

js中图片的路径不是相对于自己js文件的，而是html文件的，也就是通过html文件找图片

```
<!-- 不管代码是在哪个层级的js文件中，最终代码执行的时候，都是以
当前引入js文件的页面为准，来进行路径的处理! -->
```

toFixed (2) : 保留2位小数，但是会转化为字符串

ctx中有canvas属性，是画布对象

```
> ctx
< CanvasRenderingContext2D
  ▶ canvas: canvas#cv
    fillStyle: "#000000"
    filter: "none"
    font: "10px sans-serif"
    globalAlpha: 1
    globalCompositeOperation: "source-over"
```

不建议直接修改对象的属性，建议通过方法去改

按住alt+左键：可以单独给想要的加this (sublime是ctrl+d)

```
// 封装loadImages函数，用于等待图片加载完成，游戏开始
var imgSrc = ['birds', 'land', 'sky', 'pipe1', 'pipe2'];

var loadImages = function( imgSrc, callback ) {
  var count = 0;
  imgsLen = imgSrc.length;
  imgList = {};

  imgSrc.forEach(function(val, index) {
    var img = new Image();
    img.src = 'images/' + val + '.png';
    imgList[ val ] = img;

    img.onload = function() {
      count++;

      if( count >= imgsLen ) {
        callback( imgList );
      }
    };
  });
};

loadImages(imgSrc, function() {
});
```

+=好于是 会出现1px缝隙

```
// 判断大土块背景位置
if( sky_x1 <= -sky.width ) {
  sky_x1 += sky.width * 2;
}
if( sky_x2 <= -sky.width ) {
  sky_x2 += sky.width * 2;
}
```

toFixed (2) : 保留2位小数，但是会转化为字符串

```
// 随机生成的管道高度
var pipeHeight = (Math.random() * 200 + 50).toFixed(2);
```

ctx.isPointInPath (x , y)

```
// ctx.isPointInPath(x, y)
// 作用：用来检测坐标 (x,y) 是否在当前路径中，如果在，就返回true，否则
// 返回false

ctx.rect(100, 100, 200, 200);
ctx.stroke();
console.log( ctx.isPointInPath(50, 50) ) false
console.log( ctx.isPointInPath(200, 200) ) true
```

设计模式

观察者模式

大致思路：

```

// 1. 绘制小鸟
//
// 1.1 绘制序列帧动画（小鸟翅膀煽动）
// 1.2 小鸟下落（改变小鸟的y坐标）
// 1.3 小鸟旋转（根据速度得到角度，速度有正负，角度也就有正负）
// 1.4 小鸟跳动（绑定单击事件）

// 2 封装Sky和Land
//
// 运动方式：匀速运动（ $s = v * t$ ）

// 3 封装Fly 全局对象
//
// 整个游戏，在全局环境中只有一个对象
// Fly对象中，包含了所有的构造函数以及，各种工具性方法（utils）

// 4 封装 Game
//
// 跟游戏相关的内容都属于游戏对象
// 比如： 时间、游戏的图片资源、游戏的角色，游戏的开始和结束

// 5 工厂模式
//
// 统一处理创建对象的过程

```

```

// 游戏开始是需要等到所有的图片资源加载完成，才能执行
Fly.loadImages(that.imgSrc, function( imgList ) {

    // 初始化角色
    that.initRoles( imgList );

    // 渲染游戏
    that.draw( imgList );

    // 绑定事件
    that.bindEvent()
});

```

很好的利用参数，其他函数想访问此参数，可以通过回调函数，回调函数里传入参数

面向对象、anglur、js、vue、