

移动 Web 第二天

空间转换、动画



黑马程序员
www.itheima.com

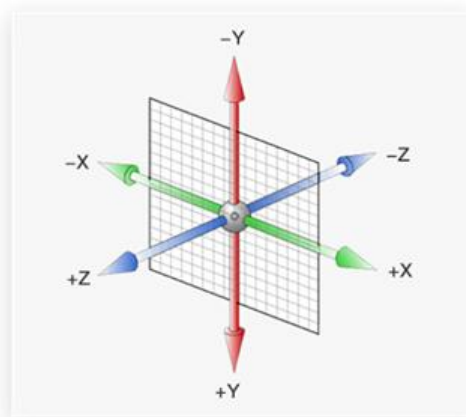
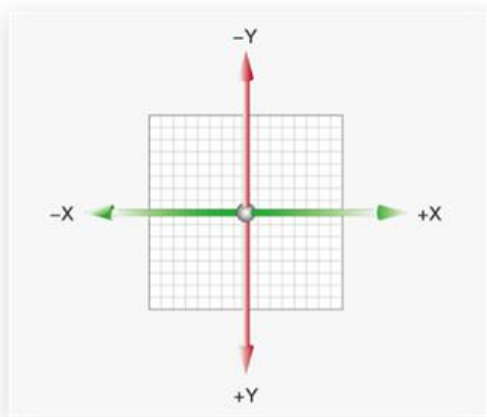
传智教育旗下
高端IT教育品牌



空间转换

空间转换

- 空间：是从坐标轴角度定义的 X 、 Y 和 Z 三条坐标轴构成了一个立体空间， Z 轴位置与视线方向相同。
- 空间转换也叫 3D转换
- 属性：transform



空间转换 - 平移

- 属性

```
transform: translate3d(x, y, z);  
transform: translateX();  
transform: translateY();  
transform: translateZ();
```

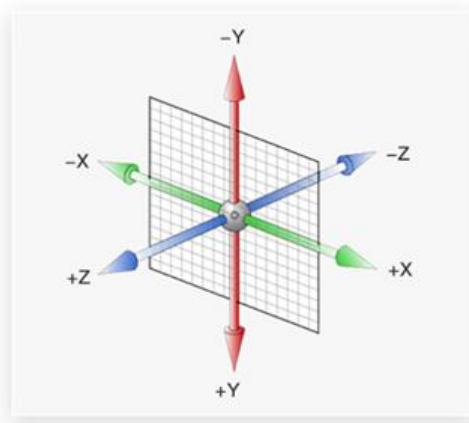
- 取值（正负均可）

- ◆ 像素单位数值

- ◆ 百分比（参照盒子自身尺寸计算结果）

- 提示

- 默认情况下，Z 轴平移没有效果



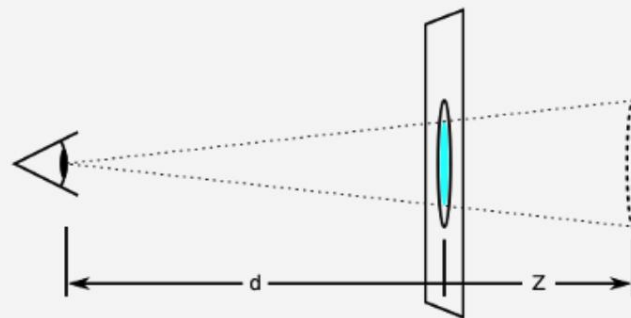
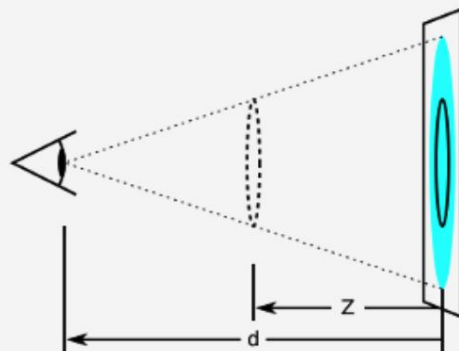
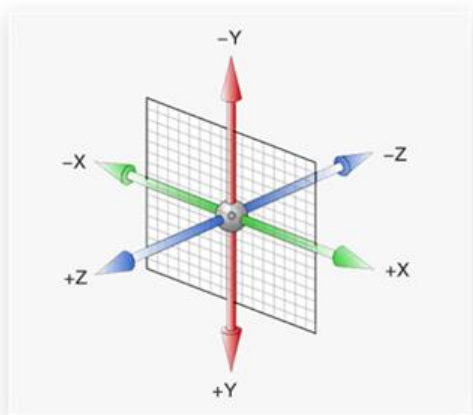
视距 perspective

作用：指定了观察者与 $Z=0$ 平面的距离，为元素添加透视效果

透视效果：近大远小、近实远虚

属性：(添加给父级，取值范围 800-1200)

```
perspective: 视距;
```



空间 - 旋转

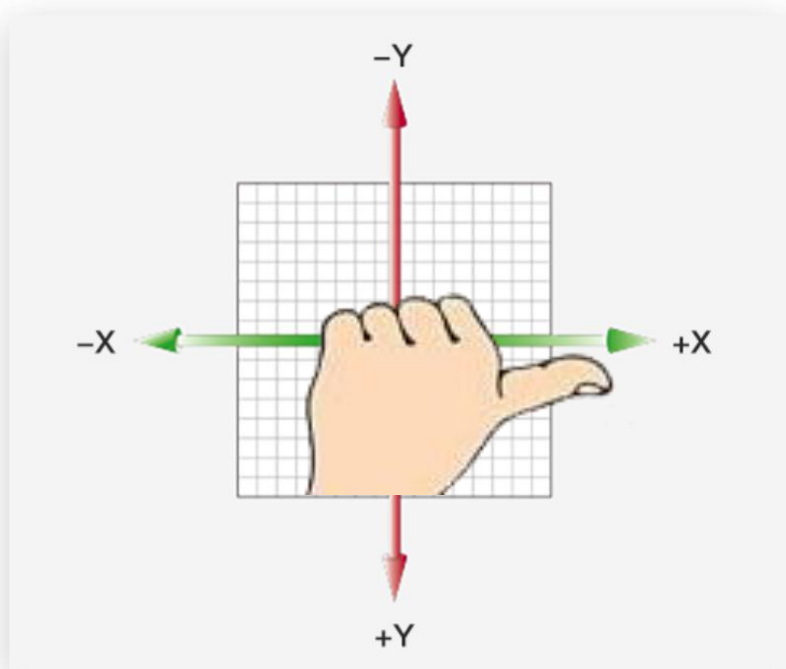
- transform: rotateZ(值);
- transform: rotateX(值);
- transform: rotateY(值);



空间 - 旋转

- 左手法则 - 根据旋转方向确定取值正负

左手握住旋转轴，拇指指向正值方向，其他四个手指弯曲方向为旋转正值方向



空间 - 旋转

- 拓展

- rotate3d(x, y, z, 角度度数)：用来设置自定义旋转轴的位置及旋转的角度
- x, y, z 取值为0-1之间的数字

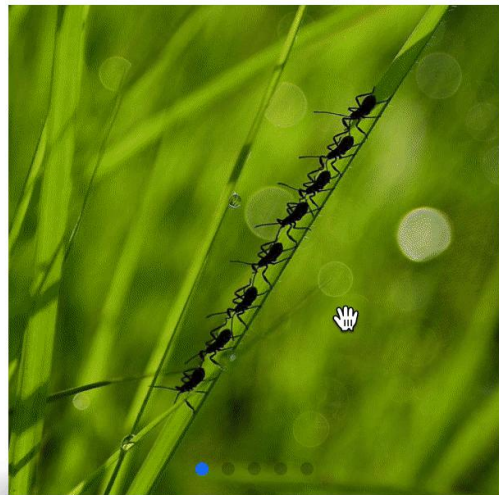
立体呈现 - transform-style

作用：设置元素的子元素是位于 3D 空间中还是平面中

属性名：transform-style

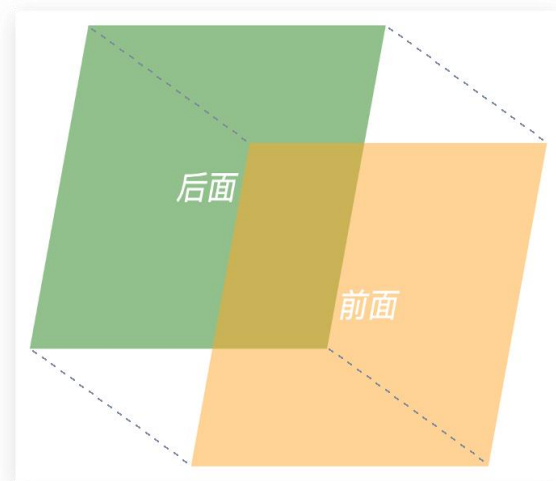
属性值：

- flat：子级处于平面中
- preserve-3d：子级处于 3D 空间



立体呈现 - transform-style

- 呈现立体图形步骤
 1. 父元素添加 `transform-style: preserve-3d;`
 2. 子级定位
 3. 调整子盒子的位置（位移或旋转）
- 提示
 - 空间内，转换元素都有自己独立的坐标轴，互不干扰





案例 - 3D 导航

首页

登录

注册

1. 搭建立方体

- 绿色是立方体的前面
- 橙色是立方体的上面

2. 添加鼠标悬停的旋转效果



案例

案例 - 3D 导航



定位



橙色旋转 (X)



橙色位移 (Z)



绿色位移 (Z)

空间转换 - 缩放

- 属性

```
transform: scale3d(x, y, z);  
transform: scaleX();  
transform: scaleY();  
transform: scaleZ();
```



动画



动画 - animation



- 过渡：实现两个状态间的变化过程
- 动画：实现多个状态间的变化过程，动画过程可控（重复播放、最终画面、是否暂停）

动画 - animation

- 实现步骤

1. 定义动画

```
@keyframes 动画名称 {  
  from {}  
  to {}  
}
```

```
@keyframes 动画名称 {  
  0% {}  
  10% {}  
  .....  
  100% {}  
}
```

2. 使用动画

```
animation: 动画名称 动画花费时长;
```


动画 - animation

```
animation: 动画名称 动画时长 速度曲线 延迟时间 重复次数 动画方向 执行完毕时状态;
```

提示:

- 动画名称和动画时长必须赋值
- 取值不分先后顺序
- 如果有两个时间值，第一个时间表示动画时长，第二个时间表示延迟时间

动画 - animation

属性	作用	取值
animation-name	动画名称	
animation-duration	动画时长	
animation-delay	延迟时间	
animation-fill-mode	动画执行完毕时状态	forwards: 最后一帧状态 backwards: 第一帧状态
animation-timing-function	速度曲线	steps(数字): 逐帧动画
animation-iteration-count	重复次数	infinite为无限循环
animation-direction	动画执行方向	alternate为反向
animation-play-state	暂停动画	paused为暂停，通常配合:hover使用

案例

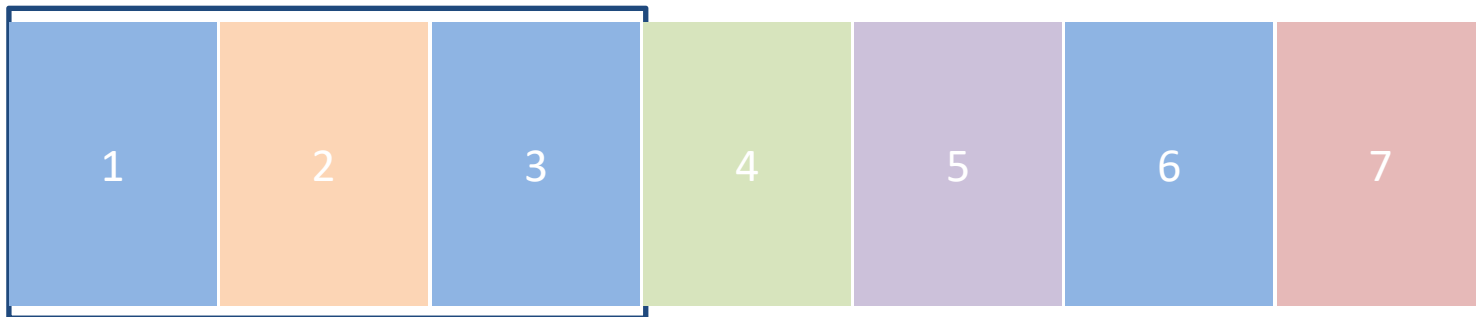
走马灯效果



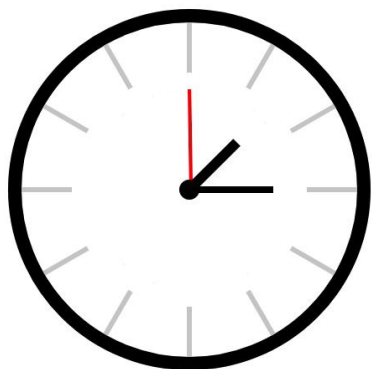
- 列表添加动画效果
- 鼠标移入区域，列表动画暂停

案例 - 走马灯效果

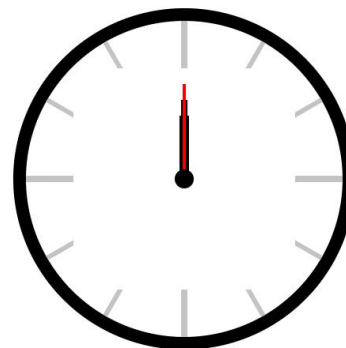
- 无缝动画原理：复制开头图片到结尾位置（图片累加宽度 = 区域宽度）



动画 - 逐帧动画



补间动画



逐帧动画

属性	作用	取值
animation-timing-function	速度曲线	steps(数字): 逐帧动画

案例

逐帧动画



核心原理：

1. steps() 逐帧动画
2. CSS 精灵图

精灵动画制作步骤

1. 准备显示区域
盒子尺寸与一张精灵小图尺寸相同
2. 定义动画
移动背景图（移动距离 = 精灵图宽度）
3. 使用动画
steps(N), N 与精灵小图个数相同

动画 - 多组动画

```
animation:  
    动画1,  
    动画2,  
    动画N  
;
```

```
animation:  
    run 1s steps(12) infinite,  
    move 3s linear forwards  
;
```

案例

全民出游





传智教育旗下高端IT教育品牌