# 学习目标

1. 通过这个小游戏，进一步熟悉面向对象的代码书写方式
2. 入门H5小游戏，以后上班多一个选择

# 效果预览

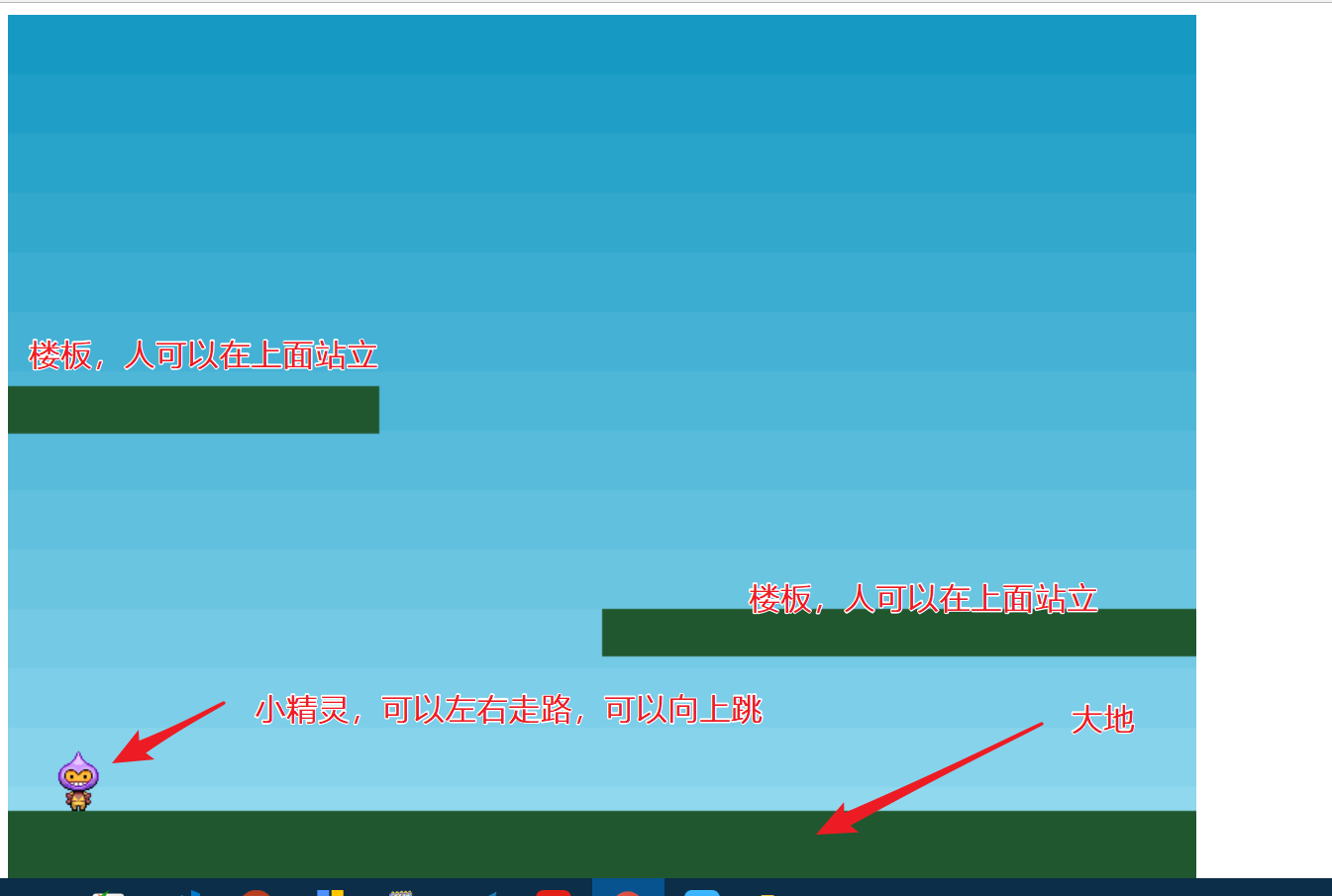
<https://zhengwei1949.github.io/making_your_first_h5_game/part9_1.html>

# 案例来源

<http://phaser.io/tutorials/making-your-first-phaser-game>

https://github.com/zhengwei1949/making\_your\_first\_h5\_game

# 需求说明：

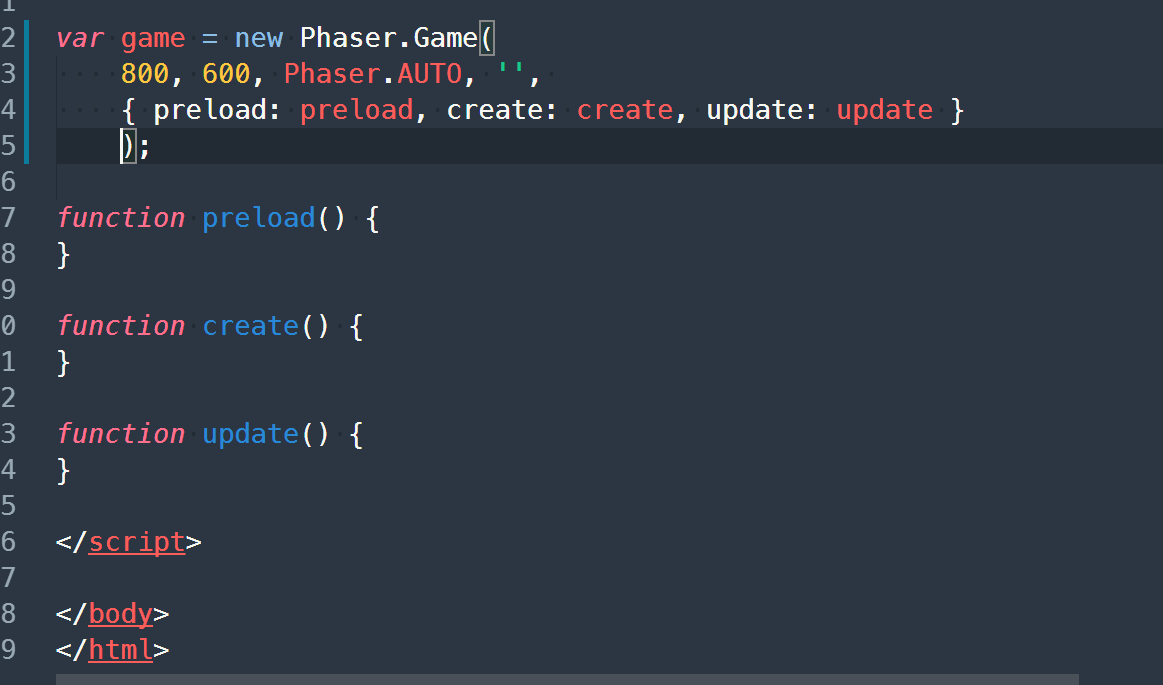


吃掉星星可以得到积分

# Phaser.js优势

1. 无须像白鹭引擎一样，要安装配置一堆的环境才能玩，只需要你的浏览器支持canvas标签就可以玩（当然，需要把代码放到服务器中去玩，否则存在跨域的问题）

# 第一步、搭建基本结构



讲解：

1、我们使用Phaser.Game创建了game对象，我们后面的所有的东西都是要加载到game对象当中

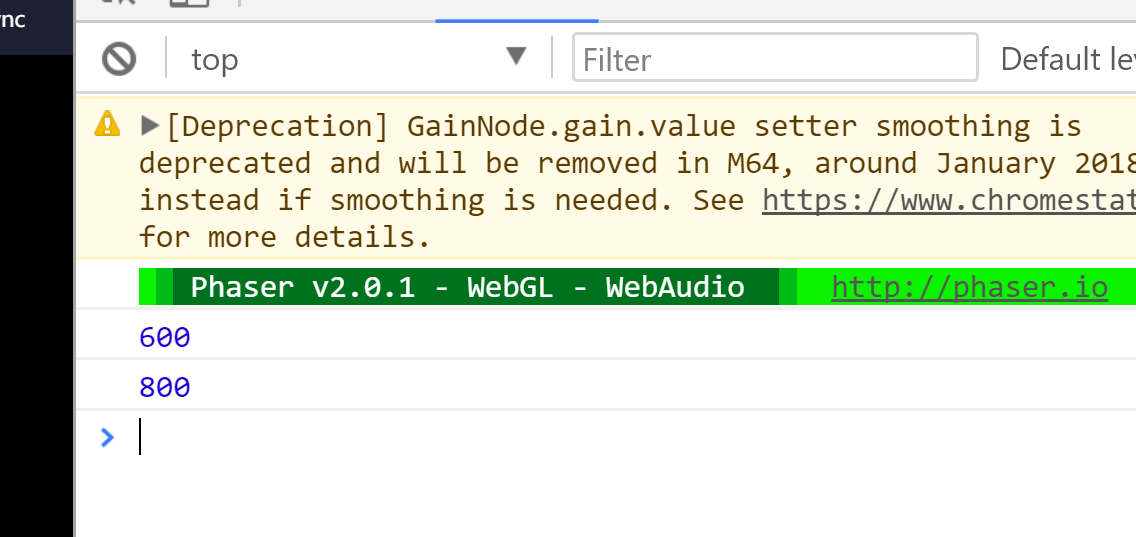
2、800，600指的是我们的游戏的世界是800\*600

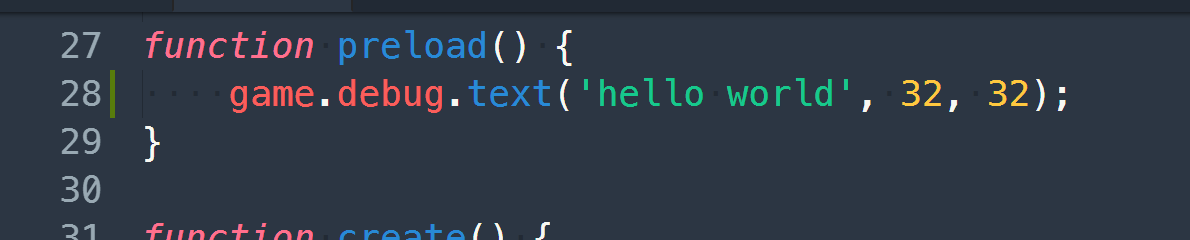
3、第三个参数的值可以是：Phaser.CANVAS,Phaser.WEBGL,Phaser.AUTO 推荐大家使用Phaser.AUTO,这样默认会使用webGL的环境，如果不支持webGL会退化为canvas的环境

第四个参数和页面中的id元素相关联，我们这里默认是空的，意思是会自动加载到body当中

1. 最后一个参数里有三个函数,preload,create,update，这块并不是必须的，一般正式的游戏会创建很多的场景state,然后这些函数会在每个场景当中，我们这个例子比较简单，所以，这块就只有一个默认的场景，函数全写在这块
2. 每个生命周期函数的执行次数
   1. preload ---> create ---> update
   2. preload,create只执行一次，update执行无数次
   3. 我们可以在preload,create,update阶段打印整个世界的宽高







# 第二步、加载资源



讲解：

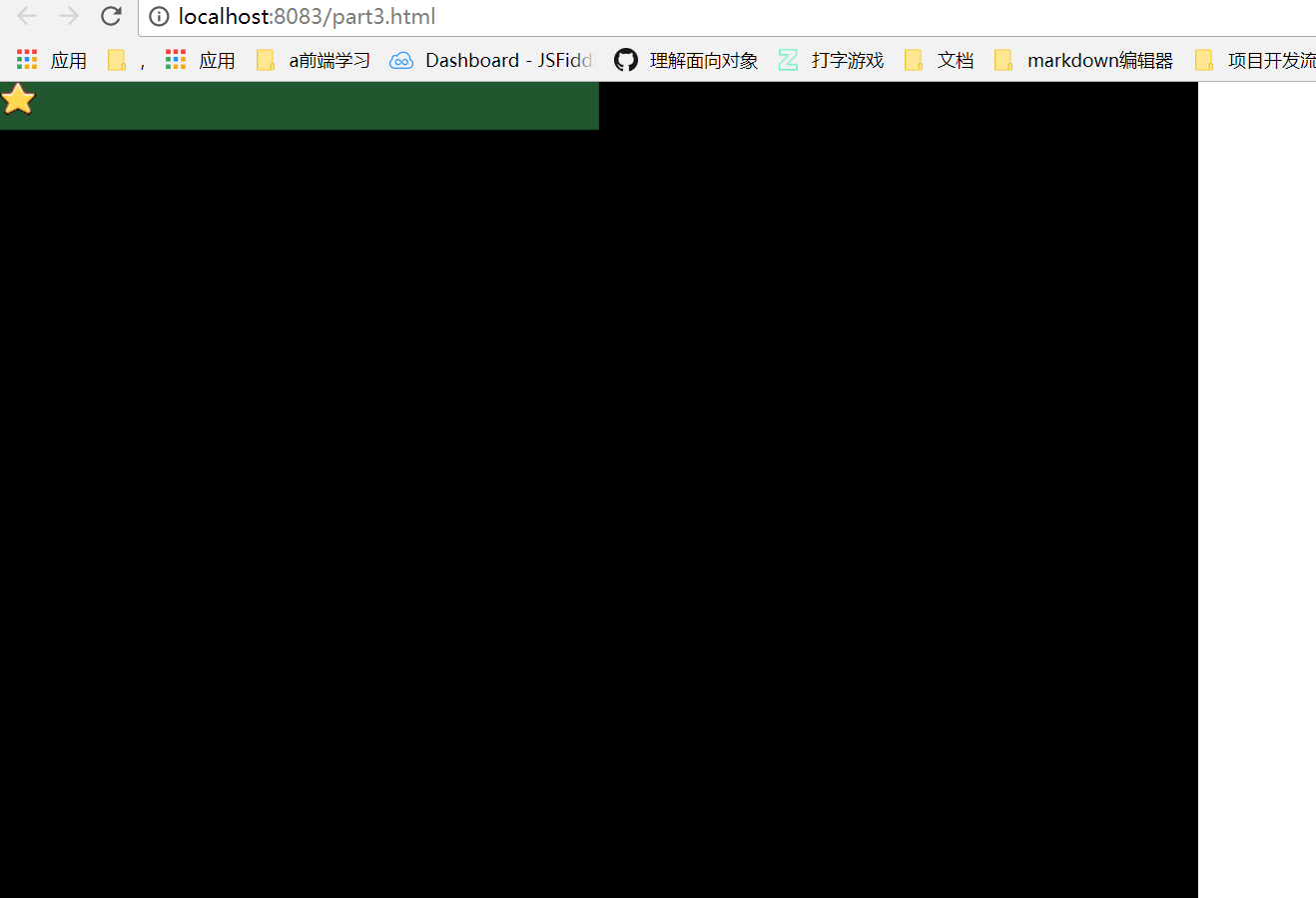
1、如果是精灵表(spritesheet)，则要设置一下每个小精灵的宽度和高度，这块大家先这么做，后会明白为什么要这样写的

# 第三步、创建一个星星精灵和大地

这块是为了测试代码是否ok,这二个小精灵后面会移除掉



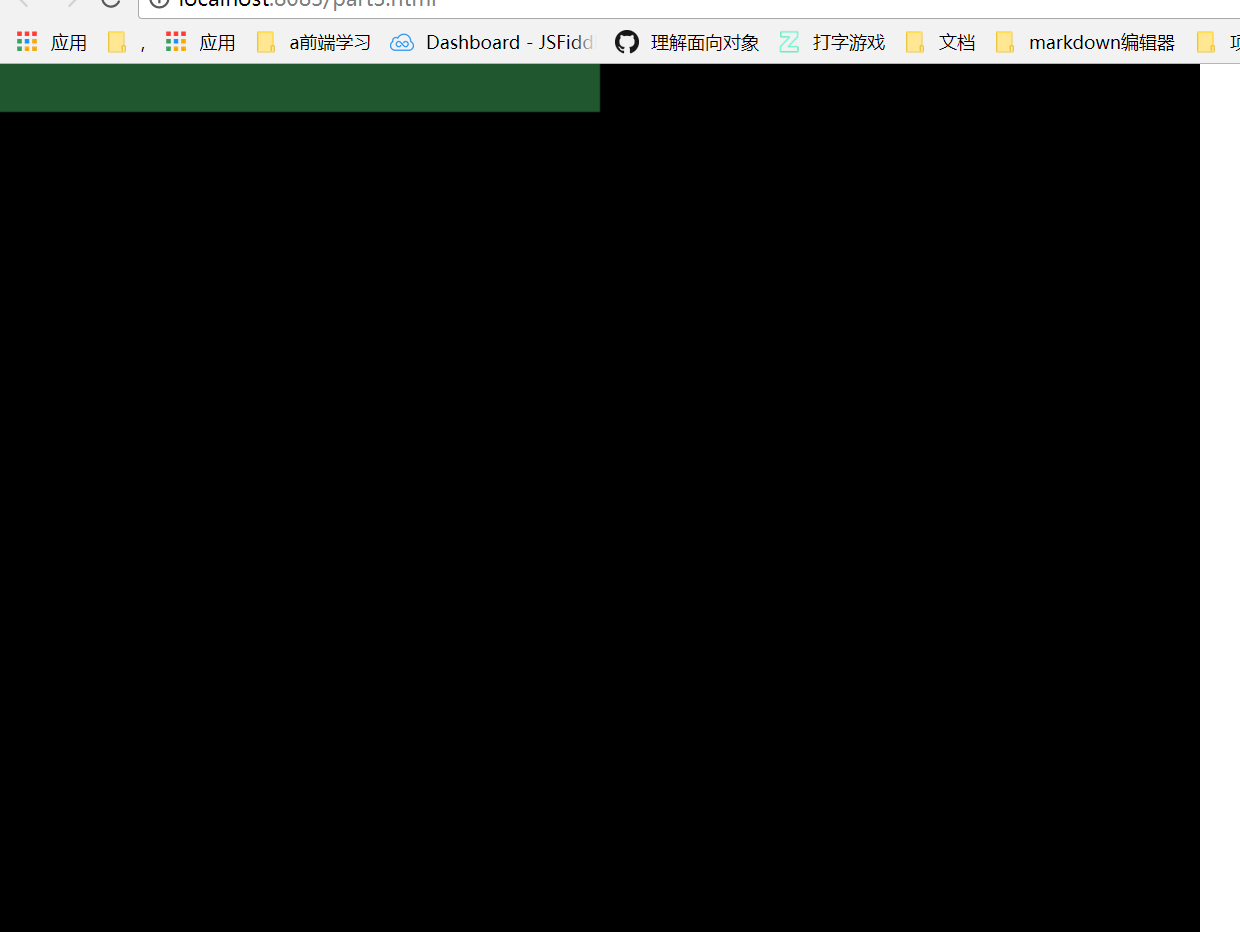
效果预览：



讲解：

1. 精灵的顺序是后面的层级比较高，大家不要反着如下这样写，否则会看不到星星：





2、默认的坐标原点是左上角(0,0),当然这块是可以修改的（通过camera，暂时我们用不上，不用管)

# 第四步、创建整个世界