js做的动画性能非常差，所以动画都用CSS3或CSS3的库。

主要内容：选择器（属性、伪类、伪元素）、box/text-shadow、border-radius、box-sizing、多重背景、渐变、过渡、动画、伸缩布局、iconfont

### 属性选择器

其特点是通过属性来选择元素，具体有以下5种形式：

1. E[attr] 表示存在attr属性即可；
2. E[attr=val] 表示属性值完全等于val；
3. E[attr\*=val] 表示的属性值里包含val字符并且在“任意”位置；
4. E[attr^=val] 表示的属性值里包含val字符并且在“开始”位置；
5. E[attr$=val] 表示的属性值里包含val字符并且在“结束”位置；
6. E[attr|="val"] 选择具有att属性且属性值为以val开头并用连接符"-"分隔(<input attr = “val-xxx”>)的E元素，如果属性值仅为val，也将被选择。

### 伪类选择器（冒号后面不能有空格，否则选不中）

除了以前学过的:link、:active、:visited、:hover锚伪类

<style>

.wrap div:first-child span{

display: block;

width: 60px;

background-color: green;

}

.sec:hover span {

width: 30px;

border: 1px solid red;

}

</style>

</head>

<body>

<section class="wrap">

<div class="sec">

<span>abc</span>

</div>

<div class="sec">

<span></span>

</div>

</section>

</body>

Q：当hover的时候为什么border可以变红但是width不会变成30px？

A：因为伪类选择器相当于hover发生时动态地给元素加了个类名，然后让元素或其选定的子元素改变样式，如果权重不够的话，这个样式自然无法覆盖前面的样式。

1、以某元素相对于其父元素或兄弟元素的位置来获取无素的结构伪类。

E:first-child第一个子元素

E:last-child最后一个子元素

E:nth-child(n) 第n个子元素，计算方法是E元素的全部兄弟元素；

E:nth-last-child(n) 同E:nth-child(n) 相似，只是倒着计算；

n遵循线性变化，其取值0、1、2、3、4、... 当n<0时，选取无效。

n可是多种形式：nth-child(2n+0)、nth-child(2n+1)、nth-child(-1n+3)等；

需要满足y=ax+b. 用这种类选择器的时候**必须有父元素**（body不算），否则无效，因为它是先找到该标签的父级，然后再数孩子。

注：指E元素的父元素，对应位置的子元素必须是E

E:empty 选中没有任何子节点的E元素；（使用不是非常广泛）

E:validity 通过验证的元素，E:invalidity 未通过验证的元素；

E:checked 被选中的表单元素

E:target 结合锚点进行使用，处于当前锚点的元素会被选中；

E:last-of-type  
 E:first-of-type  
 E:nth-of-type

日历案例：选中最后三个nth-last-child(-n+3)

CSS3中的选择器在JQ中也可以使用，例如：

$("#out > div:nth-of-type(3)"); //冒号**前后**都不能有空格，否则选不中

|  |  |
| --- | --- |
| li + li 选中li之后的**一个**li | li ~ li 选中li之后**所有**同级的li |

兄弟选择器

### Pseudo-elements:

|  |  |  |  |
| --- | --- | --- | --- |
| ::after | ::first-letter | ::selection | ::placeholder兼容性奇差 |
| ::before | ::first-line | ::backdrop |  |

伪元素能够在不添加其他标签的情况下新增一个元素。

::before / ::after 默认是**行内**元素，双标签也只支持这两个元素，必须有content。如果要设置宽高，要改为块级元素。伪元素选择器内部无法再添加后代元素。  
伪元素选择器只能够添加给双标签元素。  
before 跟 after 只是名字不同而已，用法一样。

### 颜色

hsl（黄鼠狼）：色相hue(颜色)、饱和度saturation、明度lightness、透明度。hue的最大值是359，到360之后重新从红色开始循环。

opacity属性会被子元素继承，但是用rgba/hsla设置的透明度**不会**继承。

### 阴影

text-shadow: 2px 2px 10px red; /\*shadow不影响布局\*/

box-shadow: inset 1px 1px 10px 2px black;

1px 1px 指的是X,Y轴的偏移量，10px指的是模糊程度（颜色从深到浅）。2px指的是阴影外延值，允许为负。先渲染2px，再在2px外渲染10px的模糊区。

### box

box-sizing 如果不设置，默认的值就是content-box  
content-box: 优先保证内容的大小，对盒子进行缩放;  
border-box: 让盒子优先保证自己所占区域的大小，对内容进行压缩。

### 多重背景

background-image:

background-size: 50px 100px / contain / cover 等比例拉伸，多出来的被隐藏

background-position:

background: url() no-repeat center top **/**cover;

background: url('images/dog.gif') no-repeat 115px 70px **/** 370px 280px; 前面是图片位置，后面是图片的尺寸;

background-origin: border-box / padding-box / content-box; 从边框 / 内边距的最外缘或是内容区开始放置背景图  
background-clip: border-box / padding-box / content-box; 从...开始裁剪

border-radius: 100px 0 0 0 / 50px 0 0 0; 前面的表示椭圆的横轴，后面的表示纵轴。

div{  
 width: 200px;  
 height: 100px;

background-color: red;  
 border-radius: 100px / 50px;

border-radius: 50% / 50%;  
 border-radius: 50%;  
 }

### 渐变

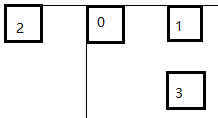
background-image: linear-gradient(to right, pink 10%, skyblue, purple);

radial-gradient

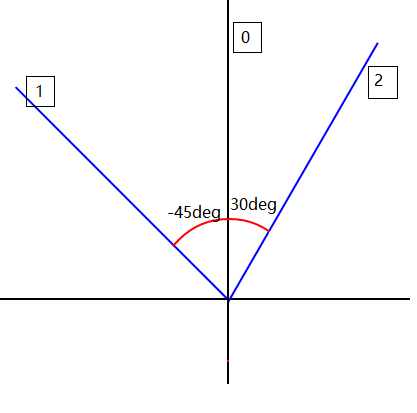
### 过渡

过渡可以为一个元素在不同状态之间切换的时候定义不同的过渡效果。

transition: all 1s; 设在div上和设在div:hover上效果不同。前者在鼠标悬停或离开时都有过度效果，后者只在悬停时有过度，离开瞬间恢复原状。

transform: translateX( ); / transform: translateY( ); 括号里的数值是相对于**初始位置**（元素左上角作为原点）的偏移量，而不是相对于现在的位置的偏移量。比如：初始位置是0，先设transform: translateX(50px); 再设transform: translateX(-50px); 元素将跑到2的位置，3的偏移量是transform: translate(50px，50px );

transform: translateZ( ); 是相对于现在的位置的Z轴，所以写的位置很重要，6张图片需要先绕Y旋转，然后再向外走，如果先朝Z走再绕Y旋转，那么这些图片将挤在一起。

transform: rotate( \_deg); 是根据**初始位置**0的旋转角度。比如一条线，想把它从1（transform: rotate(-45deg)）移到2，写上transform: rotate( 30deg)即可。

## 3D变换：http://www.zhangxinxu.com/wordpress/2012/09/css3-3d-transform-perspective-animate-transition/

translateX (50%); 指的是**自身**宽度的一半。

|  |  |
| --- | --- |
| transition过渡 | transform变形 |
| transition-property: width, height; | transform: translate(x, y); |
| transition-duration: .3s; | transform: scale(.5 2); 缩放到原来的x倍  transform: scaleZ() 单独设置没有效果 |
| transition-delay: 1s, 2s; 表示宽度延迟1s，高度延迟2s | transform: rotate(deg);  transform: rotate3d(1,2,3, 30deg); |
| transition-timing-function: linear; | transform: skew(deg, deg); |
| transition: width 1s .3s ease; 复合写法 | transform - style: preserve - 3d; |
|  | transform - origin: left, top; |

transform - origin属性不能设置在:hover上，只能设置在元素自身上。

transform-style: preserve - 3d; 加到父元素上

perspective: 500px; 展现“近大远小”的透视效果，但只能看到一个面，可以设在父盒子上，也可以设在子盒子上，与transform的其他属性写在一起。

backface-visibility: hidden; 背面看不到

**\***, ::before, ::after { }; \*无法匹配到伪元素生成的标签

The **transitionend** event is fired when a CSS transition has completed. In the case where a transition is removed before completion, such as if the transition-property is removed or display is set to “none”, then the event will not be generated.

* 当页面最小化时，transitionEnd 会停止执行
* transform: translate( )属性也会触发transitionEnd

dom.addEventListener(**'webkitTransitionEnd'**,**function**(e){ });

dom.addEventListener(**'transitionEnd'**,**function**(e){ });

## CSS3 transitions and animations allow you to animate [a specific set of CSS properties](http://www.w3.org/TR/css3-transitions/#animatable-properties). Two of the properties that cannot be animated are the [display](http://cssvalues.com/#display) and visibility properties. display和visibility 属性在加上transition: all 1s; 之后是无效的。

## Animation

基础语法：@keyframes

兼容性写法（属性的末尾 tab键自动生成）：-webkit-transition: all 1s;

|  |  |  |
| --- | --- | --- |
| webkit (谷歌,Safari) | o opera浏览器 | ms 微软的ie和火狐 |

animation: toright 2s 1s linear steps(60s) infinite ;

|  |  |
| --- | --- |
| animation-name: toright; | animation-duration: 2s; |
| animation-delay: 1s; | animation-timing-function: linear; |
| animation-iteration-count: infinite; |  |
| animation-play-state: paused;  在js中的写法：dom.style.animationPlayState =“paused”; | |

复合写法的几个属性位置可以随意变动，但是时间的顺序不能改变，第一个是duration，第二个是delay。

## ICONFONT

|  |  |
| --- | --- |
|  | 使用方法 |
| bootstrap | 给空的<span></span> 标签加上类名 |
| 阿里巴巴图标库 | fontclass 最简单 |
| symbol |
| unicode |

## 多列布局

-webkit-column-count: 4; 共有4列

-webkit-column-width: 400px; 列宽

-webkit-column-gap: 40px; 列间距

-webkit-column-rule: 1px dashed grey; 列间分隔线

-webkit-column-span: all;

## 整屏滚动

<script type="text/javascript">  
 $(document).ready(function () {  
 $(".demo").fullpage({  
 afterLoad: function (anchorLink, index) {

}  
 });  
 });  
</script>

整屏滚动的div的类名必须是.section并且会在这个div下面自动生成一个新的div，所以原来div的子级现在变成了孙级。

paddingTop： 内容与顶部的距离

afterLoad：滚动到某一屏后的回调函数，接收 anchorLink 和 index 两个参数，anchorLink 是锚链接的名称，index 是序号，从1开始计算

示例：一滚动就打印出字符串

滚动打印出索引

滚动打印出当前页改变背景颜色

**过渡和动画**

这篇文章详述了动画和过渡的相似和不同，以便大家在需要的时候选用合适的属性。

**相似性**

1. 指明哪个CSS属性会被改变

2. 用函数来控制属性值改变的速率

3. 指明过渡/动画所需要的时长

4. 可以侦听过渡/动画事件，然后进行需要的操作

5. 属性的改变是可视的，非常直观

**差异性**

1. 触发

- 过渡：必然是由CSS属性值的改变而触发的，触发过渡有以下几种情况：

     - 用 :hover 伪类改变属性值

     - 用 JS 添加或删除类名

     - 用 JS 设置一个行内样式，样式里修改了某个属性值

- 动画：不需要触发，一旦定义完成就会自动展示

2. 循环

- 过渡：触发一次只能执行一次。当然也可以利用`transitionEnd`事件来多次触发，但是操作就不方便了

- 动画：可以用 `animation-iteration-count`属性来规定循环播放的次数

3. 复杂度

- 过渡：只是简单地从起点到终点，中间的过程无法控制

- 动画：在开始和结束之间可以定义多个关键帧，实现类似 flash 的效果

4. 正式程度

- 过渡：浏览器**\*\*只\*\***监听在`transition`中有规定的属性。简便起见，通常会简写为`transition: all 0.5s ease;`，但是不推荐这样写，因为这样浏览器需要监听所有可能变化的属性

- 动画：不需要在每一帧都列出所有变化的属性，即使属性只在终点（100% { } 那一帧）出现了一次，也可以平滑过渡

5. 和 JS 联用

- 过渡：不关心属性是通过什么途径改变的，只要属性值改变，过渡就会发生，所以用 JS 来**\*\*改变\*\***属性十分可行

- 动画：用 JS 来**\*\*改变\*\***关键帧十分复杂

**总结**

- 如果效果复杂且需要循环就选择动画，简单的需求就选择过渡

- 如果需要用 JS 计算某个属性值，就选择过渡