

第一天

1. Css概念

CSS 指层叠样式表 (Cascading Style Sheets)(级联样式表)

Css是用来美化html标签的，相当于页面化妆。

◆样式表书写位置

```
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style type="text/css">
    样式表内容
  </style>
```

2. 选择器

2.1.写法

选择器是一个选择谁（标签）的过程。

选择器{属性:值; 属性:值;}		
属性	解释	
Width:20px;	宽	
Height:20px;	高	
Background-color:red;	背景颜色	
font-size:24px;	文字大小	
text-align:left center right	内容的水平对齐方式	

text-indent:2em;	首行缩进
Color:red;	文字颜色

3. 选择器

3.1.基础选择器

◆标签选择器

标签{属性： 值;}

特点： 标签选择器定义之后，会将页面所有的元素都执行这个标签样式。

```
div{
    font-size:50px;
    color: green;
    background-color:yellow;
    width:300px;
    height:200px;
}
p{
    color: pink;
    font-size: 60px;
}

</style>
</head>
<body>
    <div>14期威武</div>
    <div>14期霸气</div>
```

★颜色的显示方式

◎直接写颜色的名称

◎十六进制显示颜色

0-9 a-f

#000000; 前2为代表红色，中间2位代表绿色，后边2位代表蓝色。

◎rgb

```
color: rgb(120,120,120);
```

◎rgba

A代表alpha 不透明度 值 0-1

```
color: rgba(102,217,239,0.5);
```

◆类选择器（重点）

写法

.自定义类名{属性:值; 属性:值; }



特点：谁调用，谁生效。

一个标签可以调用多个类选择器。

多个标签可以调用同一个类选择器。

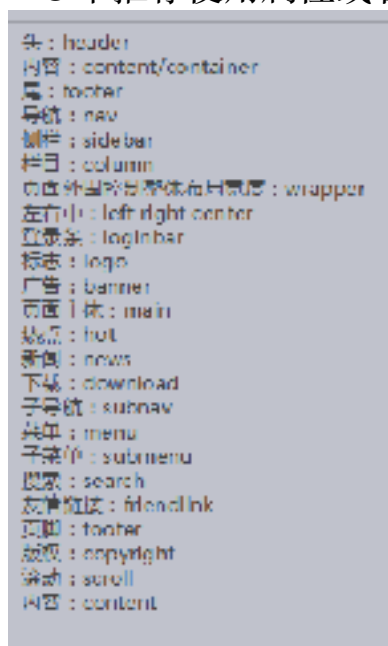
★类选择器命名规则

◎不能用纯数字或者数字开头来定义类名

◎不能使用特殊符号或者特殊符号开头（_）来定义类名

◎不建议使用汉字来定义类名

◎不推荐使用属性或者属性的值来定义类名



◆ID选择器

写法

#自定义名称{属性:值;}

特点：一个ID选择器在一个页面只能调用一次。如果使用2次或者2次以上，不符合w3c规范，JS调用会出问题。

一个标签只能调用一个ID选择器。

一个标签可以同时调用类选择器和ID选择器。

◆通配符选择器

*{属性:值;}

特点：给所有的标签都使用相同的样式。

★不推荐使用，增加浏览器和服务器的负担。

3.2.复合选择器

概念：两个或者两个以上的基础选择器通过不同的方式连接在一起。

◆交集选择器

标签+类（ID）选择器{属性：值；}

特点：即要满足使用了某个标签，还要满足使用了类（id）选择器。



◆后代选择器（重点）

选择器+空格+选择器{属性：值;}

后代选择器首选要满足包含（嵌套）关系。

父集元素在前边，子集元素在后边。

特点：无限制隔代。

只要能代表标签，标签、类选择器、ID选择器自由组合。

```
.box{
    font-size:40px;
    color:red;
}
div span{
    font-size: 50px;
}
.box span{
    background-color: blue;
}
.box .miss{
    color:red;
}
</style>
</head>
<body>
    <div class="box">
        <p><span class="miss">14期威武</span></p>

    </div>
    <div class="box"><span>14霸气</span></div>
```

◆子代选择器(不常用)

选择器>选择器{属性:值;}

选中直接下一代元素。

```
div>span{
    color:red;
    font-size:40px;
}
p>span{
    color:green;
    font-size:60px;
}
</style>
</head>
<body>
    <div>
        <p><span>14期威武</span></p>
        <span>14期霸气</span>
    </div>
</body>
```

◆并集选择器

选择器+, +选择器+, 选择器{属性:值;}

Font:italic 700 16px/40px 微软雅黑;

```
font: italic 700 16px/40px 微软雅黑;
```

4.3.文字的表达方式

◆直接写中文名称。

```
div{  
    font-family: 微软雅黑;  
    font-size: 60px;  
}
```

◆写字体的英文名称。

```
div{  
    font-family: microsoft yahei;  
    font-size: 60px;  
}
```

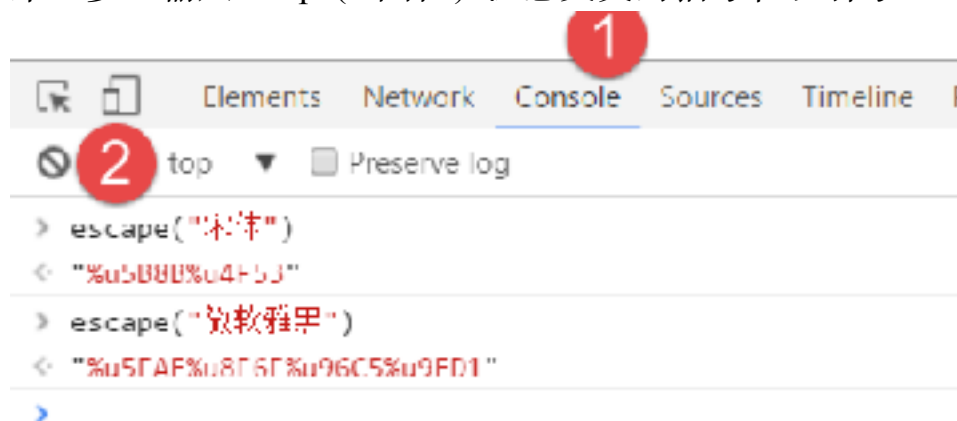
◆unicode 编码

字体名称	英文名称	Unicode 编码
宋体	SimSun	\5B8B\4F53
新宋体	NSimSun	\65B0\5B8B\4F53
黑体	SimHei	\9FD1\4F53
微软雅黑	microsoft yahei	\5FAE\8F6F\96C5\9ED1
楷体_GB2312	KaITI_GB2312	\6977\4F53_GB2312
隶书	LiSu	\96B6\4E66
幼园	YouYuan	\5E7C\5706
华文细黑	STXihei	\534E\6587\7EC6\9ED1
细明体	MingLiU	\7EC6\660E\4F53
新细明体	PMingLiU	\65B0\7EC6\660E\4F53

第一步: fl2

第二步: 找到console

第三步：输入escape(“宋体”) 注意英文的括号和双引号。



第二天

5. 样式表书写位置

◆内嵌式写法

```
<head>  
  <style type="text/css">  
    样式表写法  
  </style>  
</head>
```

◆外链式写法

写在head里，`<link rel="stylesheet" href="1.css">`

```
<!-- 外链式写法 -->  
<link rel="stylesheet" href="1.css">
```

◆行内样式表


```
<!-- 行内样式表 -->
```

```
<h1 style="font-size:30px; color:red;">14期霸气</h1>
```

◆三种写法特点:

★内嵌式写法，样式只作用于当前文件，没有真正实现结构表现分离。

★外链式写法，作用范围是当前站点，谁调用谁生效，范围广，真正实现结构表现分离。

★行内样式表，作用范围仅限于当前标签，范围小，结构表现混在一起。（不推荐使用）

6. 标签分类(显示方式)

6.1.块元素

典型代表,Div,h1-h6,p,ul,li

特点: ★独占一行

★可以设置宽高

★ 嵌套（包含）下，子块元素宽度（没有定义情况下）和父块元素宽度默认一致。

6.2.行内元素

典型代表 span ,a ,strong , em, del, ins

特点: ★在一行上显示

★不能直接设置宽高

★元素的宽和高就是内容撑开的宽高。

6.3.行内块元素(内联元素)

典型代表 input img

特点: ★在一行上显示

★可以设置宽高

6.4.块元素、行内元素

◆块元素转行内元素

display:inline;

```
/*块元素转换为行内元素*/  
div,p{  
    display: inline;  
}
```

◆行内元素转块元素

display:block;

```
span{  
    display: block;  
}
```

◆块和行内元素转行内块元素(应用多)

display:inline-block;

```
/*块和行内元素转行内块元素*/  
div,a,span,strong{  
    display: inline-block;  
    width:200px;  
    height: 200px;  
    background-color: yellow;  
    text-align:center;  
}
```

7. css三大特性

7.1.层叠性

当多个样式作用于同一个（同一类）标签时，样式发生了冲突，总是执行后边的代码(后边代码层叠前边的代码)。和标签调用选择器的顺序没有关系。

```
.box2{
  font-size: 200px;
  color: blue;
}
.box{
  font-size: 60px;
  color: red;
}
</style>
<head>
<body>
<div class="box box2">14点式</div>
<div class="box box2">14点式</div>
```

7.2. 继承性

继承性发生的前提是包含（嵌套关系）

- ★文字颜色可以继承
- ★文字大小可以继承
- ★字体可以继续
- ★字体粗细可以继承
- ★文字风格可以继承
- ★行高可以继承

总结：文字的所有属性都可以继承。

◆特殊情况：

h系列不能继承文字大小。

a标签不能继承文字颜色。

7.3.优先级

默认样式<标签选择器<类选择器<id选择器<行内样式

<!important

0 1 10 100 1000 1000以上

```
/*a*/
a {
  color: pink;
  font-size: 180px;
}

/*b*/
.b {
  color: green;
  font-size: 50px;
}

/*c*/
#c {
  color: red !important;
  font-size: 50px !important;
}

</style>
</head>
<body>
  <div class="box" id="von" style="font-size: 10px; color: yellow;">
    3.4 权重</div>
  </body>
</html>
```

- ◆优先级特点
- ★继承的权重为0
- ★权重会叠加

8. 链接伪类

a:link{属性:值;} a{属性:值}效果是一样的。

a:link{属性:值;}	链接默认状态	
a:visited{属性:值;}	链接访问之后的状态	
a:hover{属性:值;}	鼠标放到链接上显示的状态	a:active{属性:值;}
值;}	链接激活的状态	
: focus{属性:值; }	获取焦点	

```
 */  
 /*链接默认状态*/  
 a{  
     color: red;  
 }  
 /*链接访问之后的状态*/  
 a:visited{  
     color: green;  
 }  
 /*鼠标放到链接上显示的状态*/  
 a:hover{  
     color: yellow;  
 }  
 /*链接激活的状态*/  
 a:active{  
     color: pink;  
 }
```

8.1.文本修饰

text-decoration: none | underline | line-through

9. 背景属性

9.1.background-color 背景颜色

9.2.background-image 背景图片

9.3.background-repeat repeat(默认) | no-repeat | repeat-x |

repeat-y 背景平铺

9.4.background-position left | right | center | top | bottom 背景定位

```
background-position: right;
```

★方位值只写一个的时候，另外一个值默认居中。

```
background-position: right bottom ;
```

★写2个方位值的时候，顺序没有要求。

```
background-position: 20px 30px ;
```

★写2个具体值的时候，第一个值代表水平方向，第二个值代表垂直方向。

9.5.Background-attachment 背景是否滚动 scroll | fixed

9.6.背景属性连写

```
/*背景属性连写*/  
background:red url("1.png") no-repeat 30px 40px scroll;
```

★：连写的时候没有顺序要求，url为必写项。

第三天

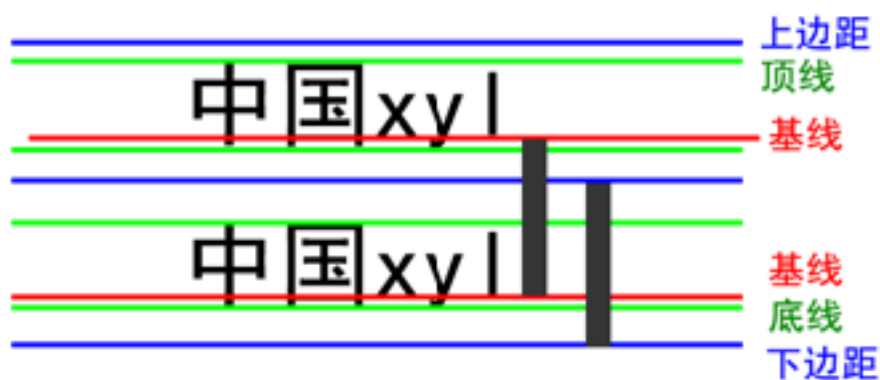
10. 行高

◆浏览器默认文字大小

浏览器默认文字大小：16px

行高：是基线与基线之间的距离

行高=文字高度+上下边距



一行文字行高和父元素高度一致的时候，垂直居中显示。

10.1.行高的单位

行高取值	文字大小	最后行高显示值
20px	20px	20px
2em	20px(* 2em)	40px
150%	20px(* 150%)	30px
2	20px(* 2)	40px

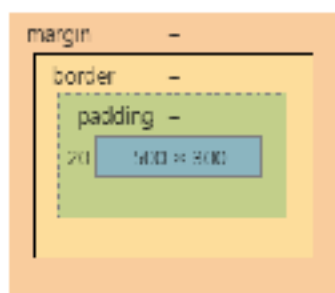
总结:单位除了像素以为，行高都是与文字大小乘积。

设置 的行高	父元素文字大小	子元素文字大小	最后 行高
40px	20px	30px	40px
2em	20px	30px	40px
150%	20px	30px	30px
2	20px	30px	60px

总结:不带单位时，行高是和子元素文字大小相乘，em和%的行高是和父元素文字大小相乘。行高以像素为单位，就是定义的行高值。

◆推荐行高使用像素为单位。

3. 盒子模型



3.1.边框 border

Border-top-style: solid 实线

dotted 点线

dashed 虚线

Border-top-color 边框颜色

Border-top-width 边框粗细

```
.box{  
  width: 300px;  
  height: 390px;  
  background: #999;  
  border-top-style: solid; /*线型*/  
  border-top-color: red; /*边框颜色*/  
  border-top-width: 5px;  
  border-bottom-style: dashed;  
  border-bottom-color: green;  
  border-bottom-width: 10px;  
}
```

◆边框属性的连写

特点：没有顺序要求，线型为必写项。

```
/*边框属性连写*/  
border-top: red solid 5px;
```

◆四个边框值相同的写法

```
/*四个边框值相同的写法*/  
border: 12px solid red;
```

特点：没有顺序要求，线型为必写项。

3.2.边框合并 border-collapse:collapse;

```
table{  
    width: 300px;  
    height: 500px;  
    border: 1px solid red;  
    border-collapse: collapse;  
}  
td{  
    border: 1px solid red;  
}
```

```
<table cellpadding="0px">  
  <tr>  
    <td></td>  
    <td></td>  
    <td></td>  
  </tr>  
</table>
```

3.3.获取焦点

```
.username{
    border: 0 none;          /*去掉边框*/
    outline-style: none;     /*去掉轮廓线*/
    background:#ccc;
    border:1px dashed green;
}
.username:focus{
    background:red;
}
```

label for id 获取光标焦点

```
<label for="username">用户名:</label><input type="text" class="
username" id="username"><br><br>
```

3.4.内边距

Padding-left | right | top | bottom

```
.box{
    padding-left:20px;
    padding-right:30px;
    padding-top:40px;
    padding-bottom: 50px;
    width: 500px;
    height: 300px;
    background: red;
}
</style>
<div class="box">14期威武</div>
```

◆padding连写

Padding: 20px; 上右下左内边距都是20px

Padding: 20px 30px; 上下20px 左右30px

Padding: 20px 30px 40px; 上内边距为20px 左右内边距为30px 下内边距为40

Padding: 20px 30px 40px 50px; 上20px 右30px 下40px 左 50px

◆内边距撑大盒子的问题

影响盒子宽度的因素：

- 1、内边距影响盒子的宽度；
- 2、边框影响盒子的宽度；

盒子的宽度=定义的宽度+边框宽度+左右内边距

◆继承的盒子 一般不会被撑大

包含（嵌套）的盒子，如果子盒子没有定义宽度，给子盒子设置左右内边距，一般不会撑大盒子,除非设置的边距值大于父盒子的宽度。

3.5.外边距

margin-left | right | top | bottom

```
.box{
  width:300px;
  height: 300px;
  background: #eee;
  margin-left: 20px;
  margin-right: 30px;
  margin-top: 40px;
  margin-bottom: 50px;
  /*外边距连写*/
  margin:20px 30px 40px 50px;
}
```

◆外边距连写

Margin: 20px; 上下左右外边距20PX

Margin: 20px 30px; 上下20px 左右30px

Margin: 20px 30px 40px; 上20px 左右30px 下 40px

Margin: 20px 30px 40px 50px; 上20px 右30px 下40px 左50px

◆垂直方向外边距合并

垂直方向两个盒子，下面设置上外边距，上面设置下外边距，最终取的设置较大的值。

◆嵌套的盒子外边距塌陷

嵌套的盒子，直接给子盒子设置垂直方向外边距的时候，会发生外边距坍塌。

解决方法: 1 给父盒子设置边框: border: solid 1px #000000;

2给父盒子设置overflow:hidden; bfc 格式化上下文
<http://www.w3cplus.com/css/understanding-bfc-and-margin-collapse.html>

4. Fireworks的基本使用

新建文件 ctrl+n

打开文件 ctrl+o

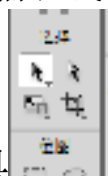
调出和隐藏标尺 ctrl+r

清除辅助线： 视图---辅助线---清除辅助线

放大镜 z 放大镜状态下alt+鼠标左键 缩小

抓手 快捷键 空格

测量距离 ★先拉出2根辅助线



★切换到指针工具

★将光标放到2根辅助线之间，按住shift键

第四天

4.1.行内元素可以定义左右的内外边距，上下会被忽略掉。

行内块可以定义内外边距。

5. 文档流（标准流）

元素自上而下，自左而右，块元素独占一行，行内元素在一行上显示，碰到父集元素的边框换行。

6. 浮动布局

float: left | right

特点：

- ★元素浮动之后不占据原来的位置（脱标）
- ★浮动的盒子在同一行上显示
- ★行内元素浮动之后转换为行内块元素。（不推荐使用，转行内元素最好使用display: inline-block;）

6.1.浮动的作用

- ◆文本绕图
- ◆制作导航
- ◆网页布局

7. 清除浮动

当父盒子没有定义高度，嵌套的盒子浮动之后，下边的元素发生位置错误。

- ◆清除浮动不是不用浮动，清除浮动产生的不利影响。

- ◆清除浮动的方法

clear: left | right | both

工作里用的最多的是clear:both;

- ★额外标签法

在最后一个浮动元素后添加标签。

```
<div class="main">  
  <div class="content"></div>  
  <div class="sidebar"></div>  
  
  <!-- 额外标签法 -->  
  <div style="clear:both;"></div>  
  
</div>
```

- ★给父集元素使用overflow:hidden; bfc

如果有内容出了盒子，不能使用这个方法。

```
.main{
    background: #eee;
    margin: 10px 0;
    overflow: hidden;
}

style>
>

div class="header"></div>
div class="main">
    <div class="content"></div>
    <div class="sidebar"></div>
```

伪元素使用
overflow:hidden;

★伪元素清除浮动 推荐使用

```
.clearfix:after{
    content: ".";
    display: block;
    height: 0;
    line-height: 0;
    visibility: hidden;
    clear: both;
}

/*兼容ie浏览器*/
.clearfix{
    zoom: 1;
}

</style>
head>
body>
<div class="header"></div>
<div class="main clearfix">
    <div class="left"></div>
    <div class="content">
        <div class="content-top"></div>
        <div class="content-bot"></div>
    </div>
    <div class="right"></div>
```

伪元素清除浮动

浮动，真的会消失于
世界吗？清除浮动
句

CSS初始化

腾讯：

```
body,ol,ul,h1,h2,h3,h4,h5,h6,p,th,td,dl,dd,form,fieldset,legend,input,textarea,select{margin:0;padding:0}
```

```
body{font:12px"宋体","Arial Narrow",HELVETICA;background:#fff;-webkit-text-size-adjust:100%;}
```

```

a{color:#2d374b;text-decoration:none}
a:hover{color:#cd0200;text-decoration:underline}
em{font-style:normal}
li{list-style:none}
img{border:0;vertical-align:middle}
table{border-collapse:collapse;border-spacing:0}
p{word-wrap:break-word}

```

新浪:

```

body,ul,ol,li,p,h1,h2,h3,h4,h5,h6,form,fieldset,table,td,img,div {margin:
0;padding:0;border:0;}
body {background:#fff;color:#333;font-size:12px; margin-top:5px;font-
family:"SimSun","宋体","Arial Narrow";}
ul,ol {list-style-type:none;}
select,input,img,select {vertical-align:middle;}
a {text-decoration:none;}
a:link {color:#009;}
a:visited {color:#800080;}
a:hover,a:active,a:focus {color:#c00;text-decoration:underline;}

```

淘宝:

```

body, h1, h2, h3, h4, h5, h6, hr, p, blockquote, dl, dt, dd, ul, ol, li, pre, form,
fieldset, legend, button, input, textarea, th, td { margin:0; padding:0; }
body, button, input, select, textarea { font:12px/1.5tahoma, arial,
\5b8b\4f53; }
h1, h2, h3, h4, h5, h6 { font-size:100%; }
address, cite, dfn, em, var { font-style:normal; }
code, kbd, pre, samp { font-family:couriernew, courier, monospace; }
small { font-size:12px; }
ul, ol { list-style:none; }
a { text-decoration:none; }
a:hover { text-decoration:underline; }
sup { vertical-align:text-top; }
sub { vertical-align:text-bottom; }
legend { color:#000; }
fieldset, img { border:0; }
button, input, select, textarea { font-size:100%; }
table { border-collapse:collapse; border-spacing:0; }

```


8. Overflow

<code>overflow:visible</code>	默认值。内容不会被修剪，会呈现在元素框之外。
<code>overflow:hidden</code>	内容会被修剪，并且其余内容是不可见的。
<code>overflow:scroll</code>	内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。
<code>overflow:auto</code>	如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。

9. 定位

定位方向: left | right | top | bottom

◆`position:static`; 静态定位。默认值，就是文档流。

```
/*静态定位*/  
position: static;
```

◆绝对定位

`Position:absolute`;

特点:

★元素使用绝对定位之后不占据原来的位置（脱标）

★元素使用绝对定位，位置是从浏览器出发。

★嵌套的盒子，父盒子没有使用定位，子盒子绝对定位，子盒子位置是从浏览器出发。

★嵌套的盒子，父盒子使用绝对定位，子盒子绝对定位，子盒子位置是从父元素位置出发。

★给行内元素使用绝对定位之后，转换为行内块。（不推荐使用，推荐使用`display:inline-block`;

◆相对定位

`Position: relative`;

特点:

★使用相对定位，位置从自身出发。

★还占据原来的位置。

★子绝父相（父元素相对定位，子元素绝对定位）（使用最多）

★行内元素使用相对定位不能转行内块

◆固定定位

Position:fixed;

特点:

★固定定位之后，不占据原来的位置（脱标）

★元素使用固定定位之后，位置从浏览器出发。

★元素使用固定定位之后，会转化为行内块（不推荐，推荐使用
display:inline-block;）

第五天

10. 定位的盒子居中显示

★:margin:0 auto; 只能让标准流的盒子居中对齐。

★定位的盒子居中：先左右走父元素盒子的一半50%，在向左走子盒子的一半(margin-left:负值。)

```
    }  
    .nav{  
        width: 960px;  
        height: 60px;  
        background:#666;  
        position: absolute;  
        bottom:0;  
        left:50%;  
        margin-left:-480px;  
    }  
    </style>  
</head>  
<body>  
    <div class="box">  
        <div class="nav"></div>  
    </div>
```

11. 标签包含规范

- ◆div可以包含所有的标签。
- ◆p标签不能包含div h1等标签。
- ◆h1可以包含p，div等标签。
- ◆行内元素尽量包含行内元素，行内元素不要包含块元素。



12. 规避脱标流

- ◆ 尽量使用标准流。
- ◆ 标准流解决不了的使用浮动。
- ◆ 浮动解决不了的使用定位。

尽量使用标准流案例：

子盒子设置margin-left:auto，子盒子相对父盒子对齐父盒子右边：

```
/*设置盒子左外边距为auto，将盒子冲到右边*/  
margin-left:auto;
```

margin-left:auto让盒子左侧充满

margin-right:auto让盒子右侧充满

margin: 0px auto; 居中对齐的由来；

13. 图片和文字垂直居中对齐

vertical-align对inline-block最敏感。默认属性是:vertical-align:baseline;

```
img{  
    /*图片和文字垂直居中对齐*/  
    vertical-align:middle;  
}
```

值	描述
baseline	默认。元素放置在父元素的基线上。
sub	垂直对齐文本的下标。
super	垂直对齐文本的上标。
top	把元素的顶端与行中最低元素的顶端对齐。
text-top	把元素的顶端与父元素字体的顶端对齐。
middle	把此元素放置在父元素的中部。
bottom	把元素的顶端与行中最低元素的顶端对齐。
text-bottom	把元素的顶端与父元素字体的顶端对齐。
length	
%	使用“line-height”属性的百分比值来排列此元素。允许使用负值。
inherit	规定应该从父元素继承 vertical-align 属性的值。

<https://www.zhangxinxu.com/wordpress/2010/05/%E6%88%91%E5%AF%B9css-vertical-align%E7%9A%84%E4%B8%80%E4%BA%9B%E7%90%86%E8%A7%A3%E4%B8%8E%E8%AE%A4%E8%AF%86%EF%BC%88%E4%B8%80%EF%BC%89/>

14. Css可见性

overflow:hidden; 溢出隐藏

visibility:hidden; 隐藏元素 隐藏之后还占据原来的位置。

display:none; 隐藏元素 隐藏之后不占据原来的位置。

Display:block; 元素可见

Display:none 和display:block 常配合js使用。

15. css之内容移除（网页优化）

◆使用text-indent:-5000em;

```

<style type="text/css">
    .box{
        width: 143px;
        height: 76px;
        background-image: url("images/logo.png");
    }
    a{
        width: 143px;
        height: 76px;
        display: inline-block;
        text-indent: -1000px;
    }
</style>
</head>
<body>
    <div class="box">
        <a href="http://www.baidu.com">搜狐</a>
    </div>
</body>

```

◆将元素高度设置为0,使用内边距将盒子撑开，给盒子使用 overflow:hidden;将文字隐藏。

```

<style type="text/css">
    a{
        display: inline-block;
        width: 143px;
        height: 0px;
        background: url("images/logo.png") no-repeat;
        padding-top: 76px;
        overflow: hidden;
    }
</style>
</head>
<body>
    <a href="http://www.baidu.com">搜狐</a>
</body>

```

效果图:



16. Css精灵图

17. 属性选择器

```
<style type="text/css">
  input[type=text][class]{
    width: 300px;
    height: 300px;
    background: red;
  }
```

18. emmet快捷键:

emmet是我们在sublime中的一个插件，在这个插件中集成很多的快捷键。

19. html:

19.1.生成结构的快捷键:

! + tab,可以生成html5的结构代码。

19.2.生成id名和类名

标签名.类名#id名+tab

没有标签名.类名+tab ==>div

19.3.生成同级元素:

标签名+标签名+标签名 “+”tab

19.4.生成子类标签

标签名>子标签名>子标签名>子标签名+tab

标签名>子标签名>子标签名>子标签名^^子标签名+tab

19.5.带固定数量的标签:

ul>li*5+tab

19.6.带有序号名称

ul>li.abc\$*3 + tab

19.7.生成带有内容的标签:

ul>li>a{item}*5

20. CSS

width:30px==>w30+tab

Height:30px==>h30+tab

Margin:30px==>mg30+tab

Padding:30px==> pd30+tab

Line-height:12px==>lh12px+tab

Background==>bg+tab

第六天

1、CSS滑动门