CSS知识点

1.标签选择器

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
   外部引用 <link rel="stylesheet" href="test1.css">
    内部引用：<style>
    h1{                                标签选择器的基本用法
        color: red;
    }
    </style>
</head>
<body>
    <h1>努力</h1><br>
</body>
</html>
```

2.类名选择器

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
   外部引用 <link rel="stylesheet" href="test1.css">
    内部引用：<style>
```

```
    .change-color{                                  类选择器的基本用法
        color: red;
    }
    .change-font{
        font-size:100px;
    }
    </style>
</head>
<body>
    <div class="change-color change-font">努力</div><br>
</body>
</html>
```

类名选择器可以有多个类名，类名之间用空格隔开.

3.id选择器

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
   外部引用 <link rel="stylesheet" href="test1.css">
    内部引用：<style>
    #change-color{                                  id选择器的基本用法
        color: red;
    }
    </style>
</head>
<body>
    <div id="change-color">努力</div><br>
</body>
</html>
```

id的命名是唯一的不能重复，只能使用一次

4.伪类选择器

向某些选择器添加特殊效果

4.1 链接伪类选择器

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
   外部引用 <link rel="stylesheet" href="test1.css">
    内部引用：<style>
    a:link{                 未访问的状态
        color: blue;
        font-size:13px;
    }
      a:visited{                已访问的状态
        color: yellow;
        font-size:13px;
    }
      a:hover{                鼠标悬停时的状态
        color: black;
        font-size:15px;
    }
      a:active{                鼠标点击时的状态
        color: white;
        font-size:16px;
    }
    </style>
</head>
<body>
    <div id="change-color"><a href="#">点击</a></div><br>
</body>
</html>
```

- 如果需要给超链接定义：访问前，鼠标悬停，当前被点击，已访问这4种伪类效果，而又没有按照一致的书写顺序，不同的浏览器可能会有不同的表现
- 超链接的4种状态，需要有特定的书写顺序才能生效

- 超链接状态顺序：

```
a:link {} a:visited {} a:hover {} a:active {}
```

注意，a:hover 必须位于 a:link 和 a:visited 之后，a:active 必须位于 a:hover 之后
可靠的顺序是：l(link)ov(visited)e h(hover)a(active)te, 即用喜欢(love)和讨厌(hate)两个词来概括

4.2结构（位置）伪类选择器（css3）

```
<style>
li:nth-child(even){color:#f00;} /* 偶数 */
li:nth-child(odd){color:#000;} /* 奇数 */
</style>

<ul>
    <li>列表项一</li>
    <li>列表项二</li>
    <li>列表项三</li>
    <li>列表项四</li>
</ul>
```

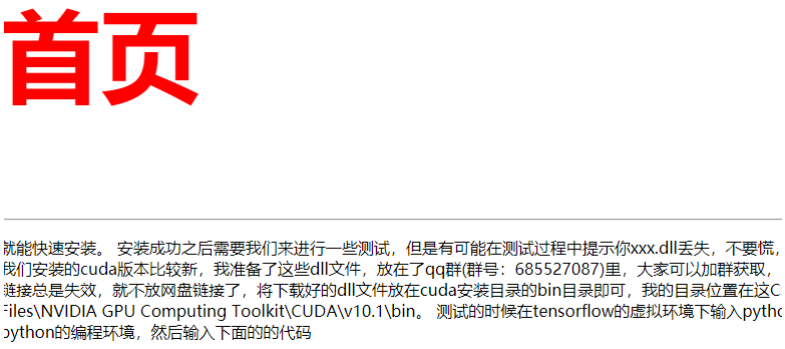
```
E:first-child        匹配父元素的第一个子元素E。
E:last-child         匹配父元素的最后一个子元素E。
E:only-child         匹配父元素仅有的一个子元素E。
E:nth-child(n)       匹配父元素的第n个子元素E。    可以是2n ,2n-1,3n 等 （n从0开始）
E:nth-last-child(n) 匹配父元素的倒数第n个子元素E。
```

4.3目标伪类选择器

定位id的使用：target

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
    <style>
        :target{
            color: red;
            font-size: 100px;
        }
    </style>
</head>
<body>
    <div>
        <h2>导航</h2>
        <ol>
            <li><a href="#star">首页</a></li>
            <li><a href="#center"> 中心 </li>
            <li> <a href="#big">扩张</a> </li>
            <li> <a href="#my">个人中心</a> </li>
        </ol>
    </div>
        <h3 id="star">首页</h3>
    .....
    </body>
</html>
```



5.文字样式

line-height:10px (越小行间距越大)

text-align:left/center/right (对齐方式)

text-indent：2em (两个汉字的距离)定义块内文本内容的缩进

letter-spacing：2px (字间距)

word-spacing:2px(单词间距，只对英文有效)

text-shadow：1px 2px 0px rgba(0,0,0,0.1)　　定义文字是否有阴影及模糊效果

第1个长度值用来设置对象的阴影水平偏移值。可以为负值

第2个长度值用来设置对象的阴影垂直偏移值。可以为负值

如果提供了第3个长度值则用来设置对象的阴影模糊值。不允许负值

设置对象的阴影的颜色。

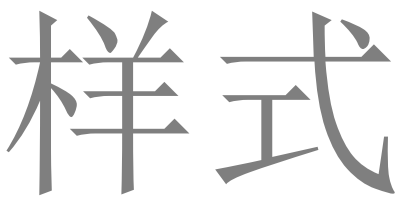
6.颜色透明度

color：rgba(r,g,b,a) a是透明度 取值在0·1之间

7.三种样式表

（1）行内样式表 使用少　写在head标签里

（2)内嵌样式表　使用少



(3)外部样式表 使用多 推荐

8.块级与行内标签

（1）块级标签：

-

，

，

,列表等，
为其典型

块级标签通常独占一行或多行，可对其设置长，宽**(**默认**100%)**，高，对齐属性，用于网页布局与结构

（2)行内标签：,,,~~.....等 是其典型~~

不占独立区域，仅仅靠里面的文字大小和图片尺寸支撑其大小，一般不设置宽，高，对齐方式

行内元素只能容纳文本或其他行内标签，但是不能套

只有文字才能组成段落,

不能放块级元素，同理

块级元素的特点：
（1）总是从新行开始
（2）高度，行高、外边距以及内边距都可以控制。
（3）宽度默认是容器的100%
（4）可以容纳内联元素和其他块元素。

行内元素的特点：
（1）和相邻行内元素在一行上。
（2）高、宽无效，但水平方向的padding和margin可以设置，垂直方向的无效。
（3）默认宽度就是它本身内容的宽度。
（4）行内元素只能容纳文本或则其他行内元素。

(3)行内块标签

在行内元素中有几个特殊的标签—<img />、<input />、<td>，可以对它们设置宽高和对齐属性，有些资料可能会称它们为行内块元素。

行内块元素的特点：
（1）和相邻行内元素（行内块）在一行上，但是之间会有空白缝隙。
（2）默认宽度就是它本身内容的宽度。
（3）高度，行高、外边距以及内边距都可以控制。

（4）display属性

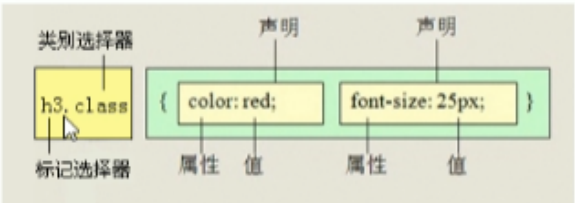
取值：

none： 隐藏对象。与 'visibility' 属性的 「hidden」 值不同，其不为被隐藏的对象保留其物理空间
inline： 指定对象为内联元素。
block： 指定对象为块元素。
list-item： 指定对象为列表项目。
inline-block： 指定对象为内联块元素。（CSS2）
table： 指定对象作为块元素级的表格。类同于html标签<table>（CSS2）
inline-table： 指定对象作为内联元素级的表格。类同于html标签<table>（CSS2）
table-caption： 指定对象作为表格标题。类同于html标签<caption>（CSS2）
table-cell： 指定对象作为表格单元格。类同于html标签<td>（CSS2）
table-row： 指定对象作为表格行。类同于html标签<tr>（CSS2）
table-row-group： 指定对象作为表格行组。类同于html标签<tbody>（CSS2）
table-column： 指定对象作为表格列。类同于html标签<col>（CSS2）
table-column-group： 指定对象作为表格列组显示。类同于html标签<colgroup>（CSS2）
table-header-group： 指定对象作为表格标题组。类同于html标签<thead>（CSS2）
table-footer-group： 指定对象作为表格脚注组。类同于html标签<tfoot>（CSS2）
run-in： 根据上下文决定对象是内联对象还是块级对象。（CSS3）
box： 将对象作为弹性伸缩盒显示。（伸缩盒最老版本）（CSS3）
inline-box： 将对象作为内联块级弹性伸缩盒显示。（伸缩盒最老版本）（CSS3）
flexbox： 将对象作为弹性伸缩盒显示。（伸缩盒过渡版本）（CSS3）
inline-flexbox： 将对象作为内联块级弹性伸缩盒显示。（伸缩盒过渡版本）（CSS3）
flex： 将对象作为弹性伸缩盒显示。（伸缩盒最新版本）（CSS3）
inline-flex： 将对象作为内联块级弹性伸缩盒显示。（伸缩盒最新版本）（CSS3）

9. css复合选择器

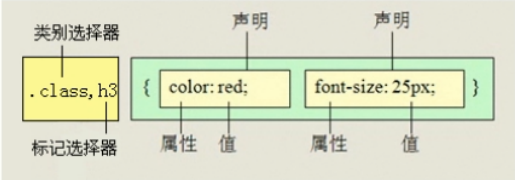
1.交集选择器

交集选择器由两个选择器构成，其中第一个为标签选择器，第二个为class选择器，两个选择器之间不能有空格，如h3.special。



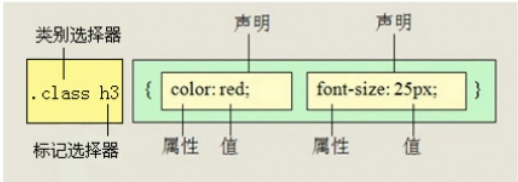
2.并集选择器

并集选择器（CSS选择器分组）是各个选择器通过逗号连接而成的，任何形式的选择器（包括标签选择器、class类选择器id选择器等），都可以作为并集选择器的一部分。如果某些选择器定义的样式完全相同，或部分相同，就可以利用并集选择器为它们定义相同的CSS样式。



3.后代选择器

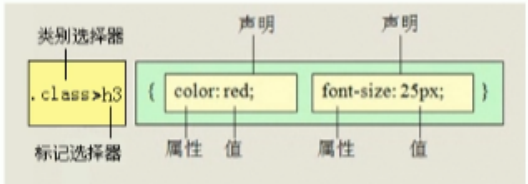
后代选择器又称为包含选择器，用来选择元素或元素组的后代，其写法就是把外层标签写在前面，内层标签写在后面，中间用空格分隔。当标签发生嵌套时，内层标签就成为外层标签的后代。



4.子元素选择器

子选择符只能命中子元素，而不能命中孙辈。

子元素选择器只能选择作为某元素子元素的元素。其写法就是把父级标签写在前面，子级标签写在后面，中间跟一个 > 进行连接，注意，符号左右两侧各保留一个空格。



5.属性选择器

选取标签带有某些特殊属性的选择器。

选择符	版本	描述
E[att]	CSS2	选择具有att属性的E元素。
E[att="val"]	CSS2	选择具有att属性且属性值等于val的E元素。
E[att~="val"]	CSS2	选择具有att属性且属性值为一用空格分隔的字词列表，其中一个等于val的E元素。
E[att^="val"]	CSS3	选择具有att属性且属性值为以val开头的字符串的E元素。
E[att$="val"]	CSS3	选择具有att属性且属性值为以val结尾的字符串的E元素。
E[att*="val"]	CSS3	选择具有att属性且属性值为包含val的字符串的E元素。
E[att|="val"]	CSS2	选择具有att属性且属性值为以val开头并用连接符"-"分隔的字符串的E元素，如果属性值仅为val，也将被选择。

6.伪元素选择器

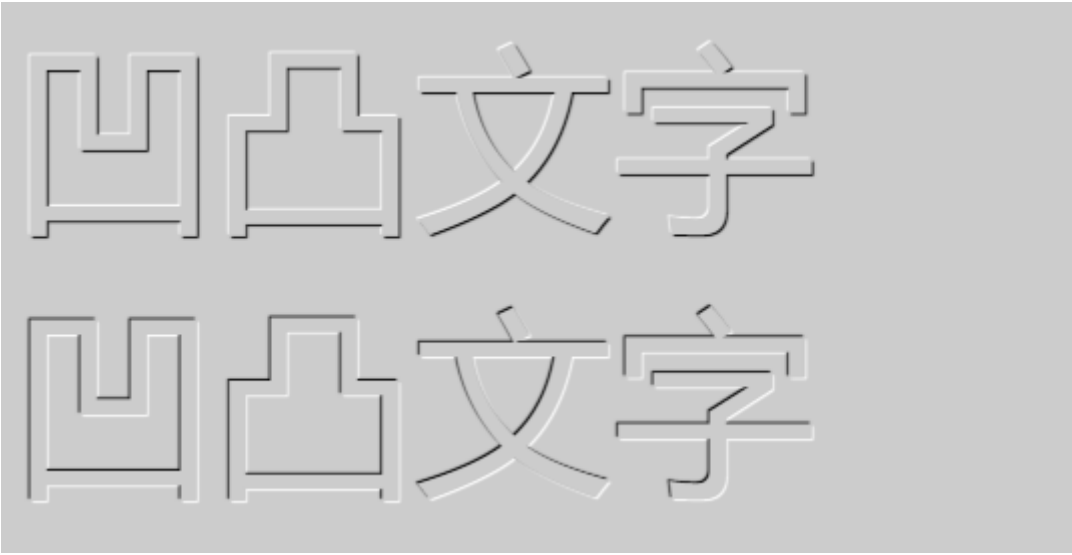
选择符	版本	描述
E:first-letter/E::first-letter	CSS1/3	设置对象内的第一个字符的样式。
E:first-line/E::first-line	CSS1/3	设置对象内的第一行的样式。
E:before/E::before	CSS2/3	设置在对象前（依据对象树的逻辑结构）发生的内容。用来和content属性一起使用
E:after/E::after	CSS2/3	设置在对象后（依据对象树的逻辑结构）发生的内容。用来和content属性一起使用
E::placeholder	CSS3	设置对象文字占位符的样式。
E::selection	CSS3	设置对象被选择时的颜色。

* CSS3将伪对象选择符(Pseudo-Element Selectors)前面的单个冒号(:)修改为双冒号(::)用以区别伪类选择符(Pseudo-Classes Selectors)，但以前的写法仍然有效。

10. 文字凹凸效果

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
    <style>
        body {
            background-color: #ccc;
        }
        div {
            font-family: "微软雅黑";
            font-size: 100px;
            color: #ccc;
        }
        div:first-child {
            text-shadow: 1px 1px 1px #000,-1px -1px 1px #fff;
        }
        div:last-child {
            text-shadow: -1px -1px 1px #000,1px 1px 1px #fff;
        }
    </style>
</head>
<body>
    <div>凸文字</div>
     <div>凹文字</div>
</body>
</html>
```



11.CSS的特性

1.CSS层叠性

样式冲突，遵循就近原则。

样式不冲突，不会层叠

2.CSS继承性

子元素可以继承父元素的样式
（**text,font,line,color**）

继承的权重为**0**

3.CSS优先级

CSS 优先规则**1**： 最近的祖先样式比其他祖先样式优先级高。

CSS 优先规则**2**："直接样式"比"祖先样式"优先级高。

先说说 **CSS 7** 种基础的选择器：

ID 选择器，如 #id{}

类选择器，如 .class{}

属性选择器，如 a[href="segmentfault.com"]{}

伪类选择器，如 :hover{}

伪元素选择器，如 ::before{}

标签选择器，如 span{}

通配选择器，如 *{}

CSS 优先规则**3**：优先级关系：内联样式 **>** **ID** 选择器 **>** 类选择器 **=** 属性选择器 **=** 伪类选择器 **>** 标签选择器 **=** 伪元素选择器

CSS 优先规则**4**：计算选择符中 **ID** 选择器的个数（**a**），计算选择符中类选择器、属性选择器以及伪类选择器的个数之和（**b**），计算选择符中标签选择器和伪元素选择器的个数之和（**c**）。按 **a**、**b**、**c** 的顺序依次比较大小，大的则优先级高，相等则比较下一个。若最后两个的选择符中 **a**、**b**、**c** 都相等，则按照"就近原则"来判断。

CSS 优先规则**5**：属性后插有 **!important** 的属性拥有最高优先级。若同时插有 **!important**，则再利用规则 **3**、**4** 判断优先级。

12 外边距合并问题

1. 使用**marhin**定义块元素的垂直外边距时，两个不嵌套的快元素，可能出现外边距合并

他么之间的外边距距离会是外边距较大的那个

2.两个嵌套的块元素，如果父元素没有内边距及边框，则父元素的上边距会与子元素的上边距合并，合并后外边距较大的。即使父元素的上外边距为**0**，也会合并。（解决：**1.**定义父元素边框，**2.**使用**overflow:hidden**）

13.content 宽度和高度

```
/*外盒尺寸计算（元素空间尺寸）*/
Element空间高度 = content height + padding + border + margin
Element 空间宽度 = content width + padding + border + margin
/*内盒尺寸计算（元素实际大小）*/
Element Height = content height + padding + border （Height为内容高度）
Element Width = content width + padding + border （Width为内容宽度）
```

14 CSS3盒子模型

box-sizing属性

1.值为**content-box** 就是以前的和模型

2.值为**border-box (**再写盒子的**padding**和**border**不会撑大盒子， 宽和高为定义 好的**width**和**height)**

15 盒子阴影

```
box-shadow:水平位置 垂直位置 模糊距离 阴影尺寸(影子大小) 阴影颜色 内/外阴影;
```

值	描述
h-shadow	必需。水平阴影的位置。允许负值。
v-shadow	必需。垂直阴影的位置。允许负值。
blur	可选。模糊距离。
spread	可选。阴影的尺寸。
color	可选。阴影的颜色。请参阅 CSS 颜色值。
inset	可选。将外部阴影 (outset) 改为内部阴影。

16 浮动特性

1.浮动主要目的是为了让多个块级元素按一行显示

浮：加了浮动特性的元素盒子是浮起来的，漂浮在标准盒子之上

漏：加了浮动的盒子不占位置，他原来的位置漏给标准盒子

特：这是特殊使用，使用需谨慎

不管是块元素还是行元素加入浮动后就具有行内块元素的特点

最好和父盒子搭配起来使用

2.*清除浮动的方法**

清除浮动的本质是为了解决父级元素因为子级浮动引起的自己内部高为0的问题

代码：

```
<style>
        .box1 {
            width: 300px;

        }
        .son1 {
            width: 100px;
            height: 100px;
            float: left;
            background-color: pink;
        }
        .son2 {
            width: 100px;
            height: 100px;
            float: right;
            background-color: red;
        }
        .box2 {
            width: 200px;
            height: 200px;
            background-color: blue;
        }
</style>
<body>
    <div class="box1 clearfix">
        <div class="son1"></div>
        <div class="son2"></div>
    </div>
    <div class="box2"></div>
</body>
```

1.在浮动盒子后面添加一个空盒子写上clear:both属性

2.在父级元素里增加属性（**overflow:hidden | auto | scroll**）也可以解决

3.在父级元素里加伪元素

```
.clearfix:after {
    content:".";
    display:block;
    height:0;
    clear:both;
    visibility:hidden;
}
.clearfix {
    *zoom : 1;  /* IE6，7专有*/
}
```

4.使用**before**和**after**双伪元素清除浮动

```
.clearfix:before,.clearfix:after {
    content:"";
    display:table; /*触发BFC，清除浮动*/
}
.clearfix:after {
    clear:both;
}
.clearfix {
    *zoom: 1;
}
```

17 定位属性

position属性**left,right,top,bottom**使用

1.static默认值

2.relative：相对自己原本的位置（原左上角）移动（原来的位置继续占有）（不脱离标准流）

3.absolute：绝对定位完全脱离标准流，不占位置

4.fixed : 和父盒子没关系，只认浏览器当前窗口，脱离标准流，不占位置

父盒子没有定位，孩子盒子绝对定位，孩子会以浏览器为基点

父亲有定位（不管绝对还相对），孩子就以父亲定位

一般父亲相对定位，孩子绝对定位

absolute与**fixed**定位会改变行内元素的特性--->行内块元素

2.定位的盒子距中

子盒子加了定位于父盒子居中 **margin**：左右 **auto** 不管用

1.首先**left**：**50%**

2.再往左走盒子的宽度一半

18.隐藏与显示

display:none (隐藏元素，不是删除，不保留位置**)**

display:block(显示元素与转换与块级元素，**)**

visibility:hidden(隐藏元素，但是保留元素位置**) | visiable(**显示**)**

overflow属性

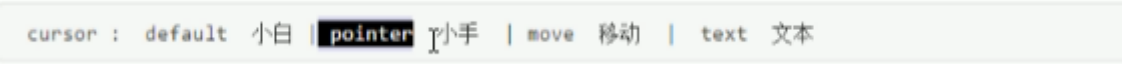
auto： 超出内容才显示滚动条

scroll: 不管超没超出内容，都显示滚动条

hidden：隐藏超出的部分

19.鼠标样式

cursor属性



20.取消轮廓线

outline:0;

21.防止拖拽文本域

```
textarea {
    resize: none;  /* 防止文本域拖拽 */
}
```

22.文字垂直对齐

vertical-align：middle

对块级元素无效，一般用于表单或图片 与文字之间的垂直对齐

23.去除图片底侧缝隙

1. 将行内块元素改为块级元素
2. 加入**vertical-align：top;**

24. word-break（英文）和white-space

25.text-overflow

text-overflow：clip 超出直接省略

text-overflow：ellipsis超出省略号显示

需要搭配**white-space:nawrap** 强制一行显示

还要**overflow:hidden** 直接裁剪

26.伪元素的本质

::before 或 **::after** 本质上是插入一个行内元素

1）伪元素即伪类，它是一个元素的子元素，其意思就是说，我们无法用**JS**获取到这些伪元素，我们无法通过**JS**对其进行增、删、改，所以这也是它们的优点，因为它们不会增加**JS**查询**DOM**的负担，即对于**JS**来说伪元素是透明的。然后因为它们也不是实际的**HTML**标签，所以可以加快浏览器加载**HTML**文件，对**SEO**也有帮助（**SEO**搜索引擎优化）。

2）如果我们把伪类的样式有**absolute**定位的话会把伪类强制变成块级元素，伪类本身是行内元素的。

3）**img**、**input**和其他的单标签是没有**after**和**before**伪元素的，因为单标签本身不能有子元素。

27.过渡属性

在CSS3里使用transition可以实现补间动画（过渡效果），并且当前元素只要有"属性"发生变化时即存在两种状态(我们用A和B代指)，就可以实现平滑的过渡，为了方便演示采用hover切换两种状态，但是并不仅仅局限于hover状态来实现过渡。

语法格式:

```
transition: 要过渡的属性  花费时间  运动曲线  何时开始;
如果有多组属性变化，还是用逗号隔开。
```

select a language

属性	描述	CSS
transition	简写属性，用于在一个属性中设置四个过渡属性。	3
transition-property	规定应用过渡的 CSS 属性的名称。	3
transition-duration	定义过渡效果花费的时间。默认是 0。	3
transition-timing-function	规定过渡效果的时间曲线。默认是 "ease"。	3
transition-delay	规定过渡效果何时开始。默认是 0。	3

前两项属性必须写；花费时间为秒（**s**）；

transition-timing-function 有不同的曲线

过渡属性写在**div**中,不在变换，如**hover**里写

如果需要所有的属性都变，在前面写一个**all**就行

28.transform属性

2D变形

1.translate(x,y):平移属性值 **x:**向**x**轴移动 其值可以为百分号

2.scale(x,y):缩放属性 默认值为 **1** ，》**1** 放大 《**1** 缩小

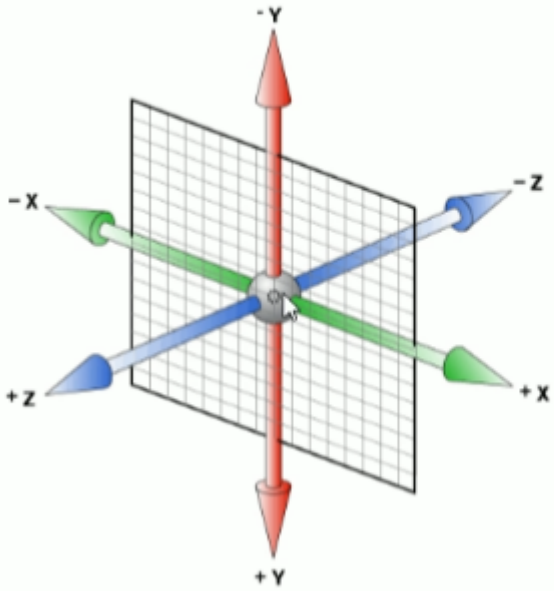
3.rotate(deg【度数】):旋转属性 如 **90deg** 顺时针 **-90deg**逆时针

4.transform-origin: 可以调整元素转换变形的原点 有两个参数 （ **left right bottom top |left right bottom top**）或者（**10px,10px**）

5.skew(deg,deg): 倾斜 一个水平（正直向左倾斜） 二垂直 （正直往上）

3D变形

CSS3中的3D坐标系与上述的3D坐标系是有一定区别的，相当于其绕着X轴旋转了180度，如下图



rotateX(deg【度数】):旋转属性 绕**X**轴旋转

。 。 。 。

透视 （perspective）

透视(perspective)

电脑显示屏是一个2D平面，图像之所以具有立体感（3D效果），其实只是一种视觉呈现，通过透视可以实现此目的。

透视可以将一个2D平面，在转换的过程当中，呈现3D效果。

- 透视原理： 近大远小 。
- 浏览器透视，把近大远小的所有图像，透视在屏幕上。
- perspective，视距，表示视点距离屏幕的长短。视点，用于模拟透视效果时人眼的位置

注，并非任何情况下需要透视效果，根据开发需要进行设置。

perspective 一般作为一个属性，设置给父元素，作用于所有3D转换的子元素

视距：眼睛到屏幕的距离 视距越大效果与越明显

translate3d(x,y,z)

backface-visibility:hidden 不是正面对屏幕就隐藏

29.动画

```
1.定义动画
@keyframes go(动画名称) {
from {
transform:translateX(0);
}
to{
transform:translateX(600px);
}
}
-----------------------------------------------------
2.引用动画
div {
animation:go 2s ease 0s infinite（无限循环） alternate
(动画名称，运动时间，运动曲线，何时开始，播放次数，是否反方向)
}
```

如果有多组动画，用空格隔开就行

30 弹性布局

浏览器窗口变换时可以按比例缩放

父盒子设置**display:flex** 子盒子就可以设置**flex:1 (**份数**)** 按份数比例划分长宽

还可以在父盒子设置最小缩放长款 **min-width**

属性

flex-direction:column 排列方式：垂直排列 默认水平

flex-wrap：**wrap**（换行 ）默认不换行（若子盒子宽之和大于父盒子，会自动压缩子盒子宽以便一行显示）

flex-flow: **(**简写形式**)**

justify-content

作用　用来在存在剩余空间时，设置为间距的方式

属性值	作用
flex-start	默认值。从左到右，挨着行的开头。
flex-end	从右到左，挨着行的结尾。
center	居中显示。
space-between	平均分布在该行上，两边不留间隔空间。
space-around	平均分布在该行上，两边留有一半的间隔空间。

align-items

作用　设置每个flex元素在交叉轴上的默认对齐方式

属性值	作用
flex-start	位于容器的开头。
flex-end	位于容器的结尾
center	居中显示。

align-items 只作用一行

align-content

作用　设置每个flex元素在交叉轴上的默认对齐方式

属性值	作用
flex-start	位于容器的开头。
flex-end	位于容器的结尾。
center	位于容器的中心。
space-between	之间留有空白。
space-around	两端都留有空白。

align-content:center（把所有的当整体居中显示）

其他属性

属性值	作用
flex-basis	设置弹性盒伸缩基准值。
flex-grow	设置弹性盒子的扩展比率。
flex-shrink	设置弹性盒子的缩小比率。
flex	flex-grow、flex-shrink、flex-basis的缩写

flex-basis：设置完基准值 宽度属性不在生效

一个扩大一个缩小

特殊写法

属性	作用
flex: auto;	flex: 1 1 auto
flex: none;	flex: 0 0 auto
flex: 0%; flex: 100px	flex: 1 1 0% flex: 1 1 100px
flex: 1;	flex: 1 1 0%

31.grid布局

二维布局

列的设置

grid-template-columns:100px 100px 100px;
每一列的宽度占**100px** **3**列

grid-template-columns:1fr 1fr 1fr ;
浮动宽度 每一列占**1**份 **3**列 （和弹性布局类似）

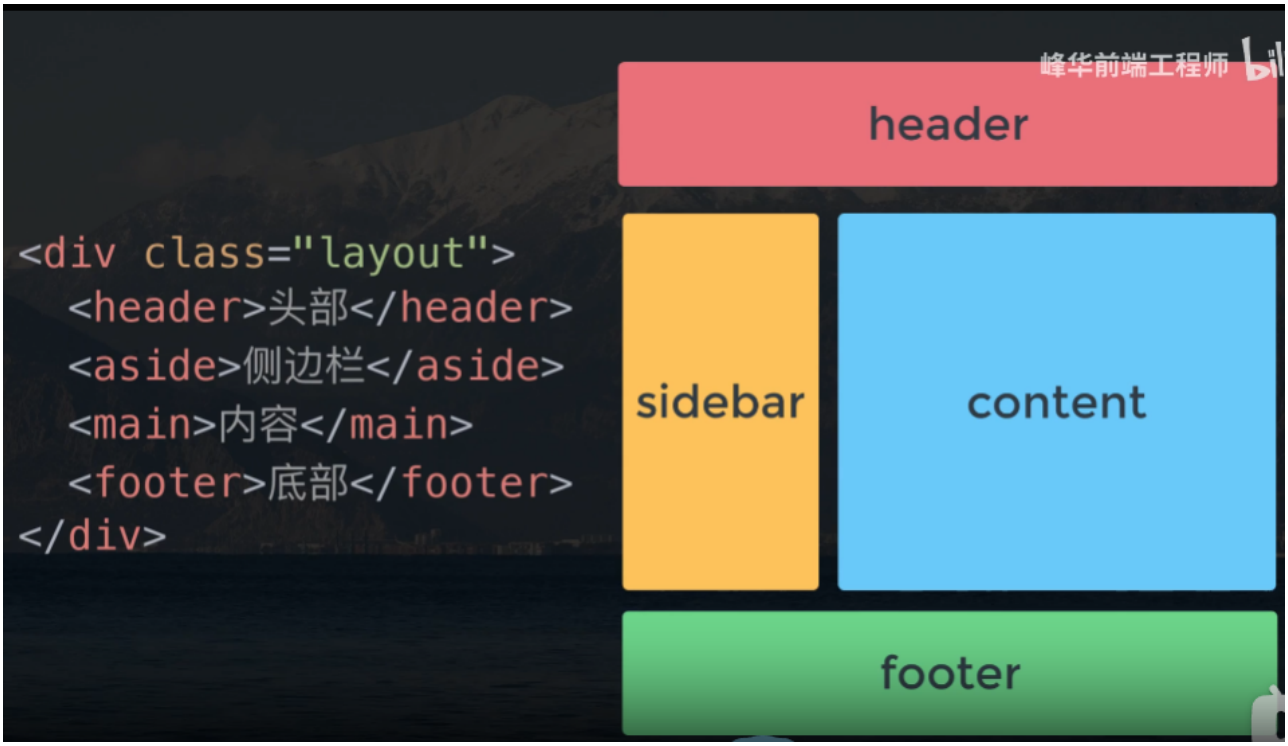
间距

gap 统一设置

column-gap:24px ; 设置列间距

row-gap:24px ; 设置行间距

grid-template-areas 区域设置




```
grid-template-areas:
    "header header header"
    "sidebar content content"
    "footer footer footer";
```

对齐方式与**flex**类似

align-items:center 垂直方向对齐

justify-items:center 水平方向对齐

子元素整体小于父盒子可以整体对齐

align-content:center 垂直方向对齐

justify-content:center 水平方向对齐

CSS3 新特性简介

1.简介

1.强大的**CSS3**的选择器

2.不借助图片的视觉效果

3.盒模型变化（多列布局和弹性盒模型）

4.阴影效果

5.Web字体和**web Font**图标

6.CSS3过渡与动画交互效果

7.媒体查询

渐进增强：满足大部分浏览器

优雅下降：满足大部分用户

简单图形实现

1.三角型

```
 <style>
        .box {
            margin: 50px auto;
            width: 0;
            height: 0;
            border-top: 50px solid red;
            border-right: 50px solid yellow;
            border-bottom: 50px solid green;
            border-left: 50px solid blue;
        }
    </style>

    <body>
    <div class="box"></div>
</body>
```

```
 <style>
        .box {
```

```
            margin: 50px auto;
            width: 0;
            height: 0;

            border: 50px solid transparent;
            border-top: 50px solid red;

             /* border-top: 50px solid red;
            border-right: 50px solid transparent;
            border-bottom: 0 solid transparent;
            border-left: 50px solid transparent; */
        }
    </style>
```