aren't loaded yet

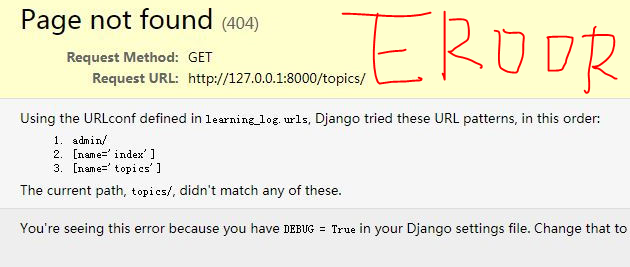
解决方法：在views.py中写入即可

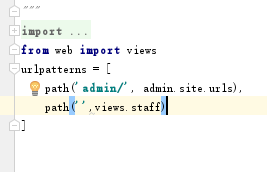
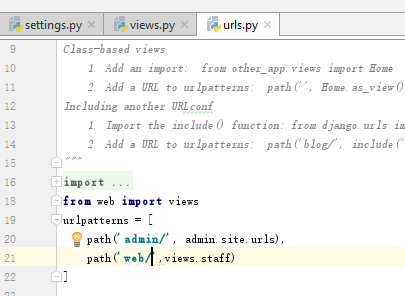
*import os,django  
os.environ.setdefault("DJANGO\_SETTINGS\_MODULE", "project name.settings")  
django.setup()*



* ERROR：Urls.py 映射后网页打开显示404

原因：path的路径设置问题，创建一个函数，path（）首个参数置空即可





‘path第一个参数置空后成功

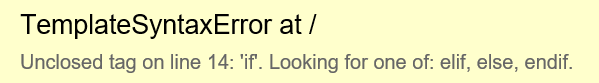
‘web’是app名字

Error：AttributeError: 'WSGIRequest' object has no attribute 'get'

Reason：get函数需要全字母大写GET，而不是Get



Error: HTML模板中引用if时出错



Error: 找不到css文件



Solution: <https://blog.csdn.net/zxyhfdl/article/details/81907021>

1. 创建static文件夹-css –js

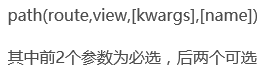
2. 配置 Setting文件

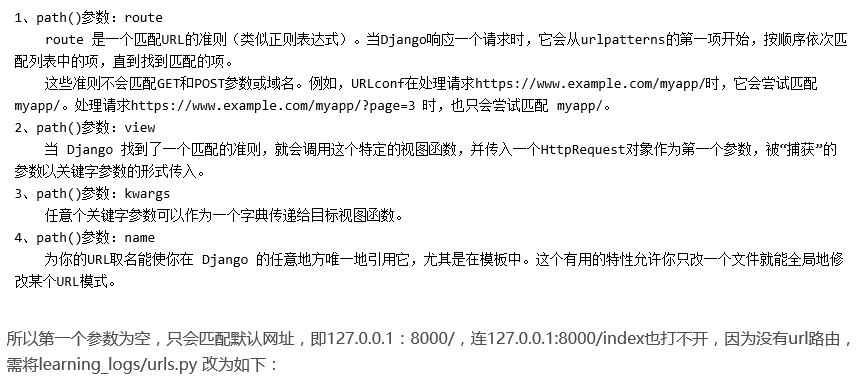
3. html增加“{% load staticfiles %} “

4. 引用方法 <link rel="stylesheet" type="text/css" href="{% static 'css/lessons.css' %}"/>

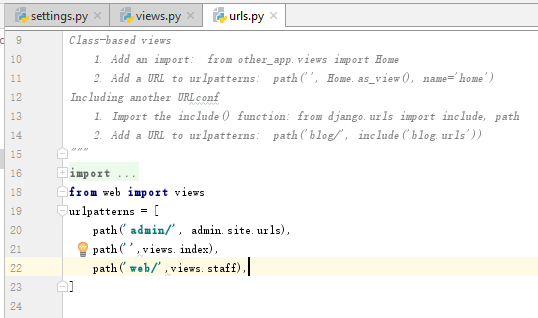
# 知识点

* 知识点：path()函数——详见https://blog.csdn.net/robator/article/details/102313804





* 知识点：Views 和 url的映射理解



在views中定义好相关的函数，在url网站中去访问，使用path()来建立映射关系时：

path(‘web/’,views.staff)

1. ‘web/’是路径设置的多层目录的一个，可以理解为url网址中 <http://127.0.0.1:8000/web/>

指向了下一层目录的引导。对应的views中是view.staff函数，这层目录下，页面显示staff()函数的东西

2. ‘’如果第一个参数置空，则当网址8000/结尾后无下一层引导，页面则显示对应的index()函数的东西

网页Urls由多个app组成，每个app 有自己的views视图，通过path（）的指示，网站打开不同的目录层，显示不同的页面内容。

* 知识点：Pycharm 调节编辑器字体：File-Settings-Editor-Font
* 知识点：cms——ContentManagementSystem，内容管理系统，有很多基于模板的优秀设计，能够加快网站开发的速度，减少开发成本，不能能够文本处理，还能够对图片、动画、声音、图像等进行处理，功能很强大。**网站后台**其实就是CMS
* 主流浏览器及其内核：(浏览器包含 shell 和 内核)

IE trident

Firefox Gecko

Google Chrome webkit/blink

Safari webkit

Opera presto

* [www.baidu.com](http://www.baidu.com) 是域名，通过dns 解析——物理地址 IP 192.168.——向百度索取文件——百度的html和css文件，下载百度的副本到自己电脑上去执行——同时下载html和CSS （异步）——边下载边渲染
* CSS字典网站 www.css88.com
* 标签 ： target， element 也叫元素