

CSS笔记1

CSS笔记1

1.快捷键

2.浏览器:

3.html标签

4.images 放图片的文件夹

5.网站头部的logo 外层套一个a标签,放链接跳转的地址

6. 块级元素和行内元素的区别? 分别列举?

1.css样式

2.css样式属性

3.块级元素和行内元素的区别? 分别列举?

【css引入方式】

【css样式】

【选择器】

【权重】- 谁的权利大

【css盒子模型】

a 标签属性行内元素:

【水平居中】

【问题】

【clear 表示清除浮动】

【浮动的方向】

1.快捷键

- win+E 打开资源文件
- F12 或者 鼠标右键 打开控制台
- file - new Project (新建文件项目仓库)
- 注释(不显示在浏览器中)：
 - ctrl+/ 单行注释
 - ctrl+shift+/ 多行注释
- 把光标移到想要复制行的任何一个地方：
ctrl+D 复制整行
- ctrl+Z 撤销 (回到上一步操作)

2.浏览器:

- 谷歌浏览器(调式我们的代码)
- 火狐浏览器
- IE浏览器

3.html标签

- 用尖括号包起来,有一个开始标签,有一个结束标签,形成一套完成的标签

1)标题类 h1~h6 h的下标越大 字号越小 h1最大 h6最小

4.images 放图片的文件夹

5.网站头部的logo 外层套一个a标签,放链接跳转的地址

- 外层套h1,h1在标题标签的权重最大
logo 背景有2种情况：

1) img 插入图片 鼠标光标移到img元素上 鼠标右键 open in new tab (在新窗口打开)-
ctrl+S 保存到本地文件夹中

例如:腾讯, 珠峰培训官网

2) 通过背景样式 引入到页面中 例如:网易

6. 块级元素和行内元素的区别? 分别列举?

块级元素：

- 1)从上到下排列,独占一行
- 2)ul/ol后面紧跟着只能是li标签
- 3)dl 后面跟紧跟着只能是dt/dd
- 4)p标签不能嵌套块级标签 包括它自己本身

div 划分快结构

h1~h6 标题类
p 文字段落
列表类 ul/li,ol/li,dl/dt/dd

行内元素：

2)从左到右,不独占一行
a,strong,b,em,i,span,sub,sup,input,button

行内块元素:img

1)可以设置宽度

1.css样式

【方式一:行内式】

把css样式写在标签里面

style="属性名:属性值;属性名:属性值"

多组属性名和属性值之间用分号隔开,如果最后一组属性名:属性值后面没有其他的值,最后的分号可以省略。

弊端:

html标签和css样式都放在页面,杂乱
没法公用,复用率比较低

【方式二:内嵌式】

把css样式和html结构分离开,放在style标签里面,style放到头部head标签里面,在title标签的下方

```
选择器{  
    属性名:属性值;  
    属性名:属性值;  
    .....  
}
```

好处:

可以实现样式的公用,复用率比行内式高

【选择器】

1)标签选择器:通过标签名选中元素

2.css样式属性

1)背景颜色:background-color

2)字体颜色:color

#fff 白色

#000 黑色

3)字体的粗细:font-weight

normal 正常(默认值)

bold 加粗

100~900 值越小 字体越细 反之 字体越粗

4)字体大小:font-size 单位:px(像素pixel)

5)段落缩进(文本缩进):text-indent 单位:em

一个em相当于一个字的间距

6)边框线：

button默认的边框线,我们要去掉它默认的边框线,border:0

####增加边框线:

- 复合属性的写法:

border:border-width border-style border-color;

例如: border:1px solid #000;

- 拆分开的写法:
 - border-width 边框的宽度
 - border-style 边框线的类型
 - solid 实线
 - dashed 虚线
 - dotted 点状线
 - border-color 边框线的颜色
- 指定某一个方向的边框线:
 - border-bottom:1px solid #000 下边框线
 - border-top:1px solid #000 上边框线
 - border-left:1px solid #000 左边框线
 - border-right:1px solid #000 右边框线

7)a标签有默认的下划线

去掉默认的下划线:text-decoration: none

8)内边距 padding

padding-top 上内边距

padding-bottom 下内边距

padding-left 左内边距

padding-right 右内边距

- 复合属性padding:10px 20px 30px 40px;
上 右 下 左 (顺时针方向)
- 第一种情况:4个值:
padding:10px 20px 30px 40px;
上 右 下 左 (顺时针方向)
padding-top:10px
padding-right:20px;
padding-bottom:30px;
padding-left:40px;
- 第二种情况:3个值
padding:10px 20px 30px;
上 左/右 下
padding-top:10px;
padding-right:20px; padding-left:20px;
padding-bottom:30px;
- 第三种情况:2个值
padding:10px 20px;
上/下 左右
padding-top:10px; padding-bottom:10px
padding-left:20px; padding-right:20px;

- 第四种情况:1个值;

`padding:10px;`

上/右/下/左 都是10px

`padding-top:10px; padding-bottom:10px`

`padding-left:10px; padding-right:10px;`

9)文字对齐居中方式

`text-align:center` 水平居中对齐

`text-align:left` 左对齐

`text-align:right` 右对齐

10)去掉列表ul前面的小黑点和ol前面的数字:

`list-style:none`

11)宽度 `width`

高度 `height`

行高 `line-height`

可以控制字垂直间距,行高和高度值一样,垂直居中;

3.块级元素和行内元素的区别？分别列举？

块级元素：

- 1)从上到下排列,独占一行
- 2)ul/ol后面紧跟着只能是li标签
- 3)dl 后面跟紧跟着只能是dt/dd
- 4)p标签不能嵌套块级标签 包括它自己本身
- 5)宽度默认是父级元素的宽度

- div 划分快结构
- h1~h6 标题类
- p 文字段落
- 列表类 ul/li,ol/li,dl/dt/dd

行内元素：

- 1)从左往右排列,不独占一行
- 2)行内元素的宽高是它本身内容的宽高,设置宽高不起作用,display:block;

WebStorm快捷键代码格式化:

code-Reformat code (ctrl+Alt+L)

【css引入方式】

1.行内式

弊端:杂乱 没法公用

2.内嵌式

实现html结构和css样式的分离 样式的公用

```
选择器{  
    属性名:属性值;  
    属性名:属性值;  
    .....  
}
```

3.外链式 项目中最常用的css引入方式
href="css样式的文件名"

【css样式】

- 1.浏览器自带的样式 禁灰的状态 我们修改不了
- 2.我们自定义的样式 可以修改

【选择器】

1.标签选择器 - 通过标签名来选择元素

2.class选择器也叫类选择器 - '.',页面上class选择器可以出现多次,复用率高,有需要公用的样式,只需要在标签里面加相同的class名即可!

3.id选择器 - '#'

注意:一个页面中,相同的id名只能出现一次,id唯一(身份证),否则会报错

4.通配符选择器 - '*' 指的选中页面上所有的元素

常用来去掉浏览器默认自带的css样式,一般放在所有样式的最上面;

5.后代选择器 - '.list dt{'

最多保留3个选择器的层级

6.相邻兄弟选择器 '+'

.box+.box 只有+后面的兄弟元素才起作用

7.交集选择器 标签名和class/id 并在一起 用来提高权重

【权重】 - 谁的权利大

1. 标签选择器 权重 1
 2. class选择器 权重 10
 3. id选择器 权重 100
 4. 通配符选择器 权重 0~1
 5. 关系选择器(后代选择器, 相邻兄弟选择器, 交集选择器)
- 权重 计算: 所有单独选择器的权重之和

`.list1 li span em` $10+1+1+1=13$

`.box+.box` $10+10=20$

`div#box` $1+100=101$

【css盒子模型】

最里层: 内容的宽度(width)和高度(height)

第二层:内边距

padding 上 右 下 左 (顺时针方向)

取值分下面几种情况:

- 1.1个值 例如:padding:10px;
上=右=下=左=10px
- 2.2个值 例如:padding:10px 20px;
上=下=10px
左=右=20px
- 3.3个值 例如:padding:10px 20px 30px;
上=10px;
左=右=20px;
下=30px;
- 4.4个值 例如:padding:10px 20px 30px 40px;
上=10px
右=20px
下=30px
左=40px

第三层:边框层 border

- 去掉边框: `border:0;`
- 增加边框:
 - `border-width` 边框的宽度
 - `border-style` 边框的类型
 - `solid` 实线
 - `dashed` 虚线
 - `dotted` 点状线
 - `border-color` 边框颜色 `#000` `black` `rgb(0,0,0)`
- 复合属性:
 - `border:1px solid #000;`
 - `border-top` 上边框线
 - `border-bottom` 下边框线
 - `border-left` 左边框线
 - `border-right` 右边框线

第四层：

外边距(块与块之间的间距) margin

上 右 下 左(顺时针)

取值分下面几种情况:

- 1.1个值 例如:margin:10px;
上=右=下=左=10px
 - 2.2个值 例如:margin:10px 20px;
上=下=10px
左=右=20px
 - 3.3个值 例如:margin:10px 20px 30px;
上=10px;
左=右=20px;
下=30px;
 - 4.4个值 例如:margin:10px 20px 30px 40px;
上=10px
右=20px
下=30px
左=40px
- 块级元素水平居中 margin:0 auto;

a 标签属性行内元素:

特点:

- 1)从左往右依次排列,不独占一行
- 2)内容超过父元素的最大宽度,会自动折行
- 3)设置宽高不起作用,要设置宽高,需要把它转化为块级元素或者行内块元素,加上属性display:block或者 display:inline-block;
- 4)行内元素设置padding和margin上下浏览器可以识别但是不起作用,左右可以起作用;

【display:block 和display:inline-block的区别】

display:block 把行内元素转化为块级元素 ,独占一行

display:inline-block 把行内元素转化为行内块元素,不独占一行,还可以设置宽高;

【面试题】

display:inline-block 有默认几像素的间距,解决办法?

給它的直接父元素加上一个属性:font-size:0;(字体大小为0)

【水平居中】

1.块级元素水平居中 `div`

`margin:0 auto;`

2.文字内容

水平居中 `text-align:center` 文本对齐

垂直居中,高度和行高一致(仅限单行文字内容)

例如:`height:40px;`

`line-height:40px;`

【问题】

1.设置`display:inline-block` 和`float` 这2种方式有什么区别？

相同点:都可以实现让元素在一行显示

区别:

`display:inline-block`

问题:默认的间距空白

解决办法:给它直接父级加上`font-size:0`

=

float 浮动

弊端:

设置border和padding 不起作用;

解决办法: 需要清除浮动(有4种方法可以清除浮动)

float:left 跟相当于转化为了块级元素,有了**float:left**;就不需要再加**display:block**;

=

【浮动产生的副作用-对象:浮动元素的父元素】

- 1.设置背景颜色不起作用; background-color
- 2.设置边框线没有被撑开,边框线合并在一起了; border
- 3.设置内边距不起作用; padding

=

【* 面试题:如何解决浮动产生的负作用 – 清除浮动】

方法一: 给浮动元素的直接父级加

`overflow:hidden overflow: auto;`

隐藏溢出容器的内容且不出现滚动条 - 多的内容溢出隐藏

方法二:给浮动元素的直接父级加高度属性 `height`

方法三:在浮动元素的直接父级结束标签之前加一个空的div,并且给新添加的div加上一个

`class='clear'`,然后在css样式里面加上一个属性 `clear:both;`

=

【clear 表示清除浮动】

1)clear:both 清除左边和右边的浮动 不允许有浮动对象;

2)clear:left 清除左边的浮动 不允许有左边的浮动对象;

3)clear:right 清除右边的浮动 不允许有右边的浮动对象;

4)clear:none 允许有左右两边的浮动对象;

【浮动的方向】

float:left 对象浮动在左边

float:right 对象浮动在右边

float:none 对象不浮动(默认值)