通配符选择器

通配符选择器用`*`表示，就是选中所有的标签，所有选择器作用范围最广，能匹配页面中所有元素。

```
*{
    属性1: 属性值1;
}
```

使用通配符选择器定义*css*样式，清除所有*HTML*标记的默认边距。

```
*{
    margin:0;  //定义外边距
    padding:0; //定义内边距
}
```

css复合选择器

1. 子元素选择器

子元素选择器只能选择作为某元素子元素（最近的一级）的元素。

把父级标签写在前面，子集标签写在后面，中间跟一个 > 连接。

```
.class>h3{
     color:red;
}
```

2. 连接伪类选择器

未访问的链接：*a:link*

已访问的链接：*a:visited*

鼠标移动到链接上：*a:hover*

选定的链接：*a:active*

块级、行内、行内快元素

元素模式	元素排列	设置样式	默认宽度	包含
块级元素	一行只能放一个块级元素	可以设置宽高	容器得*100%*	容器级可以包含任何标签
行内元素	一行可以放多个行内元素	不可以设置宽高	本身内容得宽度	容器文本或其它行内元素
行内块元素	一行可以放多个行内块元素	可以设置宽高	本身内容得宽度	

标签元素转化display

块转行：`display: inline`

行转块：`display: block`

块、行转为行内块：`display: inline-block`

css背景图（background）

1. 背景颜色

```
background-color: 颜色值;
```

2. 背景图片

```
background-image: url();
```

3. 背景平铺（repeat）

```
background-repeat: repeat | no-repeat | repeat-x | repeat-y
```

参数	作用
repeat	背景图片在纵向和横向平铺（默认）
no-repeat	背景图不平埔
repeat-x	背景图在横向平铺
repeat-y	背景图在纵向平铺

4. 背景位置（position）

```
background-position:length || length

background-position:position || position
```

参数	值
length	百分数 / 由浮点数字和单位标识符组成的长度
position	*top* / *center* / *bottom* / *left* / *center* / *right*

5.背景附着（固定）

背景附着就是背景图片是滚动还是固定得。

```
background-attachment: scroll | fixed
```

srcoll：背景图片是随对象内容滚动的。

fixed：背景图片固定。

6. 背景简写

background：背景颜色 背景图片地址 背景平铺 背景滚动 背景位置

7. 背景透明

```
background: rgba(0,0,0,0.3)
```

css三大特性

1. css层叠性

样式冲突：遵循就近原则，那个样式离得近，就执行那个样式。

2. css继承性

子标签会继承父标签的某些元素，如文本颜色和字号（*text-*、*font-*、*line-*这些元素的开头可继承，以及*color*属性）。

想要设置一个可继承性的属性，只需将它应用于父元素即可。

3. css优先级

标签选择器	计算权重公式
继承或者*	*0，0，0，0*
标签选择器	*0，0，0，1*
类、伪类	*0，0，1，0*
*ID*选择器	*0，1，0，0*
行内样式*style=""*	*1，0，0，0*
！*important*	无穷大

！*important* > 行内样式*style=""* > *ID*选择器 > 类、伪类 > 标签选择器 > 继承或者 *

盒子模型

1. 表格的细线边框

两个单元格之间的边框会出现重叠，从而使边框变粗。

通过*css*属性：

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

*border-collapse:collapse*表示相邻边框合并在一起。

2. 内边距padding

值得个数	表达意思
*1*值	*padding*：上下左右内边距
*2*值	*padding*：上下内边距 左右内边距
*3*值	*padding*：上内边距 左右内边距 下内边距
*4*值	*padding*：上 右 下 左

3. padding不影响盒子大小的情况

如果没有给一个盒子指定宽度，给这个盒子指定*padding*，则不会撑开盒子。

4. 块级盒子水平居中

盒子必须指定宽度。

给左右的外边距都设置*auto*。

```css
.header {
    width:960px;
    margin:0 auto;
}
```

常见写法：

- *margin-left*：*auto*；*margin-right*：*auto*；
- *margin*：*auto*；
- *margin*：*0 auto*；

5. 插入图片和背景图片区别

1. 插入图片用的最多，比如产品展示类，移动位置只能靠盒子模型*padding*、*margin*。
2. 背景图片用于小图片背景或者超大图片背景 背景图片只能通过*background-position*。

6. 清除元素默认的内外边距

```css
* {
    margin:0;
    padding:0;
}
```

行内元素为了照顾兼容性，尽量只设置左右内外边距，不要设置上下内外边距。

7.相邻块元素垂直外边距的合并

当上下两个块级元素相遇时，如果上面元素有下外边距*margin-bottom*，下面的元素有上外边距*margin-top*，则他们的垂直间距不是*margin-bottom*与*margin-top*之和。

取两个值中的较大者这种现象被称为相邻元素垂直外边距的合并（也称外边距塌陷）。

解决方法：尽量只给一个盒子添加*margin*。

8. 嵌套块元素垂直外边距合并

对于两个嵌套元素的块元素，如果父元素没有内外边距及边框，父元素的上外边距和子盒子的上外边距会发生合并。

解决方案：

可以为父元素定义上边距 `border: 1px solid transparent` 。

可以给父级指定一个上*padding*值 `padding-top: 1px` 。

为父元素添加 `overflow: hidden` 。

去掉列表默认样式： `list-style:none`

拓展

1. 圆角边框

语法：

```
border-radius:lenght
```

其中每一个值可以为数值或百分比形式。

如果是矩形就用精确单位，只用高度的一半即可。

2. 盒子阴影

语法：

```
box-shadow:水平阴影  垂直阴影  模糊距离（虚实）  阴影尺寸（大小）  阴影颜色  内/外阴影
```

前两个属性必须写，其余的可以省略。

浮动（float）

语法：

```
选择器 { float: 选择器 };
```

属性值	描述
none	默认不浮动
left	左浮动
right	右浮动

设置浮动的盒子，会脱离标准流，漂浮在普通流的上方，不占有位置。

*float*属性会改变元素*display*属性，相当于转变为行内元素，中间是没有空隙。

1. 清除浮动

清除浮动的本质：清除浮动主要为了解决父级元素因为子元素引起内部高度为*0*的问题（父级元素不给高的情况下，由子元素撑开高度），清除浮动之后，父级就会根据浮动的子盒子自动检测高度。父级有了高度就不会影响下面的标准流。

清除浮动的方法：

1、在*css*中，`clear`属性用于清除浮动。

```
选择器{ clear:属性值; }
```

属性值：

- *left*：清除左侧浮动。
- *right*：清除右侧浮动。
- *both*：同时清除左右两侧浮动影响。

在实际开发中，只用*clear*：*both;*

做法是通过在`浮动元素末尾`添加添加一个空的标签。

```
<div class="box">
     <div class="son1"> </div>
     <div class="son2"> </div>
     // 添加一个空的标签
     <div class="clear" style="clear:both"></div>
</div>
<div class="h1"> </div>
```

缺点：添加许多无意义的标签，结构化较差。

2、 父级添加*overflow*属性方法。

```
overflow: hidden | auto | scroll 都可以实现
```

3、 使用*after*伪元素清除浮动

```
clearfix:after {
    content:"";
    display:block;
    height:0;
    clear:both;
    visibility:hiden;
}
clarefix {
    *zoom:1; // ie6、7
}
```

4、 使用双伪元素清除浮动

```
.clearfix:before,
.clearfix:after{
    content:"";
    display:table;
}
.clearfix:after {
    clear:both;
}
.clearfix {
    *zoom:1;
}
```

定位（position）

1.定位模式

值	语义
static	静态定位
relative	相对
absolute	绝对
fixed	固定

相对定位的特点：

- 相对与自己原来在标准流中位置来移动的。
- 原来在标准流的区域继续占有，后面的盒子依然以标准流的方式对待它。

绝对定位的特点：

- 绝对是以带有定位的父元素来移动位置的，如果父元素没有定位，则以浏览器文档为准移动位置。
- 不留有原来的位置，完全是脱标的。

规定定位的特点：

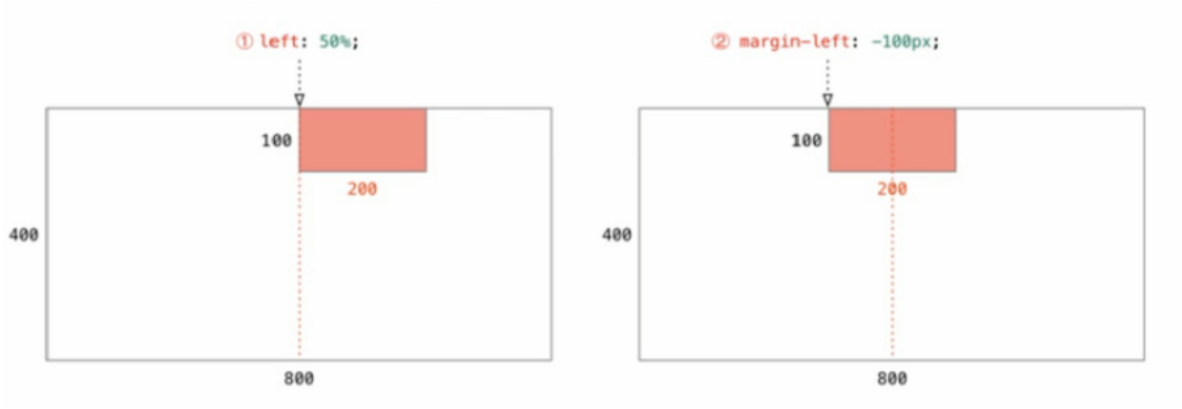
- 完全脱标，只认浏览器的可视区。
- 和父元素没有关系，单独使用，不随滚动条滚动，固定在屏幕上。

2、定位拓展

2.1 绝对定位盒子水平居中

`绝对定位/固定定位的盒子不能通过margin: auto设置水平居中`。

在使用绝对定位时要实现水平居中：



1、`left: 50%` 让盒子的左侧移动到父元素的水平中心位置；

2、`margin-left: -100px` 让盒子向左移动自身宽度的一半；

2.2 堆叠顺序（z-index)

在使用定位布局时，可能会出现盒子重叠的情况。

加了定位的盒子默认后来者居上，后面的盒子会压住前面的盒子。

使用 `z-index` 层叠等级属性可以调整盒子的堆叠顺序。

特性：

1、属性值：整数、负数或*0*，数值越大盒子越靠上，不带单位。

2、如果属性相同，后来者居上。

注：*z-index*只能应用于相对定位、绝对定位和固定定位的元素，其它标准流、浮动和静态定位无效。

2.3 定位改变display属性

*display*是显示模式，可以改变显示模式有一下方法：

- 用*inline-block*转化成行内块。
- 用浮动默认转化为行内块。
- 绝对和固定定位。

一个行内盒子如果加了浮动、固定和绝对定位，不用转换，就可以给这个盒子设置高度和宽度。

注：给盒子添加浮动元素、绝对（固定）定位元素的都不会触发外边距合并问题。

2.4 圆角矩形设置4个角

圆角矩形可以为*4*个角分别设置圆度，按顺序。

```
border-top-left-radius:20px
border-top-right-radius:20px
border-bottom-left-radius:20px
border-bottom-right-radius:20px
```

简写：

```
border-radius:左上角 右上角  右小角 左下角;
```

css高级技巧

1. 元素的显示与隐藏

1.1 display显示

```
display:none;    隐藏
display: block; 显示
```

1.2 visibility可见性

```
visibility: visible;   显示对象
visibility: hidden;    隐藏对象
```

特点：隐藏之后继续保留原来的位置。

1.3 overflow溢出

检索或设置当前对象超过其指定的高度及宽度时如何管理内容。

`visible` 不剪切内容也不添加滚动条。
`hidden` 不显示超过对象尺度的内容，超过的部分隐藏。
`scroll` 不管超出内容否，总是显示滚动条。
`auto` 超出自动显示滚动条，不超出不显示。

鼠标样式cursor

设置或检索在对象上移动的鼠标指针采用何种系统预定义的光标形状。
`default` 小白默认。
`pointer` 小手。
`move` 移动。
`text` 文本。
`not-allowed` 禁止。

轮廓线outline

去掉轮廓线：*outline：0* 或 *outline：none*

防止拖拽文本域resize

`resize: none`

vertical-align 垂直对齐

只针对 `行内元素` 或 `行内块元素`

```
vertical-align: top (顶端对齐) | middle垂直对齐) | bottom
```

去除图片底侧空白缝隙：给*img*添加 `vertical-align: middle | top | bottom` 让图片不要和基线对齐。

或给*img*添加 `display: block` 转化为块级元素就不存在该问题。

溢出文字省略号显示

1. white-space

设置或检索对象内文本显示方式，通常用于强制一行显示内容。

```
white-space:normal; 默认处理方式
white-space:nowrap; 强制在一行内显示所有文本，直到文本结束或遇br标签才换行。
```

2. text-overflow文字溢出

检索或设置是否使用一个省略标记（.....）标示对象内文本的溢出。

```
text-overflow:clip;  不显示省略标记，简单的裁切
text-overflow:ellipsis; 当对象内文字溢出时显示省略标记
```

注意： `一定要首行强制在一行内显示，再和overflow搭配使用`

```
1、先强制一行显示
white-space:nowrap;
2、超出的部分隐藏
overflow: hidden;
3、超出的部分用省略号替代
text-overflow: cllipsis;
```

拓展

1. margin负值

压住盒子相邻的边框

多个盒子浮动在一行，会出现盒子相邻边框出现变粗的效果，可以给盒子：*margin-left*：*-1px*。

2.css三角形

1、用*css*可以模拟三角形效果。

2、宽度高度为*0*。

3、*4*个边框都要写，只保留需要的边框颜色，其余的不能省略，都改为*transparent*透明。

4、照顾兼容低版本浏览器加上*font-size*：*0*；*line-height*：*0*；

css3

1、css3属性选择器

选择符	简介
E[att]	选择具有*att*属性的*E*元素
E[att="val"]	选择具有*att*属性且属性值等于*val*的*E*元素
E[att^="val"]	匹配具有*att*属性、且值以*val*开头的*E*元素 选择类
E[att$="val"]	匹配具有*att*属性、且值以*val*结尾的*E*元素
E[att="val"]*	匹配具有*att*属性、且值中含有*val*的*E*元素

2、结构伪类选择器

`first-child` 选择第一个元素。

`last-child` 选择最后一个元素。

`nth-child(n)` 选择第几个元素。

*n*可以是关键字，*even*（*2n*）是偶数，*odd*（*2n+1*）是奇数

`first-of-type、last-of-type、nth-of-type(n)` 选择指定类型的元素

```
div span:last-of-type {
     color:pink;
}
<div>
    <p></p>
    <span></span>
    <span></span>
    <span></span>
</div>
```

3、伪元素选择器

`::before` 在元素内部的前面插入内容。

`::after` 在元素内部的后部插入内容。

注：

- *before*和*after*必须有 `content` 属性。
- *before*和*after*创建一个元素，是属于行内元素。
- 伪元素和标签选择器一样，权重为*1*。

4、2D转换

1.1 移动translate

语法：

```
transform: translate(x,y)
transform: translateX(n)
transform: translateY(n)
```

- *2D*转换的移动，沿着*x*和*y*轴移动元素。
- 最大的优点：不会影响其他元素的位置。
- 对行内标签没有效果。
- *translate*中的百分比单位是相对于自身元素的*translate*（*50%*，*50%*）。

1.2 旋转rotate

让元素在二维平面内顺时针或逆时针旋转。

```
transform:rotate(度数)
```

- 单位是*deg*。
- 角度为正是顺时针，负为逆时针。
- 默认旋转中心点是元素的中心点。

1.3 转换中心点transform-origin

```
transform-origin: x y;
```

- 默认转换的中心点是元素的中心点（*50% 50%*）。
- 还可以给*x y*设置像素或方位名词（*top bottom left*等）。

1.4 缩放scale

给元素添加该属性可以控制放大还是缩小。

```
transform: scale(x,y)
```

- 注意其中的x和y用逗号分隔
- transform:scale(1,1) ：宽和高都放大一倍，相对于没有放大
- transform:scale(2,2) ：宽和高都放大了2倍
- transform:scale(2) ：只写一个参数，第二个参数则和第一个参数一样，相当于 scale(2,2)
- transform:scale(0.5,0.5)：缩小
- sacle缩放最大的优势：可以设置转换中心点缩放，默认以中心点缩放的，而且不影响其他盒子

5、动画

1.1 动画基本使用

1.使用keyframes定义动画：

```css
@keyframes move {
            /* 开始状态 */
            0% {
                transform: translateX(0px);
            }
            /* 结束状态 */
            100% {
                transform: translateX(1000px);
            }
        }
```

5.2 动画常用属性

属性	描述
@keyframes	规定动画。
animation	所有动画属性的简写属性，除了animation-play-state属性。
animation-name	规定@keyframes动画的名称。（必须的）
animation-duration	规定动画完成一个周期所花费的秒或毫秒，默认是0。（必须的）
animation-timing-function	规定动画的速度曲线，默认是“ease”。
animation-delay	规定动画何时开始，默认是0。
animation-iteration-count	规定动画被播放的次数，默认是1，还有infinite
animation-direction	规定动画是否在下一周期逆向播放，默认是“normal“,alternate逆播放
animation-play-state	规定动画是否正在运行或暂停。默认是"running",还有"paused"。
animation-fill-mode	规定动画结束后状态，保持forwards回到起始backwards

5.3 动画简写属性

animation：动画名称 持续时间 运动曲线 何时开始 播放次数 是否反方向 动画起始或者结束的状态;

```
animation: myfirst 5s linear 2s infinite alternate;
```

- 简写属性里面不包含 animation-play-state
- 暂停动画：animation-play-state: puased; 经常和鼠标经过等其他配合使用
- 想要动画走回来，而不是直接跳回来：animation-direction ： alternate
- 盒子动画结束后，停在结束位置： animation-fill-mode ： forwards

5.4 速度曲线细节

animation-timing-function：规定动画的速度曲线，默认是“ease”

值	描述
linear	动画从头到尾的速度是相同的。匀速
ease	默认。动画以低速开始，然后加快，在结束前变慢。
ease-in	动画以低速开始。
ease-out	动画以低速结束。
ease-in-out	动画以低速开始和结束。
steps()	指定了时间函数中的间隔数量（步长）

*steps()*就是分几步来完成动画，有了*steps*就可以不用写*ease*或者*linear*。

6. 3D转换

1.1 透视perspective

- 透视也称为视距，就是人的眼睛到屏幕的距离。距离视距点越近，在电脑平面成像越大，越远成像越小。
- 透视的单位是像素。

透视写在被观察父盒子上

移动transform3d

transform: translateX(100px);

transform: translateY(100px);

transform: translateZ(100px);

transform: translateX,Y,Z(100px); 分别指要移动的轴的方向的距离。

2.1 3D旋转rotate3d

3d 旋转指可以让元素在三维平面上沿着*x*轴、*y*轴、*z*轴或者自定义轴进行旋转。

transform： *rotateX,Y,Z*（*45deg)*

transform：*rotate3d(x,y,z,deg)* 如：*transform*：*rotate3d(1,0,0,180deg)*；沿着*x*轴旋转*180*度。

2.2 3D呈现transform-style

控制子元素是否开启三维立体环境。

transform-style·：*flat* 子元素不开启*3D*立体空间 默认。

transform-style：*preserve-3d* 子元素开启立体空间。

代码写给父级，但影响的是子盒子。