**笔记整理**

**移动端布局，拿到640px/750px的设计稿后，元素宽高都要除以2方能适配**

## css3前缀

|  |  |  |
| --- | --- | --- |
| 浏览器 | 内核 | 前缀 |
| Chrome，Safari | Webkit | -webkit- |
| Opera | Presto | -o- |
| IE | Trident | -ms- |
| Firefox | Gecko | -moz- |

<http://www.iconfont.cn>字体图标

注意：页面制作先布局，先确定好每个div的区域

\*在做作业面之前，会先设置一个all的div用来控制整个页面，会设置min-width，max-width min-height max-height等属性

\*在写样式之前，先创建一个base.css样式用来写统一样式

## 属性过滤器

**ie6不兼容子代选择器**

\_下划线属性过滤器 div｛\_属性：属性值；属性：属性值｝：符合标准的浏览器不会识别加了下划线的的属性，而忽略了这个声明，但ie6以下的浏览器会继续解析这个属性。Height在ie6类似与min-height的功能，这个是区分ie6和其他浏览器的方法

！important过滤器div｛属性：属性值！important；｝：被浏览器有限识别，但ie6不识别

\*属性过滤器：div｛\*属性：属性值；｝:添加了\*的属性能被ie7以下的浏览器识别，其他浏览器会忽略

\9：ie版本识别，其他浏览器不识别div｛属性：属性值\9;｝

\0：ie8以上的版本识别，其他浏览器不识别div｛属性：属性值\0;｝

## 页面开始先写CSS样式

\*｛margin:0；padding:0；｝

body，html｛height:100%；｝

ul，li｛listy-style：none；}

a｛text-decoration：none；｝

clearfix：：after｛display：block；content：“”；overflow：hidden；clear：both；height：0；｝

**line-height=上边距+字体大小+下边距**

**line-hight：百分比（根据字体大小来变）**

## 页面

<!DOCTYPE html>

<html lang=”en”>

<head>

<meta charset=”utf-8>

<title></title>”

</head>

<body>

<div id=”header”></div>

<div id=”banner”></div>

<div id=”main”></div>

<div id=”news”></div>

<div id=”link”></div>

<div id=”footer”></div>

</body>

</html>

## 创建站点

文件夹

images

Javascript

Html

Css

Index.html

## html标签

1、<h1></h1>语义化最重，一个页面只能有一个”h1”标签

2、<p></p>

3、<hr/>

4、<br/>

5、视觉化标签 语义化标签

<b></b> <strong></strong> 加粗

<u></u> <ins></ins> 下划线

<i></i> <em></em> 倾斜italic

<s></s> <del></del> 删除线

6、<pre></pre>

7、<ul>

<li></li>

</ul>

<ol>

<li></li>

</ol>

<dl>自定义列表

<dt></dt>标题

<dl></dl>解释

</dl>

8、<a href=”链接地址” target=”\_blank/\_self”></a>

9、<img src=”图片地址”alt=”图片挂了显示的文字”title=”图片主题”/>

10、<table border=””cellspacing=””cellpadding=””bgcolor=”颜色”height=””width=””align=””>

<tr>

<td valign=””rowspan=””colspan=””></td>

</tr>

</table>

11、<form action=”提交地址”method=”get/post”name=””>

<input type=”text” value=”文本框内默认显示的文字”maxlength=”字符长度”/>

<input type=”radio”name=””（名字要一致） checked/>

<input type=”checkbox”name=””（名字要一致）checked/>

<select name=””>

<option value=”名字” selected>名字

</select>

<input type=”submit”value=”按钮上面的文字”/>

<input type=”reset”value=”按钮上面的文字”/>

<input type=”button”value=”按钮上面的文字”/>

<input name="'" type="button" value=“按钮内容” />

（他和submit的区别是 ，submit是提交按钮 起到提交信息的作用，button只起到跳转的作用，不进行提交。）

<textarea cols=”数值”rows=”数值”></textarea>

</form>

12、<div></div>

13、<span></span>

块级元素独占一行，内联元素由内容撑开宽高

### CSS运用三种方式

内嵌 <标签 style=”css样式;”></标签>

内联 <style>css样式</style>

外联 <link rel=”stylesheet”href=”css样式表”>

@import url(“css样式表”)

@import和外联的区别

1、@import在CSS里面，link由html提供

2、在加载页面的时候，link会在加载的时候同时导入CSS样式，而@import会在加载完成后再导入CSS样式

3、link不存在浏览器兼容性问题，@import要ie5以上才能被识别

4、link的权重大于@import

## CSS选择器

标签选择器 p{}

类选择器 .p{}

id选择器 #p{}

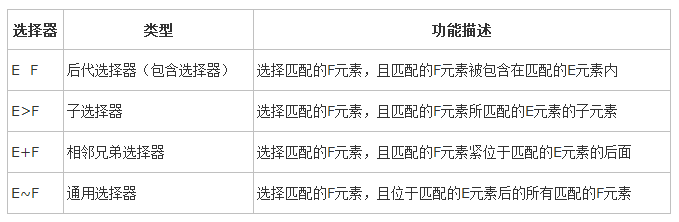
通配符 \*{}

群组选择器 p,div,span

后代选择器 div p span

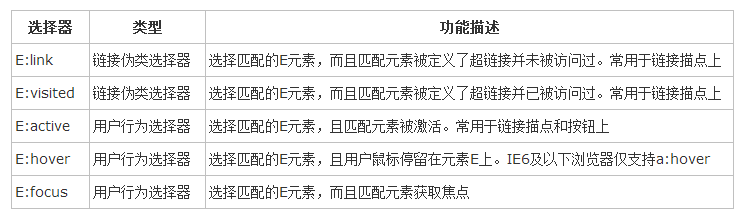
交集选择器 p.类名

### 伪类选择器

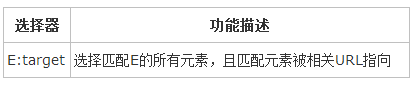


~选择元素下边的所有兄弟节点

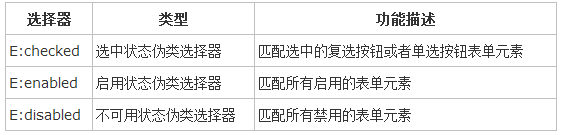
注意:兄弟选择器选中的是【E标签之后】的所有兄弟F



**hover和active可以用在其他标签**



匹配到刚好是目标的元素



input[type="checkbox"]:checked+span{background:red;}

<input type="checkbox"><span>span1</span>

### 结构伪类选择器

E：first-child

满足为第一个元素，同时要满足为指定类型的标签

E：last-child

最后一个子元素，同时要满足为指定类型的标签

E：nth-child(n)

E:nth-child(13)，从一开始数的第13个元素

索引从1开始计算，第n个子元素，同时要满足为指定类型的标签

nth-child(2n)：偶数

nth-child(2n-1)：奇数

nth-child(-n+5)：0~5/(-n+7就是0到7)

nth-child(even/odd): 所有的偶数/奇数

E:nth-last-child(n):同E:nth-child(n) 相似，只是倒着计算

E：nth-of-type(n)

指定了类型，在计算的时候只从指定的类型开始计算索引。

注意：区分nth-child与nth-of-type

E:nth-last-of-type(n/odd/even/2n+1) 倒着数

First-of-type=nth-of-type（1）

E：empty

选中没有任何子节点的E元素【空格也算子节点】

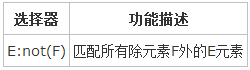
E：only-child

其父元素只有唯一一个节点，且唯一节点要满足为该类型

E：only-of-type

其父元素的该类型的节点只有一个，选中该节点

### 否定伪类选择器

取反

### 属性伪类选择器

E[attr] 拥有attr属性的该元素被获取

E[attr=value]拥有attr属性，且属性值等于value的该元素被获取到

E[attr\*=value] 拥有attr属性，且属性值包含value的该元素被获取

E[^=value] 拥有attr属性，且属性值以value开头的该元素被获取

E[$=value] 拥有attr属性，且属性值以value结尾的该元素被获取

## 权重

！important>行内样式>id选择器>类选择器>标签选择器>通配符>继承

权重计算 对于同一个元素 0 0 0 0

第一个0 行内样式

第二个0 id选择器个数

第三个0 类选择器个数

第四个 0 标签选择器的个数

## CSS属性

**font-size**

**font-family**

**color**

**font-weight：**bold/bolder/normal;

**font：字体粗细 字体大小/行高 字体样式**

**text-transform**：none/capitalize首字母大写/uppercase全部大写/lowercase全部小写

**font-style**：italic斜体/obtique文字倾斜/normal

水平对齐方式{text-align:left/right/center/justify;}

垂直对齐方式{vertical-align:top/bottom/middle;}

（在一个盒子里面，想要img和文本垂直三种方式对齐，就在img标签里面用vertcal-align）只能用在行内元素

**line-high**

**text-decoration**：none/underline/line-through/overline

**text-indent**

**letter-spacing/word-spacing**

**list-style-type**：disc(实心圆)/circle（空心圆）/square(实心方块 )none(去掉列表符号)；

**List-style-image:url(“”)**

List-style-position:inside/outside

关于列表属性的简写：list-style: square inside;

border：1px solid red;

solid:实线，dashed:虚线，dotted:点状线，double:双线，none:没有边框

### background

**background-image:url(“”)**

background-position: {水平方向属性值 垂直方向属性性;}

水平方向值：left/center/right或数值

垂直方向值： top/center/bottom或数值

向右、向下是负值，精灵图定位一般是负值

**background-size:cover**（会覆盖整个盒子，一般与background-position：center一起使用）/contain（可能会留白）/percentage/length数值 width数值;会控制整个背景画布的大小

**background-color**

**background-repeat**：no-repeat/repeat-x/repeat-y

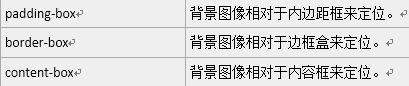
**background-attachment**：scoll/fixed（固定）

语法：选择符{background：属性值1 属性值2 属性值3;}

**background-origin:**

background-origin 属性规定 background-position 属性相对于什么位置来定定位。

background-origin: padding-box|border-box|content-box;



**background-clip:**

设置最终显示那些区域。

## margin的塌陷现象

每个大盒子里面的第一个盒子，添加margin-top的话会出现塌陷现象

解决方法：

1 给父元素加个边框

2 在父元素overflow：hidden 可能会让容器某个部分显示不正常

3 在父元素适用padding-top

4 浮动

## 清除浮动

父元素没有高度的话，子元素浮动，父元素会塌陷

解决方法：

1 给父元素添加声明overflow:hidden;(触发一个BFC)

2 在浮动元素下方添加空div,并给该元素添加声明：

div{clear:both; height:0; overflow:hidden;}

3 万能清除浮动法

选择符：：after｛

overflow：hidden；生成一个BFC

clear：both；清楚浮动

visibility：hidden；不希望清楚浮动被看到

display：block；将after变成块级元素

content：“”；不需要内容

｝

## position

相对定位position：relative；

绝对定位position：absolute；

1 如果绝对定位元素没有父元素或者是所有的父元素都没有定位，那么将来top left rigt bottom是相对于html定位的；

2 如果绝对定位的元素有祖先，而且祖先有定位（非static），那么这个绝对定位的元素偏移是以自己的有定位的祖先为基础。

静态定位position: static;

固定定位position: fixed;定位相对于浏览器可视区域而定，也脱离了文本流

定义元素为包含块（containing block）：给绝对定位元素的父元素添加声明position：relative；

Example:

<div style=”position=relative”>

<div style=”position=”absolute”></div>

</div>

z-index：value/auto；数值越大越在上面，此属性仅仅作用在position上面，可用正数，也可以用负数

相对定位和绝对定位的区别：

1、参照物不同，绝对定位的参照物是包含块（已定位的父元素），相对定位的参照物是元素本身默认的位置；

2、绝对定位将对象从文档流中拖离出来因此不占据空间，相对定位不破坏正常的文档流顺序无论是否进行移动，元素仍然占据原来的空间。

**中点**

**top/left/right/bottom：50%**

**margin-top/margin-bottom：-height.value**

**margin-left/margin-right：-width.value**

**中点定位尽量使用这个办法**

## 元素

### 伪元素

伪元素 ::before 和 ::after 一般和content一起使用

div::before{content:url();content:””;}

::first-letter 定义对象第一个元素的首字母样式，只能用于块级元素

::first-line 定义对象第一个元素的首行样式，只能用于块级元素

如果要在ie9以下的浏览器适用，将其变成伪类使用

：：before→：before

：：selection

### 块级元素、行内元素、行内块元素

浮动了的元素相当于行内块级元素

**float：left/right：浮动的位置不占行，可看作行内（块级）元素，浮动的元素虽然脱离了标准流，但还是占据了文本流**

1 display:block;块级元素 !!!<li>标签的display:list-item

2 display：inline行内元素 vertical-align必须作用在行内元素

3 <img><input><textarea>dispay:lnline-block 行内块级元素

行内元素不能设置宽高，且margin-top/bottom padding-top/bottom对其无效

置换元素，根据属性改变标签内容，如<img>

## overflow：hidden/visible/scroll/auto(滚动条应用)

隐藏/显示/无论是否超越都显示滚动条/超越才显示滚动条

属性：

overflow 水平及垂直方向内容溢出时的设置

overflow-x 水平方向内容溢出时的设置

overflow-y 垂直方向内容溢出时的设置

一边overflow-x：hidden，overflow-y：scroll；另外一边会自适应。

设置滚动条

scrollbar-face-color(立体滚动条凸出部分的颜色)

scrollbar-highlight-color(滚动条背景条的颜色)

scrollbar-base-color(滚动条背景的亮光色，基底)

scrollbar-arrow-color(上下按钮三角箭头的颜色)

scrollbar-shadow-color(立体滚动条阴影的颜色)(滑动滚动条边框色,ie显示)

scrollbar-dark-shadow-color(立体滚动条强阴影的颜色（浏览器不显示）)

## 文本溢出text-overflow

属性 1、值：clip 无省略号 ellipsis 省略号

2、配合width和overflow:hidden和white-space:nowrap（不换行）一块使用

## text-shadow

可以写多个阴影，用逗号隔开

让单词强制换行 word-wrap：break-word

使用自定义字体 @font-face｛font-family：“起名字”；src：url（链接）；｝之后方可使用

<http://www.iconfont.cn>字体图标

## marquee滚动字幕

<marquee></marquee>

属性

behavior scroll滚动/alternate来回滚动/slide滑动一次

direction up/down/left/right

scrollamount滚动速度=”value”

height=”value”

width=”value”

loop循环次数

scrolldelay 停顿时间

vallign left/center/right/

bgcolor 运动区域的背景颜色

## box-shadow

属性

h-shadow 水平方向阴影，正负值调节左右

v-shadow 垂直方向阴影，正负值调节上下

blur 阴影模糊区域

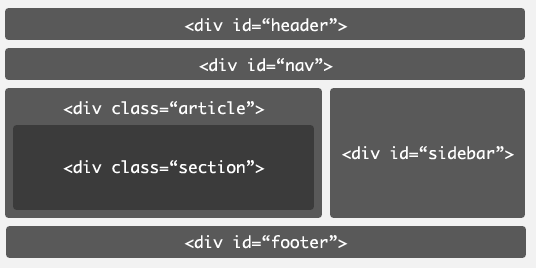
spread 阴影的扩散面积尺寸

color 阴影的颜色

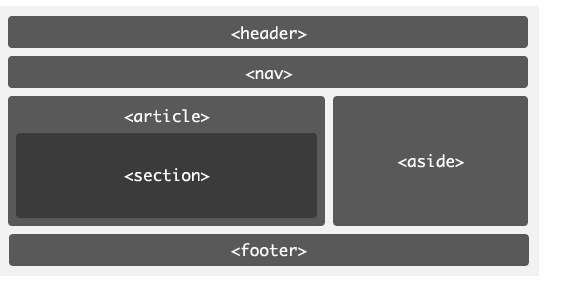
inset 内阴影，outset为外阴影

## Html5新标签（一般在移动端使用）

旧html写法



html5语义化



语义化标签只兼容IE9以上的浏览器



### <header></header>标题

里面必须有标题标签<h1></h1>----<h6></h6>

如果存在多个标题，可以适用**<hgroup></hgroup>**包含

Example：

<hgroup>

<h1></h1>

<h2></h2>

</hgroup>

<header> 标签不能被放在 <footer>、<address> 或者另一个 <header> 元素内部，但它不局限于写在网页头部，也可以写在网页内容里面如<article></article>。

header提供页眉信息尝试导航一类的信息

header里面至少包含一个标题标签

### <main></main>主体

一个个页面最好只是用一次

### <article></article>文章

article里面可以嵌套article，例如，一篇博客的文章，可以用article显示，然后一些评论可以以article的形式嵌入其中。

### <section></section> 专题

标签定义了文档的节。比如章节、头部、底部或者文档的其他区域，它表示一段专题性的内容，通常由内容及其标题组成。区块

### <aside></aside>

侧边栏，文章链接或者广告，作者简介

### <figure></figure> 群组

<figcaption></figcaption> 对<figure>的内容进行描述

<figure>用于对元素进行组合。一般用于图片，文字组合。

<figcaption>是 figure的子元素，用于对figure的内容进行说明

### <nav></nav>

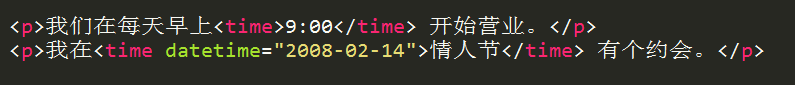
### <footer></footer>

眉脚一般会包含作者姓名、文档版权信息、使用条款链接、联系信息等. （可多个）

### <time></time>

表示时间

datetime属性

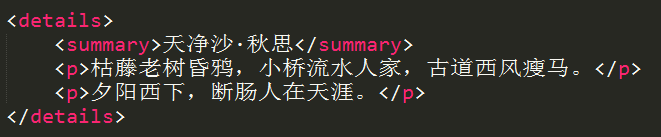


### <details></details>

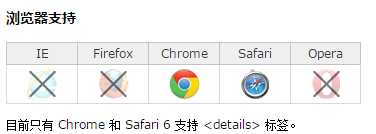
用于描述文档或文档某个部分的细节。

<summary> 可以为<details>定义标题。标题可见，用户点击标题时，会显示出 details中的内容。任何形式的内容都能被放在 <details> 标签里边。

open属性,规定 details中内容是否默认可见。<details open="open">



浏览器支持



### <mark></mark>

定义带有记号的文本，在需要突出显示文本时使用 <mark> 标签。可改变背景颜色

### <progress></progress>

max属性最大限度

value定义当前值

### <meter></meter>

属性

high 定义的值界定为较高的值

low 定义的值界定为较低的值

max 定义最大值，默认是1

min 定义最小值，默认是0

optimum 定义最佳的值，比high高意味着越高越好，比low低，意味着越低越好

value 定义当前值

### <ruby></ruby>

<ruby> 标记定义 注释或音标

<rt> 标记定义对ruby的注释内容文本

<rp> 告诉那些不支持 Ruby元素的浏览器如何去显示

Example

<ruby>垚<rp>(</rp><rt>yao</rt><rp>)</rp></ruby>

### 使语义化标签兼容IE8以下，需要加载JS脚本

可以引用一个js插件解决HTML5语义化标签在IE6-8不兼容问题。

<!--[if lt IE 9]>

<script src="路径 /html5.js"></script>

<!--[endif]-->

下载地址：<http://www.bootcdn.cn/html5shiv/>

HTML5在默认情况下表现为内联元素，对这些元素进行布局我们需要利用CSS手工把它们转为块状元素方便布局

<style>

article, aside, canvas, details, figcaption, figure,

footer, header, hgroup, menu, nav, section, summary

{

display: block;

}

</style>

## 重新定义语义化标签

|  |  |
| --- | --- |
| HTML元素 | HTML5中的意义 |
| <b> | 代表内联文本，通常是粗体，没有传递表示重要的意思 |
| <i> | 代表内联文本，通常是斜体，没有传递表示重要的意思 |
| <dd> | 可以同details与figure一同使用，定义包含文本，dialog也可用 |
| <dt> | 可以同details与figure一同使用，汇总细节，dialog也可用 |
| <hr> | 表示主题结束，而不是水平线，虽然显示相同 |
| <menu> | 重新定义用户界面的菜单，配合commond或者menuitem使用 |
| <small> | 表示小字体，例如打印注释或者法律条款 |
| <strong> | 表示重要性而不是强调符号 |

## form新增标签

<fieldset align=“left/center/right/justify”></fieldset>表单字段级，自动生成边框

<legend></legend>字段级标题，必须为fieldset里的第一个子元素

将label放在checkbox外层，就可以不用写name

<input type=”image”src=””wdith=””height=”” alt=””>可作为提交按钮

**<datalist>**

与<input>配合,实现类似于拥有输入功能的下拉列表。<input> 元素的 list 属性连接<datalist>的ID，来关联 <datalist> 元素。

Example

<form action=””name=””method=””>

<input type=”text”list=”fruits”>

<datalist id=”fruits”>

<option value=”苹果”>苹果</option>

<option value=”榴莲”>榴莲</option>

<option value=”香蕉”>香蕉</option>

</datalist>

</form>

**<output>**

**<output> 标记定义一些输出类型,计算表单结果配合oninput事件**

<form oninput="res.value=no1.value\*no2.value” >

<input type="text" name="no1">

<input type="text" name="no2">

<output name="res"></output>

外部的input标签可以通过form的属性关联到某个form的标签

<form action=””name=””method=””id=”text”>

<input type=”text”>

<input type=”reset” form=”text”>

</form>

## form新增类型

<form action="#">

<!-- type -->

颜色：<input type=**"color"**><br>颜色选择器

邮箱：<input type=**"email"**><br>

数字：<input type=**"number"**><br>

电话号码: <input type=**"tel"**><br>

网址：<input type=**"url"**><br>

搜索：<input type=**"search"** results="n"（出现放大镜图标）><br>

<!-- step 步长 -->

范围：<input type=**"range"** max="100" min="0" value="50" step="50"><br>

特定范围内的数值选择器，min、max、step( 步数，一次移动的数值 )、value

日期：<input type=**"date"**><br>

月：<input type=**"month"**><br>

周：<input type=**"week"**><br>

时间：<input type=**"time"**><br>

<input type="submit">

</form>

<form>

## form新增属性

！！！只有设置了name属性的表单才能上传到服务器

<label for=“id名字相同”>用户名</label>

<input type=”text”placeholder=”请输入文字”>占位符

</form>

利用label for将文字和控件关联起来

**required** 必填

Example

<input type=“password” required>

**autofocus**一般用在type=“text”后面，自动聚焦

Example

<input type=”text” autofocus>

**mutiple**

文件上传多选或多个邮箱地址

Example

<input type=”file” name=””mutiple>

**pattern正则**

<input type=”text”pattern=””验证字段>

## 插入视频音频

1 视频<video src=””></video>视频格式后缀为ogg 音频<audio src=””></audio>

2 <video>

<source src=”路径”type=”video/ogg”）（type属性必须添加）>

</video>

width和height设置视频宽高

autoplay 视频就绪自动播放

controls 向用户展示播放控件

poster 显示等待加载的图片poster=”路径”

loop 循环播放

## 新增表格样式

rules

rules="groups/rows/cols/all/none" 添加组分隔线

说明：

rows:位于行之间的线条

cols:位于列之间的线条

all：位于行和列之间的线条

none:没有线条

groups:位于行组和列组之间的线条

列分组

**<colgroup span/\*列数\*/="分组列数"></colgroup>有多少组就写多少次**

**<col span/\*列数\*/="分组列数">**

**区别：**

ruls对col span不起作用

行分组

<thead></thead> 表头

<tbody></tbody> 表体

<tfoot></tfoot> 表尾

说明：一个Table中，只能包含一个thead,一个tfoot,但可包含多个tbody。

border-spacing：value

border-collapse：collapse/separate合并相邻单元格边框 (该属性必须给table添加)

empty-cells：show/hide单元格无内容是否显示边框

<caption></caption>表格标题

caption-side:top/left/right/bottom (left/right只有火狐能用，top bottom在**ie6**以下

不能识别，默认top)

## 边框圆角

border-radius：100px/100px，X轴，Y轴，（50%是圆）

顺序：border-top-left-radius/border-bottom-right-radius

Border-image-source：默认放在边框的四个角落

Border-image-slice

4个值上右下左 3个值 上右下（左随右） 2个值 上右（下随上，左随右） 1个值全部

border-image-width 边框图片宽度

border-image-outset 边框图片向外偏移

border-image-repeat：**stretch**（拉伸）**repeat**（重复）**round**（铺满）

## 渐变

渐变不兼容火狐和谷歌需要加前缀-moz—webkit-

gradient渐变 linear线性只有角度是百分比颜色

属性值gradient（linear，起点位置，结束位置，开始颜色，结束颜色）

例如background-webkit-gradient（linear，0（水平方向） 0（垂直方向），0 100%，from（red），to（blue））；

顺序：left top；right bottom

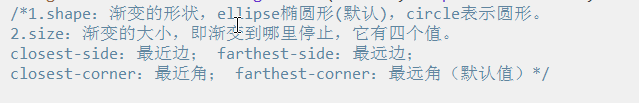


gradient（linear，0 0，100% 100%，color-stop（0.5，red），color-stop（0.5，yellow），color-stop（1，green））；取值范围0-1

linear-gradient(135deg(角度),red 百分比，blue 百分比)

径向渐变先确定圆心

-webkit-radial-gradient(50%（水平方向）,50%（垂直方向）,ellipse（形状） closest-set（形式），颜色 百分比，颜色 百分比);



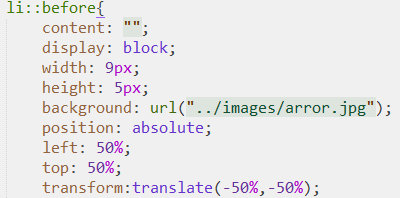
重复线性渐变Repeating-linear-gradient（角度，颜色 百分比，颜色 百分比）

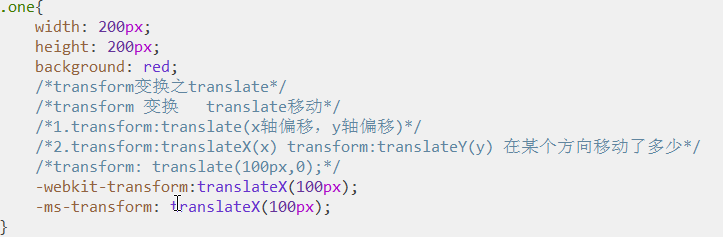
重复径向渐变Repeating-radial-gradient（角度/left top。。。,ellipse（形状） closest-side（形式），颜色 百分比，颜色 百分比）

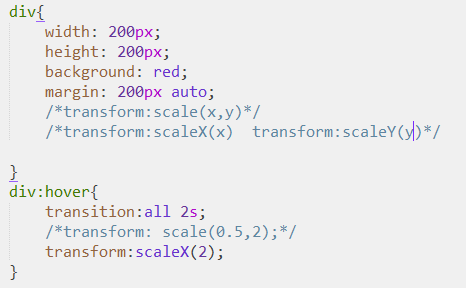
要加浏览器前缀保证兼容性

## transform

transform translate的用法 中点

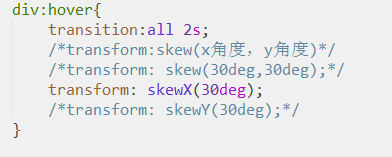




transform scale（X，Y）数值倍数

transform rotate（角度）：正值顺时针旋转，反之逆时针旋转

transform skew（X角度，Y角度）：改变元素的形状



transform-origin:(0,0)改变变换的中心点

## 盒模型

Box-sizing：border-box（边框内都是width）（怪异模式）/content-box（默认）



## 3D变换

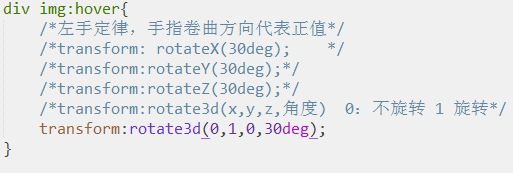
tramsform-style：flat(默认保持平面变换)/（preserve-3d（保持3D变换，加在父元素上面））

tramsform：translate3D（X,Y,Z）/translateX(X)/ translateY(Y)/ translateZ(Z)

perspective：value设置三维透视距离（观看距离），加在父元素上面

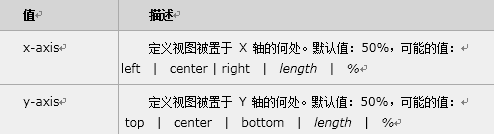
tramsform：scale3D（X,Y,Z）/scaleX(X)/ scaleY(Y)/ scaleZ(Z)

tramsform：rotate（X,Y,Z，deg）



perspective-origin：站在哪个位置看





立体物体需要设置透明度才能看到后方

## 动画animation

**动画里面的属性需要上下对应**

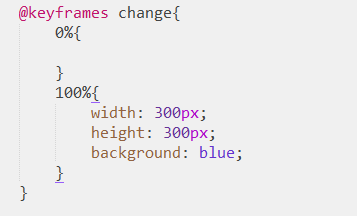
步骤

（1）通过@keyframes指定动画序列；

（2）通过百分比将动画序列分割成多个节点；

（3）在各节点中分别定义各属性

（4）通过animation将动画应用于相应元素





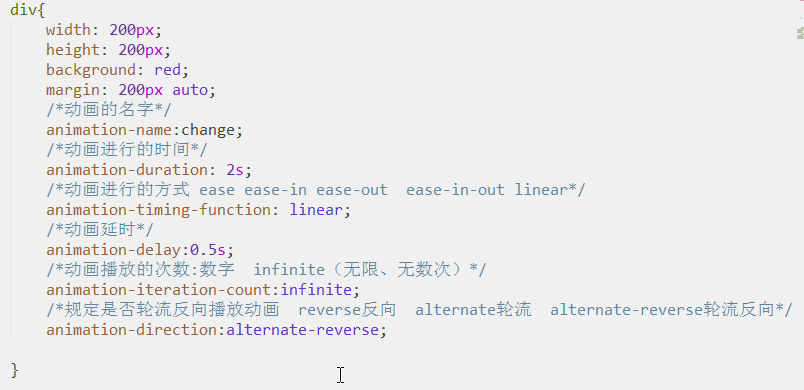


简写

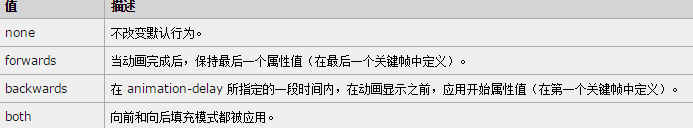


direction:reverse（反向播放）alternate（重复播放）alternate-reverse（反向重复播放）

animation-play-state:paused; 暂停动画



animation-fill-mode：forwards





## 移动端-CSS弹性盒布局flex-box：定固高不定固宽

具有兼容性，默认不换行，子元素会自动拉伸

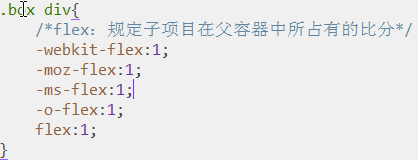
display：-webkit-flex；

display：-moz-flex；

display：-ms-flex；

display：-o-flex；

display：flex；



**flex规定子项目在父容器中所占有的份额**

默认主轴是水平方向的轴，侧轴是垂直方向的轴，两个均可改变

**flex-direction**：column（垂直方向）/column-reverse（垂直方向的反向）/row（水平方向，默认值）/row-reverse（水平方向的反向）**父容器**

改变主轴方向，写在容器上面

**侧轴**如果没有被定义长度，则会默认被拉伸

**flex-wrap**：wrap（换行）/nowrap（不换行，默认值）/wrap-reverse（换行后反转，主轴水平时左右反转，垂直时上下反转）

**flex-flow**：flex-direction||flex-wrap两个值同时定义或单独定义都生效

**justify-content**主轴对齐，主要用来定义伸缩项目沿主轴线的对齐方式；**父容器**

flex-start：伸缩项目向一行的起始位置靠齐；

flex-end：伸缩项目向一行的结束位置靠齐；

center：伸缩项目向一行的中间位置靠齐；

**space-between**：伸缩项目会平均的分布在行里；

**space-around**：伸缩项目会平均的分布在行里，两端保留一半的空间；

**align-items**侧轴对齐，要用来定义伸缩项目沿侧轴线的对齐方式，每个项目在自己的位置上对齐；**父容器**

flex-start：伸缩项目在侧轴起点边的外边距 紧靠住 该行在侧轴起始边；

flex-end：伸缩项目在侧轴终点边的外边距 紧靠住 该行在侧轴终点边；

center：伸缩项目的外边距盒 在该行的侧轴上居中放置；

**baseline**：伸缩项目根据伸缩项目的基线对齐；

**stretch**：伸缩项目拉伸填充整个伸缩容器。（如果没有给高度，那么侧轴默认就是拉伸状态）

**align-self**，控制单个项目在侧轴的对齐方式，属性与align-item一致，可覆盖align-item。

align-content多行侧轴对齐，与justify-content相对，每个项目会从父容器的位置开始对齐，与换行一起使用flex-wrap：wrap；**父容器**

flex-start：伸缩项目向一行的起始位置靠齐；

flex-end：伸缩项目向一行的结束位置靠齐；

center：伸缩项目向一行的中间位置靠齐；

**space-between**：伸缩项目会平均的分布在行里；

**space-around**：伸缩项目会平均的分布在行里，两端保留一半的空间；

**stretch：拉伸**

**伸缩项目属性**

**显示顺序 order：**默认状态是按照标准流的顺序排列，在flexbox模型里，可以通过order改变伸缩项目的顺序。

不定义order的伸缩项目会排到前面

order：1； 排第一

order：2； 排第二

## 移动端-CSS3多列布局

核心属性：

**columns：column-width 列宽 column-count 列数**

column-width：**定义每列列宽； 类似于最小宽度min-width；** auto 自适应；

column-count：**定义分列列数；最多列数**，auto自适应（由列宽、容器宽和列间距决定），也可固定

**column-gap**：定义列间距； 不能为负数；

**column-rule**：定义列边框；与定义边框一样：2px dashed #ccc;

column-span：定义多列布局中子元素的跨列效果；通常用于标题（意思是此属性写在h2标签里面）；

none：不跨列；

all：跨所有列

## meat标签的定义

许多智能手机都使用了一个比实际屏幕尺寸大很多的虚拟可视区域，主要目的就是让页面在智能手机端阅读时不会因为实际可视区域变形。所以你看到的页面还是普通样式，即一个全局缩小后的页面。

为了让智能手机能根据媒体查询匹配对应样式，让页面在智能手机中正常显示，特意添加了一个**meta**标签。这个标签的主要作用就是让智能手机浏览页面时能进行优化，并且可以自定义界面可视区域的尺寸和缩放级别。

### 三种viewpoint视口

**布局视口：**以屏幕分辨率为基准，实际上布局视口的宽度要比屏幕宽出很多。

**视觉视口：**用户看到的网站展示区域，一般视觉视口和设备宽度一致。并且它的CSS像素的数量会随着用户缩放而改变。

**理想视口：**为了使网站在移动端有最理想的浏览和阅读宽度而设定。需要手动添写meta视口标签，一般视口大小都设置为设备大小。

### 设置移动端标准视口

**快捷键：meta：vp**（要移动端优化适配必须使用此代码）

<**meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0"**/>

width 控制viewpoint的宽度，可以是固定值，也可以是device-width设备宽度

user-scalable:用户是否可以缩放  
initial-scale 控制初始化缩放比例，1.0表示不可以缩放  
maximum-scale 最大缩放比例  
minimum-scale 最小缩放比例

## 媒体查询

### 分界点：

超小屏幕xs （移动设备）768px以下

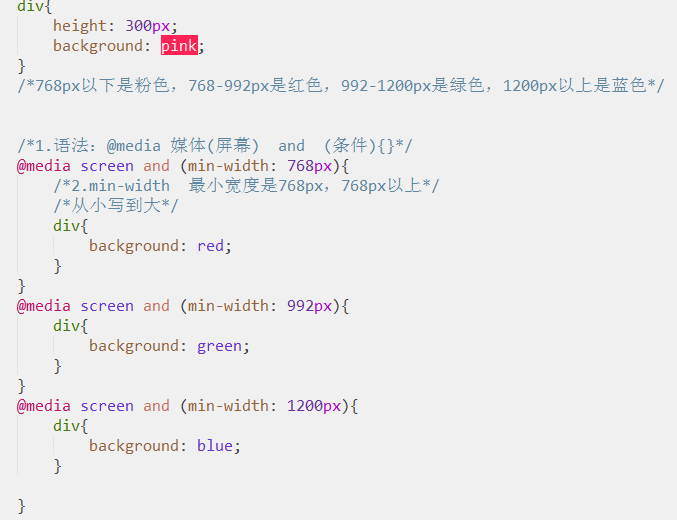
小屏设备sm 768px-992px

中等屏幕md 992px-1200px

宽屏设备lg 1200px以上

### 语法：@media screen and （max-width：768px）｛｝

1. 若当前页面宽度大于min-width，则样式生效，所以媒体查询min-width应从小写到大;
2. 若当前页面宽度小于max-width，则样式生效，所以媒体查询max-width应从大写到小。



min-width：value

**min-device-width：value（设备宽度）**

### 调用不同样式



@media (**orientation**（检查设备是否处于横向）: landscape（横屏）/orientation：portrait（竖屏）)

## 弹性布局（100%布局）——拉勾网、天猫首页

好处：充分发挥大手机的优势——显示内容越多

缺点：屏幕过大，间距过大，比例失调。

特点：

- 顶部与底部的bar不管分辨率怎么变，它的高度和位置都不变；

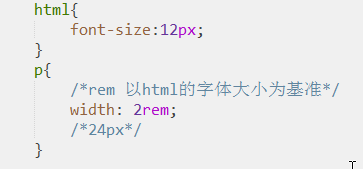
- 中间每条招聘信息不管分辨率怎么变，招聘公司的图标等信息都位于条目的左边，薪资都位于右边。

## rem布局：使得每个页面在不同设备上看起来都基本一致

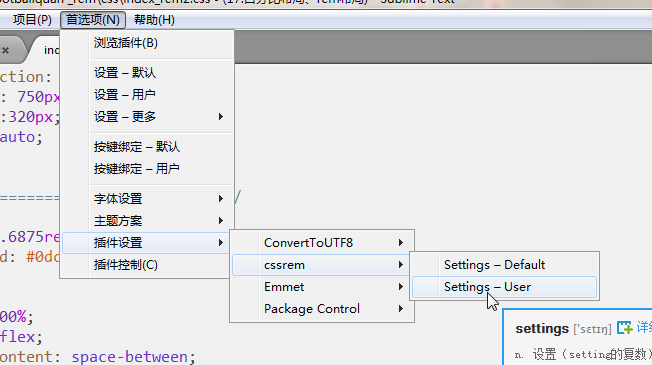
rem以html的字体大小为基准

基于iphone5开发的，640px，字体大小是64px

基于iphone6开发的，750px，字体大小是75px



改变rem基准方法



百分比布局：拿到640px的稿，将所有元素的宽高除以2再做。若也想通过js代码控制，则把html控制那句话删掉，量出来是多少是多少即可。

rem布局一：拿到640px的稿，按照量出来的大小做，然后加上以下代码，将字体以外的px改成rem

<script type="text/javascript">

// 把尺寸放大N倍（N是window.devicePixelRatio）

var wd = document.documentElement.clientWidth\*window.devicePixelRatio/10;

//物理像素\*设备像素比=真实像素

document.getElementsByTagName("html")[0].style.fontSize = wd + "px";

// 把屏幕的倍率缩小到N分之一（N是window.devicePixelRatio）

var scale = 1/window.devicePixelRatio;

var mstr = 'initial-scale='+ scale +', maximum-scale='+ scale +', minimum-scale='+ scale +', user-scalable=no';

document.getElementById("vp").content = mstr;

</script>

**推荐！rem布局二：拿到640px的稿，按照量出来的大小除以2，然后加上以下代码，将字体以外的px改成rem**

**<script type="text/javascript">**

**// 把尺寸放大N倍（N是window.devicePixelRatio）**

**var wd = document.documentElement.clientWidth\*window.devicePixelRatio/10;**

**//物理像素\*设备像素比=真实像素**

**document.getElementsByTagName("html")[0].style.fontSize = wd + "px";**

**// 把屏幕的倍率缩小到N分之一（N是window.devicePixelRatio）**

**var scale = 1/window.devicePixelRatio;**

**var mstr = 'initial-scale='+ scale\*2 +', maximum-scale='+ scale\*2 +', minimum-scale='+ scale\*2 +', user-scalable=no';**

**document.getElementById("vp").content = mstr;**

**</script>**

字体大小通过媒体查询直接手动调整