## css的四种导入方式

**head**>

<style>

p{ color: brown ;font-size: 40px } <!--//对 p标签进行css修改-->

a{ text-decoration: dashed } <!--/\*去掉a链接的下划线\*/-->

</style>

<!--第三种方法 ：链接css的文件-->

<link href="css笔记的css代码.css" rel="stylesheet" type="text/css">

<!--第四种方法 import导入式-->

<**style**>  
 **@import "css笔记的css代码.css"**;  
 </**style**>  
</**head**>  
<**body**>  
<**p**>css，js重点：1 怎么找到标签，2如何做作标签对象（elemnt）</**p**>   
<**p**>css：层叠样式表，用来控制网页数据的表现（渲染一个标签），可以使网页的表现与数据内容分离</**p**>  
<**hr**>   
<**dl**>  
 <**dt**><**h2**>1.css 引入方式</**h2**></**dt**>  
 <**dd**>1.1 ：   
 <**div style="color**:**red**;**background-color**:**darkgray"**>第一种引用方式，最好不要使用</**div**>  
 </**dd**>  
 <**dd**> 1.2 ：在head的style标签里添加引用  
 <**div**>在head的style标签控制div标签</**div**>  
 <**p**>在head的style标签控制P标签</**p**>  
 <**a href=""**>在head的style标签控制标签</**a**>  
 </**dd**>

<**dd**> 1.3 ：把css代码单独放到一个文件里，通过html语言方式引用，link引入的次数不限，重点使用。  
 </**dd**>

<**dd**> 1.4 ：把css代码导入式的方法引用，按照css语言代码引入，import导入的次数是有限制的  
</**dd**>  
</**dl**>  
</**body**>

## css的四种选择器

### 2.1 基础选择器

<**head**>  
 <**style**>

/\* 1.1 \*为通用选择器 \*/

\*{ color: red;

}

/\* 1.2 E 标签选择器 \*/

div{ color: aquamarine;  
 }

/\*1.3 id选择器\*/  
 **#pid-11**{ **background**:**yellow**;  
 }

/\*1.4 class 选择器\*/  
 .**a-l1**{ **color**: **red**;  
 }  
 **div**.**div-11**{ **color**: **firebrick**;  
 }

</**style**>

</**head**>

<**ol**> <**li**>1.基础选择器-可混合使用   
 1.1 \*为通用选择器，匹配仍和元素 \*{ margin:0;padding:0 }   
 1.2 E 标签选择器，匹配所欲使用E标签的元素 .eg: p { color:green; }   
 1.3 id选择器， #info和E#info 匹配所有id属性等于footer的元素。 eg： #info{background:#ff0;}   
 1.4 class 选择器， .info和 E.info 匹配所有class属性中包含info的元素。 eg： .info{background:#ff0;}   
 <**div**> 1.2.基础选择器,E 标签选择器测试</**div**>  
 <**p id="pid-11"**>1.3.基础选择器,id选择器测试</**p**>  
 <**p id="pid-12"**>id选择器需要用 #id值来调用</**p**>  
 <**a class="a-l1"**>1.4基础选择器,class选择器测试</**a**>   
 <**a class="a-l2"**> class 选择器 需要用 .class名来调用 class名可以重复</**a**>  
 <**div class="div-11"**>E 标签选择器和class 选择器 混合使用方法</**div**>  
 </**li**>

### 组合选择器

<**style**>

*/\* 2.1 多元素选择器\*/* **#pid-21**,**a**.**a-21**{  
 **background**: **#9f67a9**;  
 }  
*/\* 2.2 E F 后代元素选择器--div内的第一个div标签\*/* .**div-21 div**{ **color**: **red**;  
 }  
*/\* 2.1 E F 后代元素选择器--div内的第二个div标签\*/* **div-21** .**div-212**{

**color**: **#ffff30**;  
 }  
*/\*2.3 E>F 子元素选择器--div-div-内的第二个p标签<\*/* .**div-21 div**>**#P2**{  
 **color**: **#233aff**;  
 }  
 .**div-21 div**+**p**{  
 **color**: **chartreuse**;  
 }

</**style**>

<**ol**><**li**>2. 组合选择器   
 2.1 E,F 多元素选择器，同时匹配带所有E元素或F元素，E和F之间用逗号分隔   
 2.2 E F 后代元素选择器，匹配所有属于E元素后代的F元素，E和F之间用空格分隔  
 2.3 E>F 子元素选择器，匹配所有E元素的子元素F   
 2.4 E+F 毗邻选择器，匹配所有紧随E元素之后的同级元素F   
 <**p id="pid-21"**>E,F 多元素选择器 测试</**p**>  
 <**a class="a-21"**>E,F 多元素选择器 测试</**a**>  
 <**div class="div-21"**>三层标签嵌套组合:  
 <**div**>div内的第一个div标签   
 <**a href=""**>div-div-内的a标签</**a**>  
 <**p id="P1"**>div-div-内的第一个p标签</**p**>  
 <**p id="P2"**>div-div-内的第二个p标签</**p**>  
 <**div**>div-div-内的div标签</**div**>  
 </**div**>  
 <**div class="div-212"**>div内的第二个div标签</**div**>  
 <**p**>与div-div同级的p标签</**p**>  
 </**div**>

</**li**></**ol**>

嵌套规则：块级元素可以包含内联元素或某些快级元素，但内联元素不能包含块级元素，它只能包含其他内联元素.li内可以包含div

### 2.3属性选择器

<style>

*/\*3.1 E [att]\*/*

[**alex**]{  
 **color**: **#233aff**;

}

*/\*3.2 E [att=val]\*/*

**div**[**alex**=**div-31**]{  
 **color**: **#ffb275**;  
}  
 **p**[**alex**=**div-31**]{

**color**: **#ff171c**;  
}

[**alex**~=**div-322**]{  
 **color**: **#58fdff**;  
}  
 [**alex**^=**s**]{  
 **color**: **#9f67a9**;  
}

</style>

3.1 E [att] 匹配所有具有att属性的E元素，不考虑它的值（E在此处可以省略，）eg：p[title]{ color：#f00;}<**br**>  
 3.2 E [att=val] 匹配所有att属性等于“val”的E元素. eg:div[class="error"]{ color：#f00;}<**br**>  
 3.3 E [att~=val] 匹配所有att属性具有多个空格分隔的值，其中一个值等于“val”的E元素。eg:td[class~="name"]{ color：#f00;} <**br**>  
 3.4 E [att^=val] 指定属性值以指定值开头的每个元素。eg:div[class^="test"]{ color：#f00;} <**br**>  
 3.5 E [att$=val] 指定属性值以指定值结尾每个元素。eg:div[class$="test"]{ color：#f00;} <**br**>  
 3.6 E [att\*=val] 指定属性值包含指定值的每个元素。eg:div[class\*="test"]{ color：#f00;} <**br**>  
  
 <**div alex**>自定义属性3.1测试</**div**><**br**>  
 <**div alex="div-31"**>自定义属性3.2测试</**div**><**br**>  
 <**p alex="div-31"**>自定义属性3.22测试</**p**><**br**>  
 <**div alex="div-32 div-321 div-322"**>自定义属性3.3测试</**div**><**br**>  
 <**div alex="siv-33 siv-331 siv-332"**>自定义属性3.4测试</**div**><**br**>  
 <**div**>3.5,3.6同上</**div**><**br**>  
</**li**>  
 <**br**><**br**>

### 2.4css伪类

<**style type="text/css"**>

*/\*没有接触过的链接\*/* **a**:**link**{  
 **color**: **#ff171c**;  
 }  
*/\*访问过的链接\*/* **a**:**visited**{  
 **color**: **#ff21d8**;  
 }  
*/\*鼠标放在链接上的状态\*/* **a**:**hover**{  
 **color**: **#233aff**;  
 }  
*/\*在连接上按下鼠标时的状态\*/* **a**:**active**{  
 **color**: **chartreuse**;  
 }  
 .**box**{  
 **width**: 100**px**;  
 }  
  
 .**top**,.**button**{  
 **width**: 100**px**;  
 **height**: 100**px**;  
 **background**: **#ffb275**;  
 }  
 */\*.top:hover{\*/  
 /\*background: cadetblue;\*/  
 /\*}\*/  
/\*鼠标悬浮在盒子上面上部分就会变颜色\*/* .**box**:**hover** .**top**{  
 **background**: **hotpink**;  
 }  
</**style**>  
<**style**>  
 .**add**:**after**{  
 **content**: **"通过css在class属性为add的主题后面加字,和其他渲染"**;  
 **color**: **#58fdff**;  
 }  
</**style**>

<**li**>css伪类<**br**>  
 css伪类是用来给选择器添加一些特殊的效果。<**br**>  
 anchor伪类：专用于控制链接的显示效果.<**br**>  
 a:link（没有接触过的链接），用于定义了链接的常规状态<**br**>  
 a:hover(鼠标放在链接上的状态)，用于产生视觉效果。<**br**>  
 a:visited（访问过的链接），用于阅读文章，能清楚的判断已经访问过的链接。<**br**>  
 a:activte（在连接上按下鼠标时的状态），用于表现鼠标按下时的链接状态。<**br**>  
 eg：<**br**>  
 <**a href="css导入的四种方式.html"**>更改链接样式</**a**><**br**><**br**>  
 <**div class="box"**>  
 <**div class="top"**></**div**>  
 <**div class="button"**></**div**>  
 </**div**>  
 </**li**>  
 </**ol**>  
<**p class="add"**>hello world</**p**>

## css的display属性



<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>display</**title**>  
 <**style**>

*/\*共用的可以一起用\*/* **div**,**p**,**a**,**span**{  
 **width**: 100**px**;  
 **height**: 100**px**;  
 }  
 **div**{  
 **background-color**: **#ff171c**;  
 }  
 **p**{  
 **background-color**: **#ffff30**;  
  
 }  
 */\*span和a为内联标签,所以不能设置长和宽,但可以通过display来更改\*/* **span**{  
 **background-color**: **#233aff**;  
 **display**: **block**;  
 }  
 **a**{  
 **background-color**: **chartreuse**;  
 **isplay**: **inline-block**; */\*d既有内联标签不独占一行又又外联标签可以设置长宽特性\*/* }  
 </**style**>  
</**head**>  
<**body**>  
<**div**>divvvv</**div**>  
<**p**>pppppp</**p**>  
<**span**>内联</**span**>  
<**a href="#"**>aaaa</**a**>  
</**body**>  
</**html**>

## css的float属性

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>float作定位用的属性</**title**>  
 <**style**>  
 */\*float: left;让div块向左飘起来\*/* .**div1**{  
 **width**: 100**px**;  
 **height**: 100**px**;  
 **background-color**: **#233aff**;  
 **float**: **left**;  
 }  
 .**div2**{  
 **width**: 200**px**;  
 **height**: 200**px**;  
 **background-color**: **#b1ff4b**;  
 **float**: **left**;  
 }  
 .**div3**{  
 **width**: 300**px**;  
 **height**: 300**px**;  
 **background-color**: **#27ffc3**;  
  
 }  
 .**div4**{  
 **width**: 400**px**;  
 **height**: 400**px**;  
 **background-color**: **#ff1f2d**;  
 }  
*/\*清楚浮动\*/* .**clearfix**:**after**{  
 **content**:**"hello"**;  
 **display**: **block**;  
 **clear**: **both**;  
 }  
 </**style**>  
</**head**>  
<**body**>  
文档流:指的是元素排版布局过程中,元素会自动从左往右,从上往下的流式排列<**br**>  
脱离文档流:将元素从普通布局排版中拿走,不会按照上面的方法排列<**br**>  
只有绝对定位的absolute和浮动的float才可以脱离文档流.<**br**>  
  
快写方式: div .\*4 +tab  
  
<**div class="div1 clearfix"** > hello world</**div**>  
<**div class="div2"**> </**div**>  
<**div class="div3"**> </**div**>  
<**div class="div4"**> </**div**>  
</**body**>  
</**html**>

## 5.css的position定位属性

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>定位</**title**>  
 <**style**>  
 . **\***{  
 **margin**: 0**px**;  
 }  
 .**div1**{  
 **width**: 200**px**;  
 **height**: 100**px**;  
 **background-color**: **#233aff**;  
 */\*如果不写就默认static，无定位，按照文本流格式排列\*/* **position**: **static**;  
 }  
 .**div2**{  
 **width**: 200**px**;  
 **height**: 200**px**;  
 **background-color**: **#b1ff4b**;  
 */\*距离原来自己相对的位置再动，还没脱离文档流\*/* **position**: **relative**;  
 */\*left: 100px;\*/  
 /\*top: 100px;\*/* }  
 .**div3**{  
 **width**: 200**px**;  
 **height**: 300**px**;  
 **background-color**: **#27ffc3**;  
 */\*设置绝对定位，从文档流脱离，悬浮在其它框上面，以body来定位原点\*/* **position**: **absolute**;  
 **left**: 100**px**;  
 **top**: 100**px**;  
  
 }  
 .**div4**{  
 **width**: 200**px**;  
 **height**: 400**px**;  
 **background-color**: **#ff1f2d**;  
 **position**: **fixed**;  
  
 }  
 */\*设置固定死的悬浮标签\*/* .**returntop**{  
 **width**: 80**px**;  
 **height**: 50**px**;  
 **position**: **fixed**;  
 **bottom**: 20**px**;  
 **right**: 5**px**;  
 **background-color**: **#ff171c**;  
 **color**: **bisque**;  
 }  
  
 </**style**>  
</**head**>  
<**body**>  
<**div class="outer"**>  
 <**div class="div1"**>1</**div**>  
 <**div class="div2"**>2</**div**>  
 <**div class="div3"**>3</**div**>  
 <**div class="div4"**>4</**div**>  
</**div**>  
  
<**div class="returntop"**>返回顶部</**div**>  
</**body**>  
</**html**>

## css的边框属性

<!DOCTYPE **html**>  
<**html lang="en"**>  
 <**meta charset="UTF-8"**>  
 <**title**>Title</**title**>  
 <**style**>  
 .**div1**{  
 **width**: 200**px**;  
 **height**: 200**px**;  
 **border**: 5**px dashed blueviolet**;  
 **border-color**: **#ff171c**;  
 **border-style**: **solid**;  
 **border-width**: 5**px**;  
 **border-left-color**: **#ffff30**;  
 **border-right-color**: **chartreuse**;  
 }  
 **ol**,**ul**{  
 **list-style**:**none***/\*squarecircle; 前面列表便签的样式\*/* }  
 </**style**>  
  
</**head**>  
<**body**>  
<**div class="div1"**>边框属性</**div**>  
<**div class="div2"**>列表属性  
<**ul**>  
 <**li**>111</**li**>  
 <**li**>222</**li**>  
 <**li**>333</**li**>  
</**ul**>  
<**ol**>  
 <**li**>111</**li**>  
 <**li**>222</**li**>  
 <**li**>333</**li**>  
</**ol**>  
</**div**>  
  
<**div**>外边距和内边距</**div**>  
外边距 border: 围绕在内边距和内容外的边框  
内边距 padding: 用于控制内容与边框之间距离  
margin: 用于控制元素与元素之间的距离  
content: 盒子的内容,显示文本和图像  
  
</**body**>  
</**html**>

1、边框阴影属性：box-shadow :

h-shadow : 水平阴影距离

v-shadow : 垂直阴影距离

blur : 可选，模糊距离

spread : 可选， 阴影的尺寸

color : 可选，颜色

inset : 可选值,将当前阴影修改为内阴影

语法规范：box-shadow:h-shadow v-shadow blur spread color inset；

也可简写为：box-shadow:h-shadow v-shadow blur color；

举个栗子： box-shadow：1px 1px 1px 1px red;

2、图片边框属性： border-image:

border-image-source : 图片路径

border-image-width:图片边框的宽度

border-image-repeat : 边框的平铺方式

取值 ：repeat : 平铺 round : 铺满 stretch : 拉伸 语法规范：border-image:source width repeat;

3、轮廓，即绘制于元素周围的一条线，位于边框的外围

语法规范： outline : width style color；

outline-width : 轮廓宽度

outline-style : 轮廓的样式

outline-color : 轮廓的颜色