

## 1. 浏览器前缀

- css3 兼容不同的浏览器，针对旧的浏览器做兼容，新浏览器基本不需要添加前缀

浏览器	内核	前缀
IE	Trident	-ms-
Firefox	Gecko	-moz-
Opera	Presto	-o-
Chrome	Webkit	-webkit-
Safari	Webkit	-webkit-

## 2. transition过渡

- 当 属性值被改变时，过渡就会起作用。
- transition-property：规定设置过渡效果 css 属性名称

```
transition-property:width,height,all;
```

- 可以单选一个属性，也可以选择 ALL 所有
- transition-duration：过渡需要的时间，规定完成过渡效果需要多少秒或毫秒
- transition-delay：过渡效果的延迟，定义过渡效果何时开始
  - 延迟：数值为正数
  - 提前：数值为负数,如果过渡需要时间为 3s，但是延迟为 -2s，过渡效果就会只显示最后一秒的动画
- transition-timing-function：规定速度效果的速度曲线

属性值	描述
linear	匀速
ease (默认值)	逐渐慢下来
ease-in	加速
ease-out	减速
ease-in-out	先加速后减速

- cubic-bezier (贝塞尔曲线)：<https://cubic-bezier.com>

## 3. transform变形

- `translate` : 位移

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

- `translateX` `translateY` `translateZ` (3D)

- `scale` : 缩放 , 值是一个比例值, 正常大小就是 1 , 会以当前元素中心点进行缩放

```
transform : scale( 宽 , 高 )
```

- `scaleX` `scaleY` `scaleZ` (3D)

- `rotate` : 旋转 , 旋转的值, 单位是角度 `deg` 弧度 `rad`

```
transform : rotate( 角度 )
```

- `rotateX` : 3D

- `rotateY` : 3D, `Y` 轴旋转, 会按 `transform-origin` 基点, 顺时针旋转

- `rotateZ` : 和 `rotate` 是等价关系, 正值按顺时针旋转, 负值按逆时针旋转

- `skew` : 斜切

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

- `skewX` : 单位也是角度( `deg` ), 正值向左倾斜, 负值向右倾斜

- `skewY`

- `transform` 注意事项:

1. 变形操作不会影响到其他元素。

2. 变形操作只能添加给块元素, 不能添加给内联元素

3. 复合写法, 可以添加做个变形操作。

- 执行是有顺序的, 先执行后面的操作, 再执行前面的操作。

- `translate` 会受到 `rotate`、`scale`、`skew` 的影响

- `transform-origin` : 设置基点 `x` `y` `z` (3d)

- `transform-origin` : `center center` (默认值), 可以设置数值和单词

- `transform` 在 HTML 中, 可以拥有很多相同属性。依次执行

- `transform: rotateX(30deg) rotateX(10deg);`

- 先旋转 `10deg` , 再旋转 `30deg` . 属于依次执行

## 4. animation动画

animation 属性	描述
<code>animation-name</code>	设置动画的名字 (自定义的)
<code>animation-duration</code>	设置动画的持续时间
<code>animation-delay</code>	动画的延迟

animation 属性	描述
animation-iteration-count	动画的重复次数,默认就是 1 , infinite 无限次数
animation-timing-function	动画的运动形式 ease 、 linear 、 ease-in ...
animation-play-state	控制动画是否暂停 running 让动画运动起来, paused 让动画停下来

- 复合写法: `animation: mybox 4s 1s infinite linear running;`

1. 运动结束后, 默认情况下会停留在起始位置。

2. `@keyframes` : `from -> 0%` ; `to -> 100%`

```
@keyframes mybox{
  0%{ transform: translate(0,0);}
  25%{ transform: translate(200px,0);}
  50%{ transform: translate(200px,200px);}
  75%{ transform: translate(0,200px);}
  100%{ transform: translate(0,0);}
}
```

- `animation-fill-mode` : 规定动画播放之前或之后,其动画效果是否可见。
  - `none` (默认值) : 在运动结束后,回到初始位置,在延迟情况下,让 0% 在延迟后生效。
  - `backwards` : 在延迟情况下, 让 0% 延迟时生效。(在延迟时就执行 0% 的效果)
  - `forwards` : 在运动结束之后, 停到结束位置。
  - `both` : `backwards` 和 `forwards` 同时生效。
- `animation-direction` : 属性定义是否应该轮流反向播放动画。
  - `alternate` : 一次正向( 0%~100% ),一次反向( 100%~0% )
  - `reverse` : 永远都是反向,从 100%~0%
  - `normal` (默认值) : 永远都是正向, 0%~100%
    - 只有播放两次及以上次数的动画, `alternate` 才会生效。
- 扩展 : 通过 `document.styleSheets[0]` 可以获取 当前页面的 style 样式

```
var style = document.styleSheet[0]
style.insertRule( rule, index )
/* rule : 要添加到样式表的规则。
   <1>'@keyframes box{...}'
   <2>'#box{ ... }'
   index: 要把规则插入或附加到 cssRules 数组中的位置。*/

style.rules.length // 查看已有样式条数
style.deleteRule(index) // 删除对应下标的 样式表规则

style.rules // 所有的属性
```

## 5. animate.css动画库

- 官网地址: <https://daneden.github.io/animate.css/>
- 基本使用:
  - `animated` : 基类, 基础的样式, 每个动画效果都需要加
  - `infinite` : 动画的无限次数

## 6. transform3D相关属性

- `transform` :
  - `rotateX()` : 正值向上翻转,
  - `rotateY()` : 正值向右翻转
  - `translateZ()` : 正值向前, 负值向后
  - `scaleZ()` : 立体元素的厚度
    - X轴方向, 网页右方; Y轴方向, 网页下方; Z轴方向, 面向自己
    - 关于旋转的正负: 手变成拳头, 大拇指竖直, 指向对应轴的方向, 其他四个手指弯曲的方向就是正
  - 3D写法
    - `scale3d()` : 三个值 `x` `y` `z`
    - `translate3d()` : 三个值 `x` `y` `z`
    - `rotate3d()` : 4个值 `0|1` (`x` 轴是否旋转) `0|1` (`y` 轴是否旋转) `0|1` (`z` 轴是否旋转) `deg`
- 3D相关属性
  - `perspective` (景深): 离屏幕多远的距离去观察元素, 值越大幅度越小。(景深要加给父元素)
    - `perspective` 相当于在屏幕前架设一个相机, 我们看到的画面, 是浏览器射影在相机里的画面, 值越大, 相机离浏览器页面越远
  - `perspective-origin` : 景深-基点位置, 观察元素的角度
    - 景深基点, 就是摄像机的位置
    - `perspective` 景深和 `perspective-origin` 基点位置 要加在同一个元素上才会生效
  - `transform-origin` : `x` `y` `z`
    - `transform-origin: center center -50px;` (`z` 轴只能写数值)
  - `transform-style` : 3D空间
    - `flat` (默认值 2D)、`preserve-3d` (3D, 产生一个3D空间)
  - `backface-visibility` : 背面隐藏 (可以理解为, 物体的背面看不到)
    - `hidden`、`visible` (默认值)
- 3D扩展: 如果景深(`perspective`)为 `d`, `z` 轴宽度为 `z`, 缩放的数值 `d/(d-z)` 是成比例的

## 7. 背景扩展

- `background-size` : 背景图的尺寸大小
  - `cover` : 覆盖 (背景图覆盖填充整个容器, 等比放大, 会溢出容器)
  - `contain` : 包含 (背景图等比放大, 但不会溢出容器)
  - `background-size : x , y;` 可以设置数值/百分比, 百分比是按照容器大小来定, 而不是图片原始大小。(也可以 `background-size : x , auto;`)
- `background-origin` : 背景图的填充位置
  - `padding-box` (默认)、`border-box`、`content-box`
- `background-clip` : 背景图的裁切方式
  - `border-box` (默认)、`padding-box`、`content-box`
  - 扩展: 以文字为路径, 裁切背景, 可以形成**渐变文字** (IE不支持)
    - `-webkit-background-clip: text;` : 谷歌前缀支持
    - `-moz-background-clip: text;` : 火狐前缀支持
- 注: 这些属性也可以改变背景填充色
- 在复合样式中, 先写 `origin` (位置), 再写 `clip` (裁切)
  - `background: url() no-repeat position/size origin clip;`

## 8. css3渐变

- `linear-gradient` : 线性渐变

```
background-image: linear-gradient(to top, red 25%, blue 75%,);  
/* 方向从下到上, 在75%到25%执行渐变, 其他地方是纯色 */
```

- `to top` : 是从下到上的渐变, 也可以设置 `to left top`, 从右下到左上
  - 也可以设置角度, `0deg` 在下方, 正值按顺时针旋转
- 扩展: 如何在一个背景里面添加两个不渐变的背景色

```
background-image: linear-gradient(red 50%, blue 50%,)
```

- 添加三个不渐变的背景色

```
background-image: linear-gradient(red 40%, blue 40% 60%, yellow 60%)
```

- `radial-gradient` : 径向渐变

```
background-image: radial-gradient(center, shape, size, start-color, ..., last-color);
```

- `center` : 渐变起点的位置, 可以为百分比, 默认是图形的正中心。
- `shape` : 渐变的形状

- `ellipse` 表示椭圆形, `circle` 表示圆形。默认为 `ellipse`, 如果元素形状为正方形的元素, 则 `ellipse` 和 `circle` 显示一样。
- `size`: 渐变的大小, 即渐变到哪里停止, 它有四个值。
  - `closest-side`: 最近边; `farthest-side`: 最远边; `closest-corner`: 最近角; `farthest-corner`: 最远角
- 正常情况下设置颜色的填充就好

```
background-image: radial-gradient(red 5%, green 15%, blue 60%);
```

## 9. 字体图标

- `font-face` 是 CSS 中的一个模块, 他主要是把自己定义的 Web 字体嵌入到你的网页中。
- 好处:
  1. 可以非常方便改变图标大小( `font-size` )和颜色( `color` )
  2. 放大后不会失真
  3. 减少减少请求次数和提高加载速度
  4. 简化网页布局
  5. 减少设计和前端工程师的工作量
  6. 可使用计算机没有提供的字体
- 使用:

```
@font-face{
  font-family : 定义字体名称
}
```

- 通常都是使用 `class` 分别给每个字体图标分开标识, 使用时引用 `class`
- 像 `.woff` 等文件都是做兼容平台处理的 `mac`、`linux`
- 使用字体图标方法:
  - 把图标添加至阿里巴巴项目, 下载项目, 项目文件放入 `css` 文件夹, 具体操作看 `demo.html`
- 如何自定义图标:
  - <https://icomoon.io/app>: 在线生成字体图标, `.css`文件内路径要确认
- 阿里 `iconfont` 使用
  1. 把需要的图标加入购物车添加至项目
  2. 下载项目到桌面
  3. 解压到项目目录
  4. 如果只用字体图标而不用彩色图标, 只需要 `iconfont.css` 和一些兼容平台的文件 (`.eot/.svg/.ttf/.woff/.woff2`)