数学、物理与动画

动画的本质

- 定时器 改变元素的属性
- 浏览器/GPU 的渲染过程

时间、位移、速度、加速度

- $t = T \cdot p$
- $s_t = S \cdot p = v \cdot t$
- $v_t = \frac{S \cdot p}{t} = \frac{S}{T}$
- $a = 0$

匀速运动

```
block.addEventListener('click', function(){
  var self = this, startTime = Date.now(),
      duration = 1000;
  setInterval(function(){
    var p = (Date.now() - startTime) / duration;
    self.style.transform = 'rotate(' + (360 * p) +'deg)';
  }, 1000/60);
});
```

时间、位移、速度、加速度

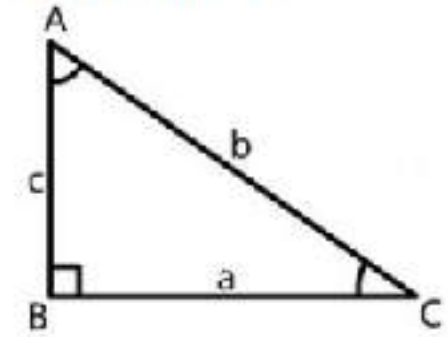
- $t = T \cdot p$
- $s_t = S \cdot p^2 = (\frac{S}{T^2})t^2$
- $v_t = \frac{2S}{T^2} \cdot t = \frac{2Sp}{T}$
- $a = \frac{2S}{T^2}$

匀加速运动（假设初始速度为0）

```
block.addEventListener('click', function(){
  var self = this, startTime = Date.now(),
      distance = 200, T = 2000;
  requestAnimationFrame(function step(){
    var p = Math.min(1.0, (Date.now() - startTime) / T);
    self.style.transform = 'translateX(' + (distance * p * p) +'px)';
    if(p < 1.0) requestAnimationFrame(step);
  });
});
```

平面上的运动

1.直角三角函数



sinA = 对边/斜边 = a/b

cosA = 邻边/斜边 = c/b

tanA　对边/斜边　a/c

```
function updateBall(radio){
    var radians = radio*Math.PI/180;
    xzz = Math.cos(radians)*speed;
    yzz = Math.sin(radians)*speed;
}
```

抛物线运动

$$y = a*x*x + b*x + c$$

限定了抛物线经过中心点(0, 0),
代入y = a * x*x + b*x + c
由于c = 0,
于是b = (y - a*x*x) / x

```
this.b = ( this.driftY - this.curvature * this.driftX * this.driftX ) / this.driftX;
```

```
//x 每一步的X轴的位置
x = self.driftX * ((+new Date - self.begin) / self.duration);
//每一步的Y轴的位置y = a*x*x + b*x + c;   c==0;
y = self.curvature * x * x + self.b * x;
```