

# 一、CSS选择器



## 一、常规选择器

- |          |             |
|----------|-------------|
| 1, id选择器 | #name {}    |
| 2, 类选择器  | .name {}    |
| 3, 标签选择器 | .tagName {} |
| 4, 通配符   | * {}        |

## 二、结构选择器（派生选择器）

- |            |                  |              |
|------------|------------------|--------------|
| 1, 后代选择器   | div p {}         |              |
| 2, 子元素选择器  | h2 > span {}     |              |
| 3, 相邻兄弟选择器 | .focus + li {}   | —— 只选择后面的兄弟  |
| 4, 后兄弟选择器  | .current ~ li {} | —— 同级后面的兄弟元素 |



### 三、属性选择器：选择标签中有以下符合要求的属性的元素

- 1, [attr]            属性名为 attr
- 2, [attr='val']    属性为attr, 值为val
- 3, [attr~='val']   属性名为attr, 其中的一个值为val
- 4, [attr^='val']   属性名为attr, 属性值以val为开头
- 5, [attr\$='val']   属性名为attr, 属性值以val结尾
- 6, [attr\*='val']   属性名为attr, 属性值中包含val字符串
- 7, [attr|='val']   属性名为attr, 属性值必须是val或val-开头

### 四、伪类选择器

:link / :visited / :hover / :active / :focus / :checked

### 五、伪元素选择器

:before / :after / ::selection / ::placeholder

参考：<https://www.html.cn/book/css/selectors/pseudo-classes/index.htm>



## 二、浮动



# inline-block/float (浮动)

## inline-block 特性：

- 1、块在一排显示
- 2、内联支持宽高
- 3、默认内容撑开宽度
- 4、标签之间的换行间隙被解析（问题）
- 5、ie6 ie7不支持块属性标签的inline-block（问题）

## float浮动：

- 1、块在一排显示
- 2、内联支持宽高
- 3、默认内容撑开宽度
- 4、脱离文档流



# float/文档流

`float:left | right | none | inherit;`

文档流是文档中可显示对象在排列时所占用的位置。

浮动的定义：使元素脱离文档流，按照指定方向发生移动，遇到父级边界或者相邻的浮动元素停了下来。

`clear:left | right | both | none | inherit`；同级的前面的所有元素不能有向左边（向右边）的浮动属性

`clear:both`；同级的前面的所有元素不允许有任何方向的浮动属性。



# 清除浮动影响的方法

因为元素设置浮动后会脱离文档流，导致父级元素失去高度，所以需要清除浮动对父级元素的影响

- 1.加高（扩展性不好）
- 2.父级浮动（页面中所有元素都加浮动，margin左右自动失效）
- 3.inline-block（margin左右自动失效）
- 4.空标签清浮动
- 5.br清浮动（不符合工作中：结构、样式、行为，三者分离的要求。）
- 6.overflow:hidden（元素需要有宽度）
- 7.after伪类清浮动方法（现在主流方法）
  - .clearFix{\*zoom:1}
  - .clearFix:after{content:"";display:table;clear:both;}



## 三、定位



## relative相对定位/定位偏移量

position:relative; 相对定位

- a、不影响元素本身的特性；
- b、不使元素脱离文档流（元素移动之后原始位置会被保留）；
- c、如果没有定位偏移量，对元素本身没有任何影响
- d、提升层级

定位元素位置控制

top/right/bottom/left 定位元素偏移量。



# absolute绝对定位/定位层级

position:absolute; 绝对定位

- a、使元素完全脱离文档流；
- b、使内嵌支持宽高；
- c、块属性标签内容撑开宽度；
- d、如果有定位父级相对于定位父级发生偏移，没有定位父级相对于document发生偏移；
- e、相对定位一般都是配合绝对定位元素使用；
- f、提升元素层级

## 定位层级

- a、定位元素默认后者层级高于前者；
  - b、不在同一父级下的标签，比较层级是没有意义的。
- z-index:[number]； 强制设置定位层级



## fixed固定定位、定位其他值

`position:fixed` ; 固定定位

与绝对定位的特性基本一致，的差别是始终相对浏览器窗口进行定位；

定位其他值：

`position:static` ; 默认值

`position:inherit` ; 从父元素继承 (不兼容)

`position: sticky`; 粘性定位

`position: relative | absolute | fixed | static | sticky | inherit | initial`



## 四、响应式



根据不同屏幕分辨率使用不同的样式和布局结构

媒体查询

@media 设备名 only ( 选取条件 ) not ( 选取条件 ) and ( 设备选取条件 ) , 设备二{sRules}

```
<link rel= "stylesheet" type= "text/css"  
media= "only screen and ( max-width :  
480px ) " href= "link.css" />
```



## 五、动画与3D



# Transition 过渡动画

## 五个属性

transition-property css样式属性名称  
transition-duration 动画持续时间（需要单位s）  
transition-timing-function 动画效果函数名称  
linear、ease、ease-in、ease-out、  
ease-in-out、cubic-bezier(n,n,n,n)  
transition-delay 延迟执行动画的时间  
animation-play-state: paused; 暂停动画执行

## 简写

transition: 样式 持续时间 动画效果 延迟时间;  
transition: 持续时间; (all 时间 ease 0)



# @keyframes 关键帧动画

## 七个属性

animation-name 帧动画名称

animation-duration 动画持续时间 (需要单位 s)

animation-timing-function 动画效果函数名称

linear、ease、ease-in、ease-out、

ease-in-out、cubic-bezier(n,n,n,n)

animation-delay 延迟执行动画的时间

animation-iteration-count 动画播放次数

n 次数 infinite 无限次

animation-direction 是否轮流播放

normal 正常播放 alternate 来回轮流播放

animation-fill-mode: none | forwards | backwards | both | initial | inherit;

动画完成时的状态

@keyframes 帧动画名称 { selector { styles } }

selector from to / %

## 简写

animation: name time ease delay count direction fill-mode;



# transform之2D变换

偏移 ( left top )

translate(x,y) translateX() translateY()

缩放 (width height )

scale(x,y) scaleX() scaleY()

旋转

rotate(角度)

倾斜

skew(x角度,y角度) skewX(角度) skewY(角度)



# transform之3D变换

每个属性都比2D多了一个Z轴，需要设置 transform-style: preserve-3d; 才能看到3D视角效果，否则只能看到平面效果

偏移 ( left top 景深)

translate3d(x,y,z,angle) translateX() translateY() translateZ()

缩放 (width height 景深)

scale3d(x,y,z) scaleX() scaleY() scaleZ()

旋转

rotate3d(x,y,z) rotateX() rotateY() rotateZ()

注：倾斜没有Z轴

transform-origin 改变被变换元素的转换参考位置



# 3D透视

## `perspective` 透视距离

当给一个元素设置透视属性时，其子元素会获得透视效果，通过设置不同的值，可以改变观察子元素的距离(z轴方向移动)

## `perspective-origin` 透视位置

需要先使用 `perspective`，可以改变在观察子元素时，观察点的X，Y方向的位置

## `backface-visibility:hidden`; 设置背面不可见

设置之后，当旋转180度查看元素背面时，将看不到正面的透视内容（如果不设置默认正面内容可以被透视）