**默认样式初始化**

**css reset 原则：**

**但凡是浏览默认的样式，都不要使用。**

**body,dl,dd,p,h1,h2,h3,h4,h5,h6{margin:0;font-size:12px;}**

**ol,ul {margin:0;**

**padding:0;**

**list-style:none;}**

**a {text-decoration:none;}**

**img { border:none**

**vertical-align: top（上对齐）;}**

**.clearFix:after { content:"";**

**display:block;**

**clear:both;}**

**.clearFix{zoom:1;}**

**table { border-collapse: collapse; }（合并表格边框）**

**td,th { padding: 0;**

**form { margin: 0; } IE6下 form上有一个margin值**

**input { outline: none;**

**margin: 0;**

**padding: 0**

**box-sizing: content-box; 标准盒模型**

**或box-sizing:border-box 怪异盒模型 }**

**/\* border-width 在各浏览器下不一致，需要根据设计图单独设置一下\*/**

**textarea { margin: 0;**

**padding: 0;**

**resize: none;**

**overflow: auto;**

**outline: none; 轮廓线无}**

**strong { font-weight: normal;**

**em { font-style: normal;**

**mark { background: none;**

**格式**

<!doctype html> /\*css Document\*/

<html>

<head>

<meta charset="utf-8">

<title>课后作业</title>

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

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

</head>

<body>

<div id="box">

</div>

<div style="width:10px "></div>

</body>

</html>

前端开发 前端开发工程师

切页面

代码的复用

交互

用户体验

代码的加载速度

各种终端的适配

网页制作/切图仔/前台

做页面、切页面

---------------------------------------------------------------

产品经理(产品原型) -- UI(页面设计图) -- 前端开发(UI，后端) -- 后端开发 -- 测试 -- 上线发布

---------------------------------------------------------------

语言:

html（Hypertext Markup Language） -- 超文本标记语言 -- 结构

css (Cascading Style Sheets) -- 层叠样式表 -- 样式

js (javascript) -- 行为

--------------------------------------------------------------

工具: 浏览器：

(IE浏览器(IE6,7,8,9,10,11) 标准浏览器: chrome谷歌浏览器，Firefox 火狐,safari 苹果)

切图工具:PS

编辑器: DW/HBuilder/...

-------------------------------------------------------------

<!doctype html> !文档的类型 html -- 文档声明 /\*css Document\*/

<html> -- html的正文开始

<head> 头部 -- 页面头信息

<meta charset="utf-8"> 编码声明

<title>无标题文档</title> 标题 -- 页面标题

</head>

<body> 主体内容 -- 页面的内容

</body>

</html>

-----------------------------------------------------------

<html> 标记、标签

<html></html> 标签对 双标签

<meta /> 单标签

-----------------------------------------------------------

<div></div> 块,区域

标签的属性 属性名="属性值" <div 属性名="属性值"></div>

行间样式--写在标签内部的style属性里 <div style="样式"></div>

写在页面专门的style标签中 -- 内部样式表

id = "idName" #idName{元素的样式}

写在独立的css文件中，通过link标签引入当前页面 --外部样式表

<link href="css的地址" rel="stylesheet" />

边框

单位：px（像素）

width 宽度

height 高度

border 边框 (复合样式,集合样式 每个样式之间已空格隔开)

border-width 边框宽度

border-style 边框样式(solid 实线/dashed 虚线/dotted 点线)

border-color 边框颜色

css中常用的颜色模式:

关键字(red,green,pink,blue,yellow,black...)

rgb();

三位或者六位的十六进制数字

0-9,0-f(0-9,a,b,c,d,e,f)

border-top （right/bottom/left）

上边框(复合样式,集合样式 每个样式之间已空格隔开)

border-top-color

border-top-style

border-top-width

background 背景

background-color 背景颜色

none 无

关键字 # rgb

六位或三位的十六进制数字

background-image: url(图片地址); 背景图 none（无）

image链接

background-repeat 背景图平铺

repeat 平铺(默认值)

repeat-x 横向平铺

repeat-y 纵向平铺

no-repeat 不平铺

background-position 背景图位置

关键字:

x:

left 图片的左侧和元素的左侧对齐 left左侧

center 图片的中间和元素的中间对齐 center中间

right 图片的右侧和元素的右侧对齐 right右侧

y:

top 图片的顶部和元素的顶部对齐 top上侧

center 图片的中间和元素的中间对齐

bottom 图片的底部和元素的底部对齐 bottom底部

具体数值: (设置图片距离元素的左上角的距离)

x:

正值：从元素的左侧开始 从左向右移动

负值：从元素的左侧开始 从右向左移动

y:

正值: 从元素的顶部开始 从上向下移动

负值: 从元素的顶部开始 从下向上移动

缩写：background: #f60 url(img/img.jpg) no-repeat 20px center;

font文字样式

arial英文字体（先英文再写中文的字体样式）

文字相关的设置:

font-style 是否倾斜(italic 倾斜 normal 正常)

font-weight 是否加粗 (bold 加粗 normal 正常)

font-size 文字大小

line-height 行高

font-family 文字字体

**缩写顺序： 加粗 倾斜 大小(必填) 行高 字体 （必填）**

缩写顺序 ：font: font-weight font-style font-size(必填项)/line-height "font-family"(必填项);

**font-family:** "主要字体","备选字体1","备选字体2";

如果用户电脑上有 “主要字体”，就显示"主要字体",如果用户电脑上没有主要字体",那就显示"备选字体1",如果用户电脑没有"备选字体1",就显示"备选字体2";

**font-family:** 英文字体,"中文字体";

英文字符 使用英文字体，中文字符 使用 中文字体;

**单位 em 1em = 1个文字大小**

font: 16px/1 Arial(英文字体); 行高是文字大小的一倍

color 文字颜色

**text-decoration** 文本修饰( line-through 删除线

overline 上划线

underline 下划线 / none 无)

**text-indent 首行缩进**

**text-align 文本对齐方式(left/center/right) (水平对齐方式)**

**vertical-align: middle;(top/bottom)( 垂直对齐方式)**

**letter-spacing 字母间距(文字间距)**

**word-spacing 单词间距(词间距)**

**word-break**： 词内断行

**break-all** 所有的词都可以发生词内断行

英文和数字 没有结束 不会词内断行

**white-space: nowrap**; 强制不换行

**\*宋体情况下，英文字符的宽度，是字体大小的一半**

当有英文字母不能确定文字大小的情况下可以吧行高设成14px/1 就是行高是文字的一倍 去试下文字大小

**a 超链接 <a href="#"></a>**

href为空时 写#（代表当前页面），不能不写东西

href 中写的是个页面地址 -- 点击跳转页面

锚点 href中写的是个id的话，点击之后会移动至此ID所在的位置

/\* 默认情况下 元素会继承 父级对于 文本和文字相关的样式设置 \*/

a 标签的样式细节:

**1. a标签字体颜色不会继承父级**（想要颜色必须重新添加）

**2. a标签默认会有下划线**

标签选择器，可以直接写标签名

a {

color: red;

}

找到页面上，所有的a标签

标签名{} 给页面上某种标签加样式

**伪类： 针对元素的某种状态添加样式**

**同一条样式 后边覆盖前边**

**:link 此链接未被访问过**

**:visited 此链接被访问过**

**:hover 鼠标移入**

**:active 鼠标按下 书写顺序 L-V-H-A （面试会问）**

<a href="https://www.baidu.com" target="\_blank">百度</a>

<a href="http://www.qq.com" target="\_blank">QQ</a>

<a href="http://www.sina.com.cn" target="\_blank">新浪</a>

**target="\_blank" 新窗口打开页面**

**target="\_self" 在当前窗口打开页面**

**<base target="\_blank"/>**

单标签 写在head里 不写在stlye（stlye属于css）里只有target和href两种属性

base 设置 页面上 所有的链接 默认的打开方式 和 链接地址 正常默认是当前页面打开不会新开页面

哪个加的下划线，下划线就跟着那个标签的字体颜色走

**a 标签 不能包含a标签**

<span></span> 区分样式

**内边距 padding 外边距 magin**

auto自动

padding 内填充(内边距、复合样式) 元素的边框和元素内容之间的距离

padding-top （right/bottom/left）

padding: 10px(上下) 20px(左右);

padding: 10px(上) 20px(左右) 30px(下);

padding: 10px(上) 20px(右) 30px(下) 40px(左);

可视宽 元素可以看见的区域的宽度

**可视宽 = width + padding + border;**

**padding 也会显示元素的背景**

margin(集合样式) 外边距 在元素的边框之外产生一个距离

margin-top(right/bottom/left)

margin: 10px(上下) 20px(左右);

margin: 10px(上) 20px(左右) 30px(下);

margin: 10px(上) 20px(右) 30px(下) 40px(左);

**margin 传递：**

**元素的上下margin ，会传递给父级**

**border阻住magin的传递**

padding和margin的区别

padding 产生在元素的边框以里，会显示元素的背景

margin 产生在元素的边框以外，不会显示元素的背景

父子级之间的距离，通过给父级加padding来设置

同级元素之间的距离，用margin来控制

**margin 叠加**

**两个相邻元素的上下margin，会叠加在一起，两个元素之间距离,以大的为准**

auto 自动

margin-left: auto; 元素居右显示

margin-right: auto; 元素居左显示

元素居中显示

margin-left: auto;

margin-right: auto;

**shift+6 省略号……**

vertical-align: top;垂直对齐方式（a标签会有一个默认的高度的增加）

**盒模型： 以我们现实生活中的纸箱为例，来理解一下我们元素的几个组成部分, 首先一堆盒子放在一起 之间会有距离(这个就是margin)，每个盒子都会有自己的外包装(border),盒子中会有一些填充的泡沫(padding),在里边才是我们装东西的地方(content:width/height);**

**选择器**

1. **Id选择器** #idName{ 需要添加的样式} 同一个页面id不能重名

问题：如果页面中多块元素的样式一致，用id的话就得每个都的去写一遍样式

2**类型选择器** ( 通过元素的标签名字，找到元素添加样式)

tagName { 需要添加的样式 } 例如：<div> <span>

问题: 每次都是给页面上，所有的某种类型标签添加样式，会影响到页面其他地方的样式

3 **class（.）选择器**(类选择器:通过元素的class属性，找到元素添加样式) .className { 需要添加的样式 }

页面上的class可以重名

同一个元素class 可以拥有多个名字，每个名字之间以空格隔开,每个名字添加的样式 元素都会识别

4 **通配符 \*** 通配符代表所有的标签 不建议使用

5 **包含选择器** 父级 (空格)要加样式的元素 { 样式}

6 **群组选择器** **,** span, a {}

找到所有span和a标签 给他们加上同样的样式 中间用逗号隔开

id选择符（#）

class选择符（.） 类选择器

类型选择符(div……)

通配符（\*）

群组选择符（，）

包含选择符（div p）

**选择器 优先级一致的情况下 ，后边的样式 覆盖前边的**

**选择器 优先级不一致的情况下，选择优先级高的**

两个包含选择器（中间是空格的）之间的优先级 加减法 为0的后面覆盖前面的

群组优先级 后面覆盖前面

选择优先级：

**JS> style(行间样式) > id > class > 类型选择 > \***

包含选择器的优先级是一种累加的关系

**标签**

<br/>折行符

section 版块

header 页面头部或者版块头部

footer 页面底部或者版块底部

nav 导航

h1-h6 标题

p 段落

strong 强调(显示加粗)

em 强调(显示倾斜)

mark 标记

time 时间

ul 无序列表

ol 有序列表

li ul或ol的列表项

dl 定义列表

dt 定义列表的列表项

dd dt的子项目

**标签语义化** 1. 方便后期添加样式

2. 提高代码可读性 -- 搜索引擎

3. 内容的可读性

**块和内嵌**

**块：1、默认独占一行**

**2、没有宽度时，默认撑满整行**

**3、支持所有css命令**

**内嵌(内联，行内):**

**1. 可以在一行显示**

**2. 不支持宽高，对上下的margin和padding等样式支持的都有问题**

**3. 代码换行会被解析成一个空格(通过控制 字体 和 字体大小来控制空格大小)**

**4. 宽度由内容撑开**

**display：block 显示为块**

**使内联元素具备块属性标签的特性**

**display：inline 显示为内嵌**

**使行块属性标签具备内联元素的特性**

**哪些标签是块：** div、h1-h6、header、ul、ol、li、nav、p、dl、dt、dd、footer、section

**哪些标签是内嵌：**time、strong、em、a、mark、span

**特殊的img标签 -- inline-block**

**只设置宽度或者只设置高度，img会等比的缩放**

**img 图片**

**src 图片路径**

**标签嵌套规范:**

**1. a标签里边不能在包含a标签**

**2. ul和li，以及ol和li，以及 dl和dt，dd他们是固定嵌套**

**1) ul和ol下边 只能嵌套li，li的父级也必须是ol或者ul**

**2) li中可以所以嵌套任何东西，甚至可以再放一个ul或者ol**

**3) dl 和 dt，dd 参考 ul和li**

**3. 内嵌标签不可以嵌套块标签**

**4. h1-h6,p,dt 这些带有标题性质的标签中同样不能嵌套块标签**

**划分页面结构，从上到下去分行，然后从左到右去分列,每一行，每一列都应该有个标签包起来**

display: none; /\* 不显示元素 \*/

display: block; /\* 显示为块 \*/

display: inline; /\*显示为内嵌 \*/

display: inline-block;(行内块)IE

**inline-block:**

**支持宽高**

**可以在一行显示**

**代码换行会被解析成一个空格**

**不设置宽度的时候，宽度由内容撑开**

**背景覆盖**：.box:hover {background-color: #0FF;

**Float浮动**

float 浮动(left/right/none)

1. 在一行显示2. 支持宽高（所有样式）3. 不设置宽度 宽度由内容撑开

4. 按照一个指定的方向去移动，碰到父级的边界或者前边的浮动元素停止

5. 脱离文档流 (半脱离文档流,提升层级半层)

6浮动之后上下margin不再叠加

**浮动： 使元素脱离文档流，然后 按照我们指定的方向去移动,碰到父级的边界或者前边的浮动元素停止**

文档流是文档中可显示对象在排列时所占用的位置。（页面中的元素根据它不同的特性会在我们的页面中去占有 不同区域的大小 -- 文档流）

脱离文档流 -- 指的是 元素不在页面中占位置

写结构的时候，先上到下分行，从左到右分列,每一行每一列都单独的包起来

**1要排列在一行的元素 都加浮动**

**2元素浮动之后，撑不开父级的高度，所以，父级如果可以设置固定的高度，就给父级设置固定高度，如果父级不能设置固定高度，就清浮动**

**3.元素浮动之后，宽度如果不需要内容撑开，一定记得给元素加宽度**

**（不加高度下边的元素会跑到浮动元素的下面使结构发生偏差出现重叠现象原因是半浮动有关）**

**Clear清浮动：**

1. clear 元素的某个方向不能有浮动元素(left、right、both两边、none)

clear:both; 在左右两侧均不允许浮动元素。

**clear:both必须加给块元素**

2. 使浮动元素还可以正常的撑开父级的高度

方法：

1.加高 问题：扩展性不好

2.父级浮动 问题：页面中所有元素都加浮动，margin左右自动失效

3.inline-block 清浮动方法： 问题：margin左右自动失效；

4.空（块）标签清浮动

问题：IE6 最小高度 19px；（解决后IE6下还有2px偏差）

5.br清浮动

问题：不符合工作中：结构、样式、行为，三者分离的要求。

6.**:after伪类** 清浮动方法 在元素内容的末尾 添加内容,并且给添加进去的内容设置样式,要添加的内容写在content里（现在主流方法）  **固定样式 .clearFix:after { content:"";**

**display:block;**

**clear:both;}**

**.clearFix{zoom:1;}**

* + - :after伪类： 元素内部末尾添加内容；

:after{content"添加的内容";} IE6，7下不兼容

* + - zoom 缩放

a、触发 IE下 haslayout，使元素根据自身内容计算宽高。

b、FF 不支持；

**方法实例**

1清浮动 在浮动元素的最下边加一个div，设置 .div {clear: both;}不常用了

2清浮动 在浮动元素的最下边加一个<br clear="all" />写在样式里

实例<div class="wrap">

<div class="left"></div>

<div class="right"></div>

<br clear="all" />

</div>

3.:after伪类 清浮动方法 在元素内容的末尾 添加内容,并且给添加进去的内容设置样式,要添加的内容写在content里（现在主流方法）

固定样式

**.clearFix:after {**

**content:"";**

**display:block;**

**clear:both;}**

**.clear {zoom:1;}**

4 触发BFC 清浮动

**BFC(block formatting context) 块级格式化上下文**

BFC css 2.1的一个概念,为元素提供了一个独立的布局环境，环境中的内容不会影响到环境外的布局,环境外布局 也不会影响到环境中内容

BFC就像给元素的周边加了一堵围墙，围墙里的东西出不去，外墙外的东西也挤不进来

BFC作用：

1. 包含浮动元素

2. 阻止子元素的margin向外传递

3. 不被浮动元素所覆盖

BFC触发条件:(符合以下任一条件即可)：

1) float的值不为none；

2) overflow的值不为visible（默认值“可见”）；

3) display的值为 table-ceil、table-caption和inline-block之一；

4) position的值不为static（默认值“无定位”）或relative（相对定位）中的任何一个；

**overflow 溢出**

**visible 可见(默认值)**

**hidden 隐藏**

**auto (溢出之后 显示滚动条，没有溢出则不显示)**

**scroll (显示滚动条)**

**以图换字**

1. 通过一个比较大的缩进的负值隐藏文字 （不推荐）

2. 给文字单加标签,display none隐藏 （不推荐）

**3. 高度为0，文字overflow:hidden掉，然后在加padding**

**通过padding区域来显示背景(用的最多的)**

**4. 设置超大行高 ，把文字移出当前元素区域，overflow hidden 掉**

width: 90px;

border: 1px solid #000;

background: url(img/title.png) no-repeat;

height: 0px;

padding-top: 20px;

overflow: hidden;

**自适应背景**

**元素浮动之后，宽度由内容撑开**

{font: 14px/30px "宋体"; **padding-right: 18px;**

border: 1px solid #000; float: left;

**background: url(img/icoN.png) no-repeat right center ;** }

**position 定位**

**position: relative（相对定位）**; (定位元素位置控制left/right/top/bottom)

**1 只加相对定位，不设置元素移动，元素和之前不会有任何的变化**

**2. 不脱离文档流 (元素移走之后，原始位置会被保留下来)**

**3. 提升层级**

**4. 根据自己的原始位置计算left,top**

**position: absolute; 绝对定位**

**1. 脱离文档流**

**2. 提升层级**

**3. 根据自己有定位的父级，来计算坐标,如果所有的父级都没有定位,就根据document（文档）来计算自己的坐标**

**4. 使内嵌元素支持所有的样式**

**5. 不设置宽度的时候，宽度由内容撑开**

**6. 触发BFC**

**元素定位之后，默认后边元素层级高于前边的元素**

**position: fixed 固定定位:**

**1. 脱离文档流**

**2. 提升层级**

**3. 固定在可视区的某个位置上**

**4. 使内嵌支持所有的样式**

**5. 不设置宽度的时候，宽度由内容撑开**

**6. 触发BFC**

**定位的其他值**

**静态定位:** position: static;(无定位，默认值)

position: inherit; 继承（不常用兼容有问题）

**z-index 层级(整数)**

**1.数值越大层级越高**

**2.层级最好在同级元素之间做比较**

**resize 设置元素大小可以进行改变**

1. **both（横向纵向都可以）/horizontal（横向）/vertical（纵向）/none**

2 必须配合overflow:auto 一块用

**定位在中间的话可以给以50%数值 再给一个margi（元素宽高一半注意边框也要算）的负值使元素位于正中间**

**opacity 透明度(0 - 1 小数)**

1. 元素设置了透明度之后，里边所有的东西都会变成透明的

（在下面再给一标签绝对定位上去覆盖透明层使之不透明显示字体）

**background-attachment：(fixed/scroll)**

背景图像是否固定或者随着页面的其余部分滚动

fixed 背景图固定 元素的背景图 不会随着滚动条而移动

background 设置了 fixed之后，background-position 就根据可视区来进行计算

**table表格**

table 表格 caption --表格标题（不建议使用）

thead 表格头部 tbody 表格内容

**tr 行 th 单元格 (加粗，居中) td 单元格**

属性: cellspacing 单元格间距 cellpadding 单元格填充

**给单元格设置高度，一行高度都会改变**

**给单元格设置宽度，一列宽度都会改变**

行间设置文本页面居中 align="center"

**border-spacing 单元格间距**

**border-collapse: collapse; 表格的单元格边框合并**

**td,th 没有margin值**

**thead，tbody，tr 没有margin和padding值**

**单元格中的内容默认垂直居中的**

**vertical-align 在单元格中 操作的是 单元格的内容的垂直对齐方式**

**colspan="7"横向合并单元格**

**rowspan="2"纵向合并单元格**

**默认样式清零**

**table {border-collapse: collapse;}**

**td,th {padding: 0;**

form**表单**

form 表单 <form action="http://www.baidu.com">

**action 提交跳转地址**

**input** 表单控件 **<input type="text" name="" />单标签**

**value 值** <input type="text" value="请输入搜索内容" />

**type** = "**text**"  **文本输入框**

type = **"submit" 提交按钮**

type=**"button" 按钮（空按钮）**

type=**"botton" 按钮（空按钮）**

type=**"reset" 重置（大多数不用）**

type="**hidden" 隐藏（不显示的）**

type="**image" src="1.jpg" 图片，src写图片地址**

type = "**password" 密码框**

type=”**file” 上传附件**

type="**radio" 单选**

type="**checkbox" 复选**

**label** 辅助input 点击label 和点击包含在label中的input效果一致**双标签**

**select 下拉菜单** 用在年月日 <select name=" "> **双标签**

**option 下拉菜单的选项** **<option>1999</option> 双标签**

**textarea 文本域**   **双标签**

**常用属性：**

**disabled 禁用掉表单的控件的可选(输入)功能,并且该项不会被提交**

**readonly 设置控件变成只读状态，不禁止控件的提交功能**

**selected 设置select的默认选中项**

**checked 设置radio或者checkbox的默认选中项**

**for for属性中写的是label要辅助的input的id**

**默认属性重置 outline 轮廓线**

**form { margin: 0; } IE6下 form上有一个margin值**

**input { outline: none;**

**margin: 0;**

**padding: 0}**

**/\* border-width 在各浏览器下不一致，需要根据设计图单独设置一下\*/**

**textarea { margin: 0;**

**padding: 0;**

**resize: none;**

**overflow: auto;**

**outline: none; 轮廓线无}**

**Html5新增表单控件**

**input** 表单控件 **<input type="text" name="" />单标签**

**type = "email" 邮箱**

**type = "url" 网址**

**type= "tel" 电话(移动端会自动显示数字键盘)**

**type = "search" 搜索**

**type = "number" 数字**

**属性值 max 最大值**

**min 最小值**

**step 间隔步数**

**type="range" 范围 横向滑动条 类似音量 与快进显示条**

**属性值 max 最大值**

**min 最小值**

**step 间隔步数**

**type = "color 颜色取值表**

**时间控件**

**datetime 国际标准时间 但是 浏览器没有实现 用 datetime-local 代替**

**type="datetime-local" 完整的时间日期**

**type="month" 月份**

**type="week" 周**

**type="date" 年月日**

**type="time" 时间**

**datalist 定义输入框可能出现的值 通过list属性和输入框关联**

**双标签 <datalist></datalist>**

**<input type="search" list="valList" />**

**<datalist id="valList">**

**<option>html</option>**

**<option>html+css</option>**

**<option>html5</option>**

**</datalist>**

**List 属性 写下面 datalist的Id**

**新增属性：**

**placeholder="" 提示信息**

**autocomplete(no/off) 是否保存输入内容**

**no 默认的保存 off 不保存**

**autofocus 自动聚焦 无选项值**

**required (设置该项 必须点写) 无选项值**

**pattern="\d{1,5}" /> 正切 设置数字限制 js会讲**

**formaction 写在提交按钮中的提交跳转地址**

**formnovalidate 取消验证**

<input type="submit"value="保存至草稿箱" formaction="http://..." formnovalidate/>

**Border-radius: “半径”（背景圆角）**

**兼容性**

**1标准盒模型:**

**内容宽(高) = width(height)**

**可视宽(高) = width(height)+padding+border;**

**(文档流宽高)占位宽(高) = width(height) + padding + border + margin; 怪异盒模型：**

**内容宽(高) = width(height) - padding - border;**

**可视宽(高) = width(height);**

**在IE下，不设置文档声明 盒模型解析会陷入怪异模式**

**解决方案：设置<!doctype html>**

**2**  **IE6下最小高度问题**

**在IE6下, 具有块属性元素，高度小于19px，会被当做19px来处理**

**解决方案：**

**1. overflow:hidden;**

**2. font-size: 0; -- 不推荐**

**3 IE6下，不支持1px的点线**

**解决办法：切背景平铺**

**4 在IE6下，父级有border的情况下，可能会造成子元素的margin失效**

**解决办法: 触发(父级的)haslayout（类似BFC）**

**haslayout 元素根据自己父级的宽度，或者自身内容的大小，**

**来重新计算自己的width和height**

**触发haslayout:**

**zoom: (除 normal 外任意值)(常用)**

**display: inline-block zoom 缩放**

**height: (任何值除了auto) 1. 元素的多少倍**

**float: (left 或 right) 2. zoom 不是标准样式**

**width: (任何值除了auto**

**position: absolute**

**haslayout: 是IE渲染引擎的一个组成部分**

**5 在IE6下，元素的内容可以撑开元素设置好的宽高**

**解决办法:正确计算元素宽高（不知道设置overflow可以不可以）**

**6 html5 新增标签，在IE8及IE8之前的IE浏览器不识别**

**解决办法: 1. 调用<script src="...地址"></script>**

**2 标签少的话可以直接写（js）**

**<script>**

**document.createElement("header");**

**document.createElement("nav");**

**</script>**

**7 (伪类)在IE6下，只支持给a标签加伪类，并且只支持加link、visited、hover、active**

**8 IE6,7下 不支持把其他标签的display设置成table相关类型**

**9 在IE6，7下，给块标签加display:inline-block无效**

**10 在IE6下，元素中除了文字之外，还有inline-block类型元素，元素的行高失效**

**解决办法： 分别包起来，分别设置**

**11 IE6下双边距BUG**

**1 在IE6下，块元素有浮动和横向margin，横向的margin值会放大成两倍**

**2 margin-left 左侧第一个元素有双边距 margin-right 右侧第一个元素有双边距**

**解决办法：**

**给元素加display: inline;**

**12 下margin失效问题**

**在IE6下，每行元素的宽度和父级宽度相差不能超过2px，否则最后一行下margin失效(IE7下直接失效);**

**解决办法:用padding代替下margin**

**13 文字溢出BUG**

**在IE6下，两个浮动元素中间有注释或者内嵌元素，父级的宽度和子元素宽度相差小于3px，元素内的文字会被复制**

**解决办法： 1.给注释或者内嵌元素用div包起来**

**2. 给父级的宽度加宽3px**

**14 Li下的间隙问题**

**IE6,7下，li本身没有浮动，但是li的内容浮动了，li下边就会多出来几px的间隙；**

**解决办法:**

**1. 给li加vertical-align: top;(常用)**

**2.给li加浮动(每一级都得去清浮动设置宽度 麻烦)**

**15 浮动和绝对定位并列的问题**

**IE6下，浮动元素和绝对定位元素是并列关系的话，绝对定位(元素)会消失掉**

**解决办法: 给绝对定位元素单独包起来(使之不再是并列关系)**

**16 相对定位的问题 在IE6，7下，元素有相对定位的话，父级的overflow对它不起作用**

**解决办法: 给元素的父级 也加相对定位**

**17 绝对定位问题 在IE6下，定位父级的宽高是奇数的话，元素的right值或者bottom值会有1px的偏差**

**解决办法：避免奇数**

**18 在IE6下，通过margin负值，使元素移出父级的位置，出去的部分会被截掉**

**解决办法：给元素加相对定位**

**19 IE6不支持固定定位**

**解决办法: 1. 用js处理**

**<script> window.onscroll = function(){**

**var box = document.getElementById("box");**

**box.style.width = "200px";**

**//box.innerHTML ="哈哈";}; </script>**

**2. css行为(强烈不建议使用)**

**20 IE6,7,8不支持opacity(透明度)**

**解决办法: filter:alpha(opacity = ) (0-100的整数)**

**opacity: .5;与filter:alpha(opacity=50);都写上**

**21 输入类型表单控件系列问题**

**textarea 文本域text文本输入password密码**

**在IE6，7下输入类型的表单控件，上下两边各有1px的间隙**

**解决办法: 给控件加浮动**

**在IE6，7下输入类型的表单控件，输入文字的时候，背景图会跟着文字一起移动**

**解决办法: 背景图加给父级(控件可能需要背景重置)**

**在IE6下输入类型的表单控件,border:none无效**

**解决办法： 1. border:0; 2. 重置元素的背景**

**22 在IE6下，label必须要加for属性(否则无效)**

**<label for="check">**

**<input type="checkbox" id="check" />复选**

**</label>**

**23 IE6不支持PNG图片的透明**

**解决办法：调用js文件（写在style外 并且条件注释掉）**

**<!--[if IE 6]>**

**<script src="DD\_belatedPNG\_0.0.8a.js"></script>(调用的文件)**

**<script>（src里写调用的文件地址及名字）**

**DD\_belatedPNG.fix('.pngImg(写图片的class名)');**

**</script>**

**<![endif]-->**

**24 \*,+ IE7及IE7之前的IE浏览器识别 (hack)**

**\_ IE6及IE6之前的浏览器识别（写在样式前在IE里读取）**

**25 !important 提升样式的优先级(写在样式的后面 ;号里边)**

**实例： background: blue !important;**

**在IE6下，在!important 后边在加一条同样的样式,会破坏掉!important的作用**

**IE6的在线审查元素**

**直接写一个js代码用完记得删掉**

**<script type="text/javascript" src="https://getfirebug.com/firebug-lite.js"></script>**

**条件注释语句**

**条件注释语句，IE9 及IE9之前的IE支持**

**<!--[if IE]>**

**这是IE**

**<![endif]-->**

**<!--[if IE 9]>**

**9**

**<![endif]-->**

**<!--[if IE 8]>**

**8**

**<![endif]-->**

**<!--[if IE 7]>**

**7**

**<![endif]-->**

**<!--[if IE 6]>**

**6**

**<![endif]-->**

**<!--[if gt IE 7]>**

**gt 高于 某个版本**

**<![endif]-->**

**<!--[if gte IE 7]>**

**gte 高于等于 某个版本**

**<![endif]-->**

**<!--[if l(L)t IE 7]>**

**lt 低于 某个版本**

**<![endif]-->**

**<!--[if lte IE 7]>**

**lte 低于等于 某个版本**

**<![endif]-->**

**图片格式**

**1 png24 -- 支持透明度，支持透明，图片质量高 图片的大小比较大**

**2 gif -- 不支持透明度，但支持透明，图片质量不高，而且大图容易失真有锯齿 -- 图片大小中等**

**3 jpg -- 不支持透明度，不支持透明，图片质量高 图片大小比较小**

**BFC和haslayout比较**

**BFC 作用：(布局环境)**

**1.防止元素被浮动元素覆盖**

**2.阻止margin传递**

**3.包含浮动元素**

**触发BFC：**

**1. display (inline-block，table-ceil，table-caption)**

**2. position （absolute，fixed）**

**3. float (left/right)**

**4. overflow (hidden, auto, scroll)**

**haslayout: (IE渲染引擎的一个组成部分)**

**触发之后，可以使元素根据内容大小来重新计算自身宽高**

**作用：**

**1.阻止margin传递**

**2.清浮动**

**haslayout 的触发条件:**

**1. display:inline-block;**

**2. width 不为auto的值**

**3. height不为auto的值**

**4. float不为none**

**5. position:absolute**

**6. zoom 不为normal的其他值**

**完整版清浮动**

**.clearfix { zoom: 1;}(IE6下触发haslayout清浮动)**

**.clearfix:after { content: "";**

**clear: both;**

**display: block;**

**浮动常犯错误**

**1 元素浮动之后，如果宽度需要内容撑开，就给元素中包含的块元素都加浮动(或者设置宽度), 否则就给元素设置宽度（**不然IE6下块元素撑满整行能把父级也撑满整行导致窜行**）**

1. **在IE6下 ，子元素会撑开父级设置好的宽高 (所以测量要仔细不然会错行)**

**3 在IE6下元素浮动排在一行，记得一行元素都得加浮动(会导致两元素之间出现间隙)**

**4 同行之间的元素一个浮动剩下的也要浮动**

**5 浮动要加在同一级元素 否则IE下会错位显示**

**Text-indenr(缩进) 尽量不要给内嵌标签里加 加给块标签**

**Css sprite(精灵)**

**1. 页面加载速度**

**2. 服务器压力**

**3. 流量问题**

**4. 代码的通用性**

**5. 代码的维护性**

**Border-radius 圆角**

**border-radius: x半径/y半径;**

**X 半径百分比 根据元素宽度计算 (包括padding 和border)**

**y 半径百分比 根据元素高度计算**

**1 border-radius: 20px(左上角,右下角) 40px(右上角，左下角);**

**2 border-radius: 20px(左上角) 40px(右上角，左下角) 60px(右下角);**

**3 border-radius: 20px(左上角) 40px(右上角) 60px(右下角) 80px (左下角);**

**4 border-top-left-radius: 30px;(设置一个角)**

**border-radius: 20px 40px 60px 80px/5px 20px 30px 40px;**

**text-shadow文字阴影**

**text-shadow 文字阴影**

**text-shadow: -10px -10px 5px #C86;**

**属性值： x 偏移 y 偏移 模糊半径 颜色**

**同一个元素可以拥有多层阴影，每层阴影之间用","隔开 多层阴影时，先写的在上边，后写的在下边**

**sin 直角三角形中 对边/斜边**

**cos 直角三角形中 临边/斜边**

**tan 直角三角形中 对边/临边**

**角度转弧度 deg\*π/180**

**rgba=rgb+alpha（0-1）**

**background: rgba(0,0,0,.5)**

**box-shadow 盒模型阴影**

**box-shadow**

**box-shadow: inset -20px -20px 20px -20px #000, inset 20px 0 20px -20px #000;**

**参数：inset (内阴影可选参数) x 偏移 y 偏移 模糊半径 扩展半径(可选参数) 颜色**

**同一个元素可以拥有多层阴影，每层阴影之间用","隔开。多层阴影时，先写的在上边，后写的在下边**

**Index主页面命名**

**Text-overflow**

**ICO引入**

**<link href="favicon.ico" rel="icon" />**

**溢出显示省略号**

**white-space: nowrap;**

**overflow: hidden;**

**text-overflow: ellipsis;**

**(css3)text-overflow 文字溢出之后是否显示省略号(ellipsis显示省略号clip不显示省略号);都兼容**

**渐变样式**

**transition: .5s;**

**css3 过渡 在元素的样式改变的时候，添加一个过渡动画**

**等高布局**

**css 2.1 (不兼容IE6)**

**min-width 最小宽度**

**max-width 最大宽度**

**min-height 最小高度**

**max-height 最大高度**

**圣杯布局和双飞翼布局解决的问题是一样的，就是两边定宽，中间自适应的三栏布局，中间栏要在放在文档流前面以优先渲染。**

**1 布局先写中间的再写左右的 给中间一个左右的margin值 ，在给左边的一个浮动之后用margin-100% 使他排列到最左测，在给右边的一个浮动之后用margin负的宽度大小的值 使他排列到最又侧，都得写浮动记得**

**2或者用绝对定位 定到两边，但是中间的也要给左右margin值**

**给父级加overflow:hidden, 再给需要等高的两个元素 都加一个大的下padding，再加一个下的margin负值使之不占位，父级的溢出隐藏使之显示等高**

**未知宽高的图片水平垂直居中**

**结构<div class="box">**

**<img src="img.jpg" /><span></span>**

**</div>**

**给父级一个 text-align: center;给图片标签加一个同级的span标签并设置display: inline-block，高度100% vertical-align: middle;垂直居中图片也设置垂直居中**

**多行文本垂直居中**

**同上差不多，给文字用一个内嵌包起来， 再在后面跟一个并列的空内嵌，设置高度100% 以及垂直对齐方式居中，记得内嵌转成内联块**

**页面中添加视频**

**<!---**

**<video controls width="800" height="500" style="background: #000;" autoplay loop>**

**<source src="iceAge.mp4" />**

**</video>**

**--->**

**<iframe height="498" width="510" src="http://player.youku.com/embed/XMTY0ODcwOTg1Mg==" frameborder=0 allowfullscreen></iframe>**

**<!--**

**<audio controls style="display: none;" autoplay>**

**<source src="iceAge.mp4" />**

**</audio>**

**属性选择器（IE6不兼容 其它的IE兼容）**

[**http://caniuse.com/**](http://caniuse.com/) **兼容的看看**

**1 p[miaov] 找到具有miaov这个属性的p标签**

**2 p[miaov="p1"]找到具有miaov这个属性的p标签，并且属性值还得是p1**

**~ 元素的属性值是一个词列表**

**3 E[attr~="value"]指定属性名，并且具有属性值，此属性值是一个词列表，并且以空格隔开，其中词列表中包含了一个value词，而且等号前面的“〜”不能不写**

**4 E[attr^="value"]指定了属性名，并且有属性值，属性值是以value开头的**

**5 E[attr$="value"]指定了属性名，并且有属性值，属性值是以value结束的**

**6 E[attr\*="value"]指定了属性名，并且有属性值，属性值中包含value**

**7 E[attr|="value"]指定了属性名，并且属性值是value或者以“value-”开头的值（比如说zh-cn）**

**Form 表单相关伪类**

**1 :foucs 得到焦点的时候 （autofocus为设置自动聚焦）**

**2 不同浏览器支持的格式不一样，单独设置提示信息（placeholder）的字体样式 ：**

**一般都可以的 input::-webkit-input-placeholder {color: yellow;}**

**火狐的 input::-moz-placeholder {color: yellow;}**

**3 input:enabled 没有设置disabled(禁止掉)的表单控件**

**input:disabled 设置disabled的表单控件**

**4 input:checked 常用 （radio单选checkbox复选）**

**找到选中的单选或复选框**

**Css2.新增的结构选择器查找毗邻元素**

**样式 .h2+p{background: red;}**

**+找到.h2这个元素后边紧跟同级的p标签**

**样式 .h2~p{background: red;}**

**~找到.h2这个元素后边所有同级的p标签**

**> 只找到第一级元素**

**单选和复选的美化**

**Css新增的结构性伪类子元素选择器**

**Last最后一个**

**1 e:nth-child(n) 找到e元素父级下的第n个子元素，并且元素的类型还得是 e e:nth-of-type(n)**

**如果不需要限制元素的类型，可以使用“\*”**

**even 偶数 odd 奇数 写在 n 的位置 选择奇 偶**

**n 等于 从0到正无穷的整数**

**3n 3的倍数**

**3n-1**

**3n+1**

**事例：p:nth-child(3){background: red;}**

**body \*:nth-child(3){background: red;**

**body \*:nth-child(even){background: red;}**

**body \*:nth-child(odd){background: green;}**

**body \*:nth-child(3n+1){ background: blue;}**

**2 e:nth-last-child()相当于倒着数的nth-child，具体参数 参考 nth-child （Last最后一个）**

**3 :first-child 相当于 :nth-child(1) （first正数第一）**

**:last-child 相当于 :nth-last-child(1) （Last最后一个）**

**4 e:nth-of-type(n) 找到e元素父级下第n个，e类型的元素**

**:nth-of-type 可以接受的参数参考 nth-child**

**5 > 只找到第一级子元素**

**实例：body>p:nth-of-type(2) {background: red;**

**6 :last-of-type相当于倒着数的 :nth-of-type**

**:last-of-type 可以接受的参数参考 nth-child**

**7 e:first-of-type正数第一个**

**其它结构性伪类**

**8 e:first-letter 找到e元素第一个字符**

**e::first-line 找到第一行**

**::selection 设置选中后的样式 (只能改变字体和背景颜色){必须双冒号}**

**9 :not排除掉哪个**

**示例：div:not(.div2){background: red;}**

**10 p:before 在p元素前面添加内容**

:before 选择器在被选元素的内容前面插入内容

**p:after在p元素后面面添加内容**

请使用 content 属性来指定要插入的内容。

**content:写添加的内容的**

**示例： p:before {content:"妙味妙味";}**

**p:after {content:"妙味";}**

**（添加进去的伪元素在页面中不能选中）**

**11 :target (target目标的意思)**

:target选择器可用于当前活动的target元素的样式。

用到锚点，点击这个链接的a标签是 a标签所链接的元素发生变化

**示例：div:target {background: red;}**

**结构<a href="#div1">div1</a>**

**<a href="#div2">div2</a>**

**<div id="div1"></div>**

**<div id="div2"></div>**

**怪异盒模型**

**box-sizing**

**border-box 怪异盒模型**

**content-box 标准盒模型**

**移动端**

**如何测试移动端页面**

**真机测试 -- xampp，N多种手机**

**模拟器测试 -- emulation**

**移动端不设置视口大小 默认一般是980**

**<meta name="viewport" content="" /> 设置视口大小**

**width (number||device-width) 部分手机不支持 把width设置成 数值**

**initial-scale 初始缩放比例**

**minimum-scale 最小缩放比例**

**maximum-scale 最大缩放比例**

**user-scalable (yes||no) 是否允许用户进行缩放**

**像素比**

**window.devicePixelRatio**

**像素一个像素的内容 放大n个像素去显示**

**<script>**

**alert(window.devicePixelRatio); 自动显示 缩放倍数**

**</script>**

**rem:**

**root em 根据根节点（html）的字体大小计算的em值**

**/\*background-size: 10px 20px;\*/ 设置 ccs3背景图大小**

**/\*background-size: 100% 50%;\*/**

**/\*background-size: 100% auto; 宽度和元素宽度保持一致 高度 等比缩放 \*/**

**/\*background-size: auto 100%; 高度和元素宽度保持一致 宽度 等比缩放 \*/**

**/\*background-size: contain; 等比缩放图片 至元素可以包含的最大范围，保证图片显示完整不超出 \*/**

**/\* background-size: cover; 等比缩放图片 铺满整个元素，但是图片可能会超出\*/**

**}**

**/\***

**background-size: 背景图的宽度 背景图的高度;**

**background-size: 100%(根据元素的宽度) 50%(根据元素的高度);**

**移动端字体： Helvetica**

**移动端的默认样式：**

**a,**

**input,**

**button {**

**-webkit-appearance: none;**}

**视口设置:**

**1. <meta name="viewport" content="width=device-width,user-scalable=no" />**

**2. <meta name="viewport" content="width=device-width,initial-scale = 1,minimum-scale = 1, maximum-scale = 1, user-scalable = no" />**

**3. 像素比 window.devicePixelRatio**

**!function() {**

**var scale = 1;**

**var ratio = window.devicePixelRatio;**

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

**var meta = document.createElement("meta");**

**meta.setAttribute("name","viewport"); meta.setAttribute("content","width=device-width,initial-scale = "+scale+",minimum-scale = "+scale+", maximum-scale = "+scale+", user-scalable = no");**

**document.head.appendChild(meta);**

**}();**

**rem 适配:**

**var html = document.querySelector("html");//获取html**

**var width = html.getBoundingClientRect().width; //获取html的宽度**

**html.style.fontSize = width/15 + "px"; （设置html的宽度是屏幕宽度的16之1得出来的结果不能小于12）**

**默认样式:**

**a,input,buttom {**

**-webkit-tap-highlight-color: rgba(0,0,0,0);**

**}**

**input,**

**button {**

**-webkit-appearance: none;**

**}**

**设计图的取图问题:**

**取大图，然后在通过background-size控制大小**

**模拟固定定位 :**

**1 就是给想要固定定位的元素单独拿出来用**

**CSS3新增样式**

**设置怪异盒模型 box-sizing: border-box;**

**text-stroke 描边**

**text-stroke: 描边的宽度 描边的颜色;**

**注意：**

**1.text-stroke 必须添加浏览器前缀**

**2.目前只有-webkit-的浏览器实现了这条样式**

**浏览器前缀:**

**-webkit- (chrome，safari，opera)**

**-moz- (firefox)**

**-ms- (IE)**

**-o- (老版opera)**

**Direction 改变文字的排列方向**

**ltr (left to right) 左到右**

**rtl (right to left) 右到左**

**注意： 必须添加 unicode-bidi: bidi-override; 一起使用**

**矢量图(标)**

**写个css,之后引入到html,再用字体的时候直接写我们起的名字就可以**

**<link href="../???.css" rel="stylesheet" />**

**Css:**

**@font-face {**

**font-family: "miaov"; /\* 我们给这个字体起的名字\*/**

**src:url('1-webfont.eot?#iefix')format('embedded-opentype'); /\*字体文件的地址\*/**

**font-style: italic(normal); /\* 这个字体初始的时候，是否是倾斜的 \*/**

**font-weight: bold(normal); /\* 这个字体初始的时候，是否是加粗的 \*/}**

**h1 {font-family: "miaov"; /\* 在使用的时候，我们直接调用上边起的名字，就可以调用到这个字体 \*/}**

**转化字体相关的：**

**在AI中绘制出矢量图的形状**

**在font lab中添加进入字体**

**在font lab中导出字体**

**上传至https://www.web-font-generator.com/导出各种兼容版本**

**可用网站：**

**http://www.fontsquirrel.com/fontface/generator**

**字珠**

**https://www.web-font-generator.com/**

**css3新增样式 线性渐变**

**linear-gradient 线性渐变**

**linear-gradient(颜色值,颜色值,颜色值,颜色值...);**

**示例：**

**background-image: linear-gradient(red,blue,green,yellow);**

**(如果不要渐变的话两个颜色之间的渐变点 数值设置成相同的数值就行)**

**颜色位置，渐变点: 常用百分比设置**

**background-image: linear-gradient(90deg(渐变方向单位角度),red,blue,green,yellow);**

**渐变方向: deg 角**

**不加前缀版:**

**0deg 从下到上**

**旋转方向是顺时针**

**勾股定理:斜边\*斜边 = 直角边1\*直角边1+直角边2\*直角边2;**

**background-image: repeating-linear-gradient(45deg,red 0px,red 20px,blue 20px,blue 40px);**

**repeating 重复的话 加在前面repeating**

**@keyframes move {**

**0%{**

**background-position: 0 0;**

**}**

**100% {**

**background-position: 600px 0;**

**}**

**} 直线 无限**

**通过：animation: 5s move linear infinite;调用**

**径向渐变：（这有一个at加入）**

**加入at 圆心点位置和形状可颠倒顺序写入**

**radial-gradient**

**background: radial-gradient(at 30px 100px(圆心点位置),red 10%,blue 30%);**

**background: -webkit-radial-gradient(30px 100px(圆心点位置),100px(x半径) 50px(y半径),red 10%,blue 100%);**

**设置半径的写法 firefox不支持**

**前缀写法 ：设置半径的写法 firefox不支持,只支持关键字:最近端，最近角，最远端，最远角，包含或覆盖 closest-side, closest-corner, farthest-side, farthest-corner, contain or cover**

**多背景图写法：**

**元素拥有多张背景图时，每张背景图之间以","隔开;**

**元素拥有多张背景图时，背景颜色写在最后.**

**多张背景图的时候，先写在上边，后写的在下边**

**背景图的原点设置：**

**背景图的原点默认 在元素padding的左上角 background-origin (背景图原点位置)**

**border-box (边框左上角)**

**padding-box(padding左上角 默认值)**

**content-box (背景图原点设置) 内容区左上角开始**

**背景图裁切：**

**background-clip 背景裁切**

**border-box(默认值)**

**padding-box**

**content-box**

**-webkit-text 在文字中显示(必须加前缀，只有webkit支持)**

**制作WIFI:**

**写一个div 设置圆角边框 在写四个边框的颜色**

**border-color: #000 rgba(0,0,0,0) rgba(0,0,0,0);**

**border-radius: 50%;**

**css3动画transition**

**兼容至IE10及以上 复合样式**

**transition 过渡 (transition 作用：在元素数值类的样式改变的时候，添加一个过渡的动画)**

**transition-duration 过渡时间 (s||ms)**

**transition-property 要运动的样式(all||attr（特定的一个）||none)**

**transition-delay 延迟时间**

**transition: .5s width, 1s .5s height, 2s 1.5s background;**

**transition-timing-function 运动形式**

**ease：（先加速后减速）默认值**

**linear：（匀速）**

**ease-in：(加速)**

**ease-out：（减速）**

**ease-in-out：（先加速后减速）**

**cubic-bezier 贝塞尔曲线（ x1, y1, x2, y2 ）**

**http://cubic-bezier.com/**

**keyframes 动画帧**

**格式**

**@keyframes 名字 {**

**0%{**

**起始样式**

**}**

**100%{**

**结束样式**

**}**

**}**

**@keyframes 名字 {**

**form{**

**起始样式**

**}**

**to{**

**结束样式**

**}**

**}**

**注意兼容移动端必须加前缀**

**@-webkit-keyframes move {**

**}**

**\*/**

**写好的动画帧通过animation: move 2s;调用**

**兼容移动端记得加前缀**

**animation 动画**

**必要属性**

**animation-name 动画名称（关键帧名称）**

**animation-duration 动画持续时间**

**animation-timing-function 动画运动形式**

**linear匀速 ease 缓冲 ease-in 由慢到快。**

**ease-out 由快到慢 ease-in-out 由慢到快再到慢。**

**cubic-bezier(number, number, number, number)： 特定的贝塞尔曲线类型，4个数值需在[0, 1]区间内**

**animation-iteration-count 重复次数(nub||infinite)**

**animation-delay 动画延迟 (只是第一次有延迟)**

**想要每次都有延迟 可以在动画帧 里添加一个阶段的一样位置模拟出延迟效果**

**复合：animation: move 2s 2s infinite;**

**animation-direction 播放前重置**

**动画是否重置后再开始播放**

**alternate 动画直接从上一次停止的位置开始执行(偶数次倒挡来回播放)**

**normal 动画跳到0%的状态开始执行**

**animation-play-state 播放状态**

**(running 继续 || paused暂停)**

**样式#box:hover #div {animation-play-state: paused;}**

**不写百分之0 百分之0默认就是优先级最高的样式**

**不写百分之100 百分之100 默认就是优先级最高的样式**

**动画结束之后,默认会回到优先级最高的样式**

**animation-fill-mode 定义动画结束和开始时的位置**

**none**

**forwards 当动画完成后，样式停在关键帧百分之百的位置。**

**backwards 元素渲染完，样式就显示是 关键帧的百分之0的位置**

**both 向前和向后填充模式都被应用。**

**样式调用:**

**animation: 关键帧的名字 动画时间 延迟 次数 偶数次播放状态 动画形式;**

**animation-fill-mode 定义动画结束和开始时的位置**

**animation-play-state 暂停和继续**

**响应式布局 媒体查询：@media**

**也得写入<meta name="viewport" content="width=device-width,user-scalable=no" /> 因为也有移动端的使用**

**媒体类型 @media not print {**

**all 所有媒体 （常用）**

**braille 盲文触觉设备**

**embossed 盲文打印机**

**print 手持设备**

**projection 打印预览**

**screen 彩屏设备**

**speech '听觉'类似的媒体类型**

**tty 不适用像素的设备**

**tv 电视**

**关键字 and (和：添加并列条件的) not（排除）**

**媒体特征**

**Width min-width max-width**

**Height min-height max-height**

**device-width min-device-width max-device-width**

**device-height min-device-height max-device-height**

**(max-width:600px)**

**(max-device-width: 480px) 设备输出宽度**

**(orientation:portrait) 竖屏**

**(orientation:landscape) 横屏**

**(-webkit-min-device-pixel-ratio: 2) 像素比**

**示例：@media all and (min-width:500px) and (max-width:600px) {#box {background: green;}}**

**max-width:300px 宽度 小于等于 300px的时候 识别**

**min-width:800px 宽度大于等于 800px的时候 识别**

**媒询中的横竖检测可以用在pad上，但是建议不要在手机上使用**

**因为输入框显示出来的的时候 宽高比会产生变化**

**另外注意书写顺序 会有样式覆盖的问题**

**引入方法：**

**1 <link href="css/landscape.css" rel="stylesheet" media="all and (orientation:landscape)" />**

**<link href="css/portrait.css" rel="stylesheet" media="all and (orientation:portrait)"/>**

**多个link引入 也会出现覆盖的问题，所以在行间设置media支持的类型样式，来限制样式读取**

**2 @import 在样式表中再引入一个新的样式表**

**1. @import url(样式表的地址);**

**2, @import 必须写在样式表的头部**

**3，可以在后面设置媒体类型 和媒体特征限制 读取类型**

**示例：@import url(css/landscape.css) all and ( orientation:portrait);**

1. **写在style每条样式后边每条样式加 @media all and (max-width:300px) {}也要注意书写或排列顺序 注意覆盖问题**

**滤镜 filter(有的需要加浏览器前缀)**

**filter 滤镜**

**blur (px) 模糊**

**grayscale(0 - 1) 灰度**

**opacity( 0 - 1) 透明度**

**drop-shadow 盒模型阴影**

**sepia(0 - 1) 褐色**

**invert(0 - 1) 反色**

**hue-rotate (deg) 色相翻转**

**saturate( 0 - ) 饱和度**

**brightness (0 - ) 亮度**

**contrast(0 - ) 对比度**

**Mask遮罩（必须加前缀）**

**1.必须加前缀**

**2.目前只有webkit支持**

**mask-image**

**mask-repeat**

**mask-position**

**实例：-webkit-mask: url(img/mask.png) no-repeat center center;**

**border-colors**

**border-top-colors**

**border-right-colors**

**border-bottom-colors**

**border-left-colors**

**1.必须加前缀 2.目前只有moz实现了**

**实例：-moz-border-left-colors: red blue pink #000 yellow green;**

**border-image**

**border-image**

**border-image-source 背景图的路径**

**border-image-slice 背景图裁切**

**border-image-repeat 图像边框是否应平铺(repeat)、铺满(round)或拉伸(stretch默认值)。目前只有webkit支持**

**实例：-webkit-border-image: url(img/bt\_blue.png) 0 11 stretch stretch;**

**transform 变换**

**alert 弹出一个对话框**

**Math.sin(弧度); 正弦**

**Math.cos(弧度); 余弦**

**Math.tan(弧度); 正切**

**Math.PI π;**

**角度转弧度: Math.PI/180\*deg**

**示例：alert(Math.tan(Math.PI/180\*60)\*50);**

**角度转弧度公式 Math.PI/180\*deg**

**正N边形外角和 = 360**

**正N边形内角 = 180-360/n;**

**正弦 (sin) = 直角三角形中，对边/斜边**

**余弦 (cos) = 直角三角形中，临边/斜边**

**正切 (tan) = 直角三角形中，对边/临边**

**三角函数**

角度转弧度 π/180×角度  
弧度变角度 180/π×弧度

正n边形指具有n([正整数](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=70214187)n≥3)条相等边的[正多边形](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=421213)，其内角和为180(n-2)°，每个内角度数为180°（n-2）/n， [外角和](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=8271962)为[360°](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=3862940).

**transform-origin (变换原点设置)**

**关键字**

**数值 示例：transform-origin: 0 0;**

**transform中 后写的先执行**

**transform 变换**

**rotate(deg) 旋转**

**scale( 0 - ) 缩放**

**scaleX()**

**scaleY()**

**skew(x,y); 斜切**

**skewX(deg) x方向斜切**

**skewY(deg) y方向斜切**

**translate(x,y) 位移**

**translateX(px) x方向位移**

**translateY(px) y方向位移**

**移动端使用transform的时候，注意要加前缀**

**矩阵：**

**a,c,e**

**b,d,f**

**0,0,0**

**正常情况下这样写matrix(a,b,c,d,e,f);**

**x 缩放 a = x\*a, c=x\*c, e=x\*e**

**y 缩放 b = y\*b, d=y\*d, f=y\*f**

**x位移: e = e + x;**

**y位移: f = f + y;**

**通过矩阵实现倾斜**

**x轴倾斜: c=Math.tan(xDeg/180\*Math.PI)**

**y轴倾斜: b=Math.tan(yDeg/180\*Math.PI)**

**旋转:**

**a=Math.cos(deg/180\*Math.PI);**

**b=Math.sin(deg/180\*Math.PI);**

**c=-Math.sin(deg/180\*Math.PI);**

**d=Math.cos(deg/180\*Math.PI);**

**matrix 矩阵**

**IE下的这样写**

**filter: progid:DXImageTransform.Microsoft.Matrix( M11= 1, M12= 0, M21= 0 , M22=1,SizingMethod='auto expand');**

**IE下的矩阵没有E和F两个参数 M11==a; M12==c; M21==b; M22==d**

**transform:3D**

**3d下所有的样式 记得在移动端使用的时候，加前缀**

**perspective 景深 加景深 能让画面动作时看着有立体效果**

**主要用旋转 和位移两种，其余的两种几乎不用**

**transform: 旋转**

**rotate();**

**\*rotateX();**

**\*rotateY();**

**\*rotateZ();**

**translate(); 位移**

**translateX();**

**translateY();**

**\*translateZ();**

**Translate3d集合写法：transform: translate3d(100px,100px,-100px);**

**transform-style: preserve-3d; 使被转换的第一级子元素保留其 3D 转换：**

**transform-origin默认值transform-origin: center center 0;**

**3D盒子的几种写法：**

**1 定位+旋转**

**2 设置旋转中心 旋转**

**3 位移+旋转**

**perspective-origin 景深基点 视角的方向**

**backface-visibility: hidden; 隐藏背面**

**display: flex弹性盒模型**

**（主轴和侧轴的关系是 主轴从左到右 侧轴就是从上到下，如果主轴方向改变了，侧轴的方向可以旋转来确定，譬如 1主向右侧向下2主向上侧向右3主向左侧向上4主向下侧向左） 弹性和模型都是根据主轴的方向来设置的**

**flex-direction 设置主轴方向**

**row 从左向右排列(默认值)**

**row-reverse 从右向左排列**

**column 从上往下排列**

**column-reverse 从下往上排列**

**justify-content 主轴对齐**

**flex-start (默认) 元素在开始位置 富裕空间占据另一侧**

**flex-end 富裕空间在开始位置 元素占据另一侧**

**center 元素居中 富裕空间 平分左右两侧**

**space-between 富裕空间在元素之间平均分配**

**space-around 富裕空间在元素两侧平均分配**

**align-items 侧轴对齐**

**flex-start (默认) 元素在开始位置 富裕空间占据另一侧**

**flex-end 富裕空间在开始位置 元素占据另一侧**

**center 元素居中 富裕空间 平分左右两侧**

**弹性盒模型默认不会自己换行，而是压缩元素**

**flex-wrap 换行**

**nowrap (默认)**

**wrap 换行**

**wrap-reverse 反向换行**

**align-content 堆栈伸缩行**

**flex-start 元素在开始位置 富裕空间占据另一侧**

**flex-end 富裕空间在开始位置 元素占据另一侧**

**center 元素居中 富裕空间 平分左右两侧**

**space-between 富裕空间在元素之间平均分配**

**space-around(默认) 富裕空间在元素两侧平均分配**

**flex-flow**

**flex-flow 是 flex-direction 和 flex-wrap 的缩写**

**flex-flow: [flex-direction] [flex-wrap]**

**弹性盒模型中的margin: auto**

**margin: auto 富裕空间平均分布在元素的两侧**

**flex 子元素的弹性设置**

**auto 具有弹性**

**none 不具有弹性**

**number**

**元素的flex值计算 元素的尺寸 = 元素的flex值/所有元素的flex值之和 \* 父级宽度**

**order 排序 数值大在后边 数值小的在前边，默认为0**

**子元素align-self 侧轴对齐**

**flex-start (默认) 元素在开始位置 富裕空间占据另一侧**

**flex-end 富裕空间在开始位置 元素占据另一侧**

**center 元素居中 富裕空间 平分左右两侧**

**flex 兼容至 IE10 老版本 chrome下 请加前缀**

**移动端使用时 请用 display:box 做兼容;**

**display:box 移动端所有样式必须加前缀**

**Box-orient 定义盒模型的布局方向**

**Horizontal 水平显示**

**vertical 垂直方向**

**box-direction 元素排列顺序**

**Normal 正序**

**Reverse 反序**

**box-ordinal-group 设置元素的具体位置**

**Box-flex 定义盒子的弹性空间**

**子元素的尺寸=盒子的尺寸\*子元素的box-flex属性值 / 所有子元素的box-flex属性值的和**

**box-pack**

**start 所有子元素在盒子左侧显示，富裕空间在右侧**

**End 所有子元素在盒子右侧显示，富裕空间在左侧**

**Center 所有子元素居中**

**Justify 富余空间在子元素之间平均分布**

**box-align 在垂直方向上对元素的位置进行管理**

**Star 所有子元素在据顶**

**End 所有子元素在据底**

**Center 所有子元素居中**

**分栏布局兼容到ie9以上**

**必须加浏览器前缀**

**columns: 栏目宽度 栏目列数;**

**column-width 列的宽度。**

**column-count 列数。**

**Column-rule 间隔线**

**column-gap 间隔距离**

**less**

**在JS第二阶段回顾**[**http://less.bootcss.com/features/**](http://less.bootcss.com/features/)

Less 是一门 CSS 预处理语言，它扩展了 CSS 语言，增加了变量、Mixin、函数等特性，使 CSS 更易维护和扩展。

**.border(){} 加了括号的类 不会被编译出来**

**.w100() { width: 100px; } 加()后css就不会被读取了**

**// 单行注释掉 /\* \*/ 区域注释掉**

**/\* Less 声明变量 \*/**

**@y: yellow; 属性值**

**@bgImg: background-image; 属性名**

**@c: contentLeft; 名称**

**@url: "../img/"; 路径**

**@bg: @y url("@{url}bg.img") no-repeat; //在变量值中使用变量**

**@x: "y";**

**#div {**

**background: @y; /\* 在属性值中使用变量 \*/**

**content: @@x; /\* 在变量中使用变量\*/**

**}**

**#div2 {**

**@{bgImg}: @y; /\* 在属性名中使用变量 \*/**

**}**

**#@{c}Row1{**

**@{bgImg}: @y; /\* 在选择名中使用变量 \*/**

**}**

**#@{c}Row1{**

**@{bgImg}: url("@{url}banner.png"); /\* 在路径中使用变量，记得加引号\*/**

**}**

**#@{c}Row2{**

**@{bgImg}: url("@{url}icon.png"); /\* 在路径中使用变量，记得加引号\*/**

**}**

**#@{c}Row3{**

**background: @bg;**

**}**

**/\* 嵌套 \*/**

**.list {**

**margin: 0;**

**li {list-style: none;}**

**p {text-indent: 2em; }**

**a {**

**color: red;**

**&:hover { &代表返回父级**

**color: blue;**

**}**

**}**

**}**

**/\* 变量作用域 \*/**

**@w100: 100px;**

**.list {**

**width: @w100;**

**li {**

**@w100: 100%;**

**width: @w100;**

**p {**

**width: @w100;**

**}**

**} 里边的可以调到外边的变量**

**} 外边的不能调到里边的变量**

**/\* 混合 \*/**

**.border(@bW,@color) { //定义两个参数**

**border: @bW solid @color;**

**}**

**#div {**

**.border(2px,red);}**

**.border2(@bW:1px,@color:#000) { // 给参数添加默认值**

**border: @bW solid @color;}**

**.border2(@bW:1px,@color:#000) { // 给参数添加默认值**

**border: @bW solid @color;**

**}**

**#div3 { // 部分顺序颠倒着写**

**.border2(#ccc,2px);**

**}**

**#div4 { //加上变量名称顺序会自动变成设置变量的顺序**

**.border2(@color:#ccc,@bW:2px);}**

**应用实例:**

**.transform(@deg:0deg){**

**-webkit-transform: rotate(@deg);**

**-moz-transform: rotate(@deg);**

**-ms-transform: rotate(@deg);**

**-o-transform: rotate(@deg);**

**transform: rotate(@deg);**

**}**

**#div5 {**

**.transform(@deg:30deg)}**

**/\* less中的判断 \*/**

**.triangle (@dir:0,@color:#000,@w:10px){**

**width: 0;**

**height: 0;**

**border: @w dashed transparent;**

**.triangleIcon(@dir,@color,@w); 通过这个元素的改变来改变样式**

**}**

**.triangleIcon(@dir:0,@color:#000,@w:10px) when (@dir = 0){**

**border-bottom: @w solid @color; when 当时候 判断并列**

**}**

**.triangleIcon(@dir:0,@color:#000,@w:10px) when (@dir = 1) {**

**border-left: @w solid @color;**

**}**

**.triangleIcon(@dir:0,@color:#000,@w:10px) when (@dir = 2){**

**border-top: @w solid @color;**

**}**

**.triangleIcon(@dir:0,@color:#000,@w:10px) when (@dir = 3) {**

**border-right: @w solid @color;**

**}**

**#border {.triangle (@dir:3); }**

**/\* 循环 \*/**

**>, >=, =, =<, <**

**.loopW(@loop) when (@loop>0){ 这一行有一个判断大于0**

**width: 10px \* @loop;**

**.loopW(@loop - 1); 这一行有个循环的判断**

**}**

**心得：循环就是把大的条件 在在条件里写一遍 他就会去判断成立与否**

**循环里必须有一个判断过程**

**#div4 {**

**.loopW(5);**

**}**

**bootstrap前端框架**

**http://www.bootcss.com/**

**兼容至 IE9以上 基于H5 CSS3的基础上**

**JS写在body内容最下面引入**

**杂项**

**莫涛1870147 2084**

dropdown **下拉菜单 class名**

**Javascript:; 代替#号的链接**

**网上直接下载图片 记得修改成所有**

**<small>副标题写在h2标签里</small>**

**<title>提示语</title>**

**外联框架 -- 在页面中 在引入新的页面**

**iframe {display: block;} iframe 为内联块**

**<iframe src="http://www.baidu.com" width="100%" height="100%" scrolling="auto" frameborder="0"></iframe>**

**scrolling 滚动条 auto no yes**

**frameborder 边框**

**热区**

**实例：**

**<img src="img.jpg" usemap="#Map" />**

**<map name="Map">**

**<area shape="rect" coords="327,273,340,299" href="http://www.baidu.com">**

**<area shape="circle" coords="248,321,12" href="http://www.qq.com">**

**<area shape="poly" coords="311,156,303,151,307,148,305" href="#">**

**</map> 图片和map 之间通过name链接**

**usemap = "#name"**

**map 热区**

**area 点击区域**

**shape = "rect" 矩形点击区域coords="左上角X,左上角Y,右下角X,右下角Y"**

**shape="circle" 圆形点击区域 coords="圆心点X,圆心点Y,半径"**

**shape="poly" 多边形点击区域 coords="第一个点X,第一个点Y,第二个点X,第二个点Y....."**

**flash引入**

**实例：**

**<object>**

**<param name="wmode" value="transparent" />**

**<embed src="1.swf" width="400" height="400" wmode="transparent" ></embed>**

**</object>**

**wmode="transparent 设置背景透明**

**本身的背景不透明IE下会出现层级的问题 用<param name="wmode" value="transparent" /> 与wmode="transparent"关联 吧背景定义成透明了 解决层级问题**

**写介绍tnt 一只大榴莲**

**回顾总结 css3新增的选择器--属性选择器**

**e[attr](元素[属性名]) 找到具有某个属性的元素**

**e[attr="val"](元素[属性名="属性值"]) 找到具有某个属性并且属性等于我们要求的属性值的元素**

**e[attr|="val"]**

**e[attr~="val"]**

**e[attr$="val"]**

**e[attr^="val"]**

**e[attr\*="val"]**

**css3新增的选择器--表单新增的伪类**

**e:foucs**

**e:checked**

**e:enabled**

**e:disabled**

**e::-webkit-input-placeholder**

**css2.1新增的结构选择器**

**+**

**~**

**>**

**css3 新增的结构性伪类**

**e:nth-child(n)**

**e:nth-last-child(n)**

**e:first-child**

**e:last-child**

**e:nth-of-type(n)**

**e:nth-last-type(n)**

**e:first-of-type**

**e:last-of-type**

**e:first-letter**

**e:first-line**

**e::selection**

**css3 新增伪类**

**::after**

**::before**

**:target**

**:not**

**css3新增样式:**

**resize**

**text-overflow**

**border-radius**

**box-shadow**

**text-shadow**

**box-sizing**

**background-size**