# h5 css3

* **<audio autoplay=”autoplay”** **controls=”controls” loop=”loop”>** **//** **音乐标签**

<source src=”yinyue.mp3” /> // 音乐格式的多种引入法

<source src=”yinyue.ogg” />

<source src=”yinyue.wav” />

</audio>

属性： **autoplay=”autoplay”** // 自动播放

**controls=”controls”** // 控制音频按钮

**loop=”loop”** // 循环播放

* **<video autoplay=”autoplay” controls=”controls” loop=”loop”> // 视频标签**

<source src=”shipin.mp4” /> // 视频格式的多种引入法

<source src=”shipin.ogg” />

<source src=”shipin.webm” />

</video>

* <header></header> // 头部标签
* <nav></nav> // 导航标签
* <section class=”containt”></section> // 块区域标签 可以当div用
* <article></article> // 文章
* <aside></aside> // 侧边栏
* <figure></figure> // 独立的流内容
* <figcaption></figcaption> // 独立流内容的标题
* <fieldset><fieldset> // 表单区域
* <legend><legend> // 表单区域标题
* <footer></footer> // 尾部标签
* ie9以下认识h5标签 （引入一个只有ie9以下认识的脚本）

**<!--[if lte IE 8]> <script src=”html5shiv.min.js”></script><![endif]--> (优盘里有)**

* **H5表单 新增属性**

1. type=” ” // 表单类型

**url** // 输入网址

**email** // 输入邮箱

**tel** // 电话号码

**number** // 数字类型

例：<input type=”number” max=”100” min=”1” step=”5”/>

max//最大值 min//最小值 step//步长

**range** // 滑动块

例：<input type=”range” max=”100” min=”1” step=”5”/>

max//最大值 min//最小值 step//步长

**color** // 拾色器

**time** // 时间 （小时 与 分钟）

**date** // 时间 （年、月、日）默认时间格式 value=”2017-04-12”

**datetime-local** // 时间 （年、月、日、时、分）

**month**  // 时间 （年、月）

**week** // 时间 （年、周(某年的第几周)）

**search**  // 带有语义 下拉菜单 通常配合datalist用

**file** // 选择文件 **属性multiple 可以实现多选 多选时按住ctrl键**

<input type=”file” **multiple=”multiple”**>

1. **placeholder**=”提示信息” // 表单里的提示信息
2. **required**=”required” // 用户必须填写内容
3. **pattern**=”\D+” // 判断正则
4. **autocomplete**=”” // 是否提示历史输入内容

两个值 ：no 不提示 off 提示

1. **autofocus=autofocus** // 自动获取光标
2. **navalidate**  // 取消所有表单验证 （放在form里面）

* **H5新增 表单元素**

1. **<datalist></datalist>** // 数据列表 （用ID名串联起来用）

例： <input text=”text” list=”text”>  
 <datalist id=”text”>

<option>内容一<optiton>

<option>内容二<option>

</datalist>

1. **<meter ></meter>** // 度量器（进度挑）

<meter max=”100” low=”60” high=”80”></meter>

max//最大值 low//最低值 low//中间值

1. **<progress></progress>** // 进度条

<progress max=”100” value=”50”></progress>

1. **<keygen />** // 生成加密字符串
2. **<output>< /output>**  // 输出结果

# Css3

* 选择器

1. 属性选择器
2. **[href]**{} // 选择了所有带有href属性的标签
3. **[href=”pdf.pdf”]**{} // 选择了所有属性href 值为pdf.pdf的标签
4. **[href^=”xls”]**{} // 选择属性为href 属性值前三位为xls的标签
5. **[href$=”ppt”]**{} // 选择属性为href属性值后三位为ppt的标签
6. **[href\*=”r.r”]**{} // 选择属性为href属性值里包括r.r的标签
7. 伪类选择器
   * 1. 位置伪类选择器
8. li**:first-child**{} // 选择父元素下的第一个子元素 且元素必须是li
9. li**:last-child**{} // 选择父元素下的最后一个子元素 且元素必须是li
10. li**:nth-child(10)**{} // 选择父元素下的第10个子元素 且元素必须是li
11. li**:nth-child(2n)**{} // 选择父元素下的个数为偶数的子元素 且元素必须是li
12. li**:nth-child(even)**{} // 选择父元素下的个数为偶数的子元素 且元素必须是li
13. li**:nth-child(2n+1)**{} // 选择父元素下的个数为奇数的子元素 且元素必须是li
14. li**:nth-child(odd)**{} // 选择父元素下的个数为奇数的子元素 且元素必须是li
15. li**:nth-child(7n)**{} // 选择父元素下的个数为七的倍数的子元素 且元素必须是li
16. li**:nth-last-child(1)**{} // 选择父元素下的倒数个数为1的子元素 且元素必须是li
17. li**:nth-child(-1n+3)**{} // 选择父元素下的个数为前三位的子元素 且元素必须是li
    * 1. 空伪类
18. **span:empty**{} // 空伪类 标签里没有内容就会执行这个伪类选择器的内容
    * 1. 排除伪类
19. **li:not**(:last-child) // 选择li元素且排除最后一个li元素
    * 1. 目标伪类
20. **h2:target**{} // 结合锚点进行使用，处于当前锚点的元素会被选中；
21. 伪元素选择器
22. div**:before**{} **// 在元素内部最前面添加东西** 必须要有content:’’; 属性
23. div**:after**{} **// 在元素内部最后面添加东西** 必须要有content:’’; 属性
24. p**::first-letter**{} // 选中p元素下的第一个文字
25. p**::first-line**{} // 选中p元素下的第一行文字
26. **p::selection**{} // 可以改变选中文本的样式

## 属性

* **box-sizing:border-box/content-box;** // 怪异盒模型 两个值

**值一 border-box ：content = width-padding-border** (常用)

值二 content-box : content = width;

* **text-shadow:****-**1px -1px 1px red,1px 1px 1px pink**;**  // 文本阴影 （可以写多个值 有立体效果）
* **box-shadow:** **-**1px -1px 1px red,1px 1px 1px red; // 盒子阴影 （可以写多个值 有立体效果）

值1：阴影水平偏移 正值向右移 负值向左移

值2：阴影垂直偏移 正值向下移 负值向上移

值3：阴影模糊度

值4：阴影的颜色

* **box-shadow:inset;**  // 内阴 影
* **border-radius:** 10px 10px 10px 10px ; // 边框圆角 四个值对应 左上 右上 右下 左下

（可以直接写数值 也可以写%比 可以写一个值 可写多值 可以写8个值 前四个对应四个角横轴半径 后对应纵轴半径）

* border-image:

1. border-image-source: url(图片路径)； // 边框图片资源
2. border-image- slice: 33 33 33 33; // 裁切图片 图片的宽度 可以写一个值
3. border -image-repeat :stretch/repeat/round; // 是是否平铺；值1 默认值 值2平铺 值3 完整显示
4. border-image-width: 10px 20px 30px 40px; // 边框图片宽度/大小
5. **border-image:url(图片) 33 round;**  综合写法

* **background-size:** 20px 20px; // 背景图片大小 值可以写多种

1. 20px 20px; // 直接写数值 宽20px 高20px
2. 100% 100%; // 写百分比 是依据盒子的宽高写的百分比
3. 100%/20px auto; // 前面依据盒子的宽写百分比或数值 后边写auto 是依据背景图片的宽的变化 同比例变化
4. auto 100%/20px; // 前面宽auto 是根据后面的高度同比例变化的
5. **cover;** // 直接铺满盒子 但是可能显示不全
6. **contain;**  // 背景图片全部在盒子内显示

* **background-origin**: // 背景原点 设置背景图片原点位置

1. padding-box; // 默认值 背景图片显示从盒子内边距开始
2. border-box; // 背景图片从盒子的边框区域开始
3. content-box; // 背景图片从盒子的内容区域开始

* **background-clip**: // 背景区域裁切

1. border-box; // 默认值 边框以外裁切掉
2. padding-box; // 内边距以外裁切掉
3. content-box; // 内容以外裁切掉

* **background:url(),url(),url()**; // 同一个盒子写多个背景图片 （写多背景图片的时候不能写背景颜色）

1. **url(1.jpg) no-repeat center center,url(2.jpg) no-repeat left top** // 多个背景图片写法

* background-image // 渐变

1. **线性渐变**

**background-image:linear-gradient(**  // 线性渐变格式

to right,/0deg, // 方向 deg//角度 0deg==0°==从下向上

red 50%, // 开始颜色 50%// 从盒子的一半的地方开始渐变

pink, // 过度颜色

hotpink // 结束颜色

);

如果没有设置background-size百分比是按照盒子的宽度计算的 设置了就是按照background-size计算的

1. **径向渐变**

**background-image:radial-gradient**(

100px 200px **at** center/20px 10px, // 100px//是X半径 200px//是Y半径

center//是盒子中心 20px//盒子中X的位置 10px//盒子中的Y的位置

red 20%, // 开始色 20% 是以半径为参照设置的

purple, // 过度色

pink // 结束色

)

* **transition // 过渡**

1. 帧动画 （图片移动 利用js）
2. **补间动画**
   * 1. transition-property: all; // all 过渡所有的属性
     2. transition-duration:2s; // 过渡的时间
     3. transition-timing-function:ease/ease-in/ease-out/ease-in-out/linear; // 动画执行的速度
     4. transition-delay:2s; // 两秒后执行 延时执行
     5. **综合写法： transtion:** 过渡的属性过渡的时间延时的时间过渡的速度**;**

* **自定义动画**

**语法： @keyframes 动画名称 { / 案例：@keyframes donggua{**

**from{ 开始的状态 } / from{width:100px;height:100px;}**

**to{ 结束的状态 } / to{width:300px;height:300px;}**

**} / }**

**animation:;** 调用动画 属性有8个

1. **animation-name :;** //动画名称
2. **animation-duration:;** // 动画时间
3. **animation-iteration-count:infinite;** // 执行次数可以写不带单位的数字 **（infinite//动画一直执行）**
4. **animation-timing-function:;** // 执行动画的运动速度 （有很多值）
   * 1. **linear** //匀速
     2. **steps(60)** // 可以写值 每秒都有顿挫感
     3. altemate //
5. **animation-delay:2s;** // 延时执行动画
6. **animation-direction:alternate** // 设置动画是否反向运动
7. **animation-fill-mode:forwards** // 动画结束时的位置或状态 **forwards//动画结束时不会返回原点**
8. **animation-play-state:****paused;** // 动画暂停
9. **综合写法 animation: 名称 动画时间 延时时间 执行次数 动画的运动速度。。。;**

* **2D转化** **transfrom**

1. **位移**

**transform:translate( x,y );**

1. **旋转**

**transform:rotate(deg);** // deg角度 可以设置正值 也可设置负值

**transform-origin: center top;/20px 30px;** // 旋转需要旋转中心 写一个值就是横坐标

// 默认盒子的中心 可写 关键字 还可以直接写数值 值1：为横向坐标 值2：为竖向坐标

1. **缩放**

**transform:scale(1);**  // 写数值不加单位 意思倍数 整数是放大 小数是缩小

transforn-origin:center top;/20px 30px; // 依据什么位置缩小 值1：横向 值2：纵向

1. **倾斜**

**transform:skew(deg);** // 可写两个值 第一个值代表竖值 第二个代表水平值

e) **综合写法**

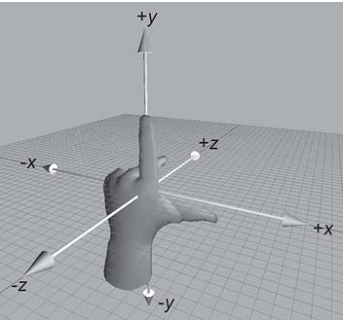
**transform:translate(x,y) rotate(360deg) scale(1.5) skew(10deg);**

* **3D 转换**

1. **perspective:1000px;** // 透视 取值为眼睛距离图片的距离 600-1000px 是人眼最舒服的距离

设置有两种方式 1. 给父元素设置 2.**作为transform的属性写进transform里面**

1. **transform-style:preserve-3d;** // 显示出3D效果
2. 3D位移

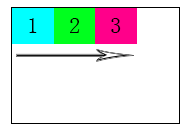
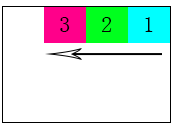
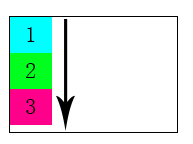
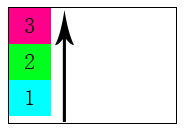


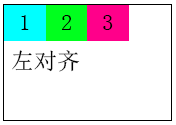
1. **transform:translateX(300px)** // 沿着x轴移动
2. **transform:translateY(300px)** // 沿着y轴移动
3. **transform:translateZ(300px)** // 沿着z轴移动
4. 3D 旋转
   * + 1. transform-origin:top/bottom/center; // 沿着哪里旋转
       2. transform:rotateX(10deg) // 沿着x轴旋转
       3. transform:rotateY(10deg) // 沿着y轴旋转
       4. transform:rotateZ(10deg) // 沿着z轴旋转
5. 3D 缩放
   * + 1. transform-origin:top/bottom/center; // 沿着哪里缩放
       2. transform:scaleX(10deg) // 沿着x轴缩放
       3. transform:rscaleY(10deg) // 沿着y轴缩放
       4. transform:scaleZ(10deg) // 沿着z轴缩放

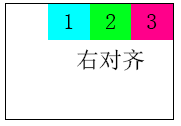
* **伸缩布局**

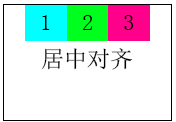
1. **display:flex;** // 给直接父元素设置成伸缩盒子 （只有伸缩盒子才有主轴和侧轴）

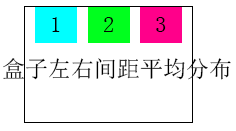
默认主轴从左向右 侧轴始终垂直主轴

1. **flex-direction:**row(默认值); //调整主轴方向
2. **row** // 默认值 左--到--右； 
3. **row-reverse** // 右--到--左
4. **column** // 上--到--下
5. **column-reverse** // 下--到--上
6. **justify-content:flex-start** // 调整子元素在主轴上的对齐方式

flex-start // 默认值 左对齐

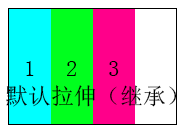
flex-end // 右对齐

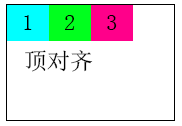
center // 居中对齐

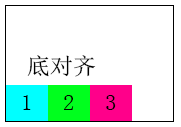
space-around // 左右间距平均分布

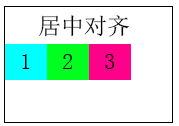
space-between // 每两个子盒子中间的间距平局分布 

1. align-items:; // 调整子元素在侧轴上的对齐方式

stretch // 默认拉伸 继承父盒子高度

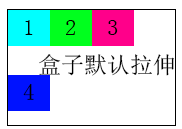
flex-start // 顶部对齐

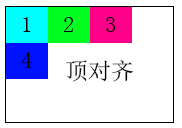
flex-end // 底部对齐

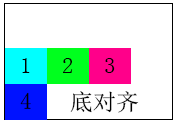
center // 中部对齐

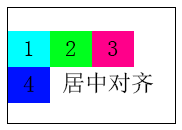
baseline // 基线对齐

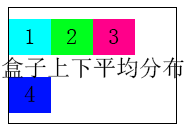
1. flex-wrap:nowrap(默认不换行) ////wrap(换行) //设置子元素是否换行
2. align-content: ; //设置元素换行后的对齐方式（该属性设置的是换行后在侧轴上的对齐方式）

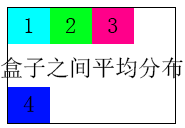
stretch // 默认拉伸

flex-start // 顶部对齐

flex-end // 底部对齐

center // 居中对齐

space-around // 盒子上下间距平均分布

space-between // 盒子之间间距平均分布

1. align-self:; // 该属性是给单个元素设置在侧轴的对其方式 （子元素的属性）

stretch // 默认拉伸

flex-start // 顶部对齐

flex-end // 底部对齐

center // 居中对齐

1. order // 元素排序 （值越大越靠后）（子元素的属性）
2. flex // 子元素在父元素中的缩放比例 （子元素的属性）

* 多列布局 （详情 优盘h5c3特效）

1. column-count:3; // 设置最多多少列 值是没有单位的数字（父盒子属性）
2. column-width:400px; // 设置列的宽度 单位px（父盒子属性）
3. column-gap:20px; // 列与列之间的距离 单位px（父盒子属性）
4. column-rule:1px solid red; // 设置列与列之间的边（父盒子属性）
5. column-span:all; // 跨几列 all：跨所有列 （子盒子属性）