**特点**：

1.一款跨平台、免费、开源HTML5游戏开发框架。

2.内部由pixi.js作为渲染引擎，有canvas和WebGl两种渲染模式。

3.简单易用的资源加载系统，一句代码就可加载图片、声音、文本文件、精灵图、二进制等文件。

4.多种过渡动画及模式供选择。

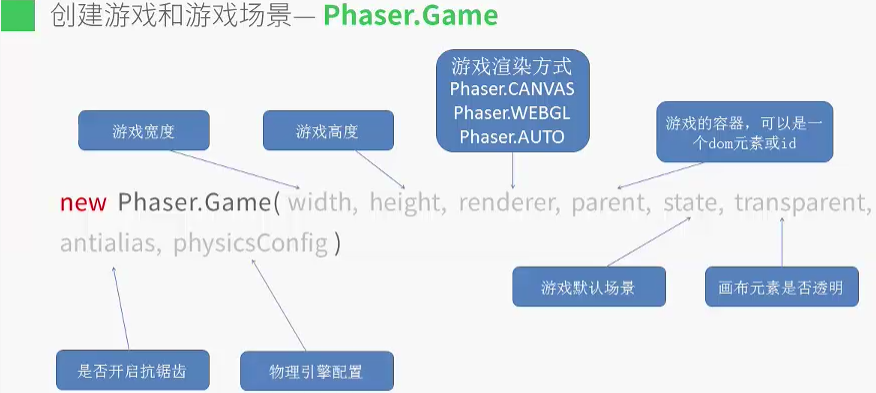
5.支持鼠标、键盘、触摸、pointer（微软独有）等多种用户交互事件。

6.支持html5 Audio和Web Audio（同一时间可播放多个声音）两种声音模式。

7.内置三个引擎，可相互配合，也能单独使用。甚至可以自定义物理引擎插件。

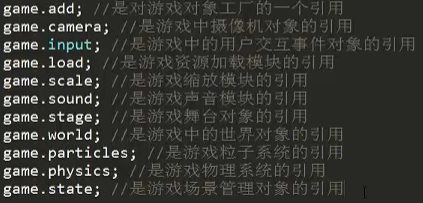
必须放到一个本地服务器环境中才能运行。

**创建游戏和游戏场景**



var game = new Game(300,400,Phaser.AUTO,'container');

控制游戏的暂停或继续：game.paused = true或false，可以在任何时候进行控制。



**游戏场景**：是指游戏中不同的内容或界面，比如游戏菜单界面为一个场景，真正玩游戏的界面为一个场景，不同关卡又是不同场景等等。场景能把一个复杂的游戏分成许多小块，从而简化游戏的开发。

场景在Phaser中叫做State。虽然把它翻译成场景并不太准确，但基本上我们可以理解成State就是场景的意思，只不过这个场景的概念可能更加广泛，例如一个只是执行某些准备工作而没有实质性画面显示出来的State，我们也把它称作一个场景。一个Phaser开发的游戏就是由众多的场景组成的。

创建场景对象的两种方式：



preload(),create(),update(),render()至少要存在一个。一个场景对象就是由这些方法构成的。

init() 一些场景的初始化代码可以写在这个方法里面。

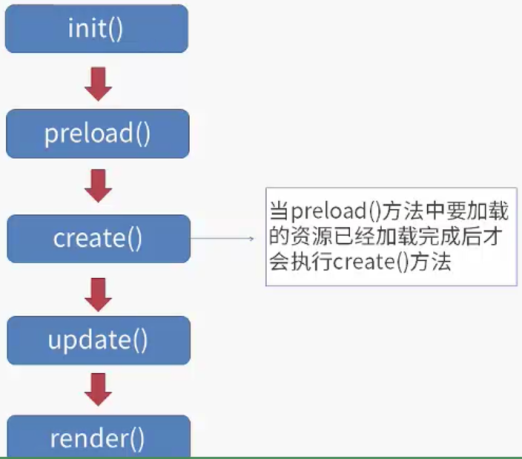
preload()用来加载游戏资源

create() 创建游戏显示对象或注册事件等

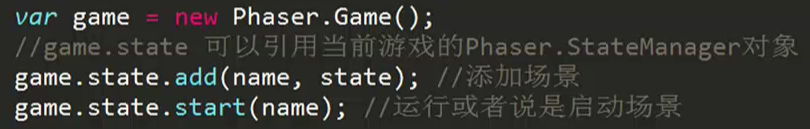
update() 在游戏的每一帧都会被调用（一般是每秒60帧），用来书写要在每一 帧都会被执行的代码

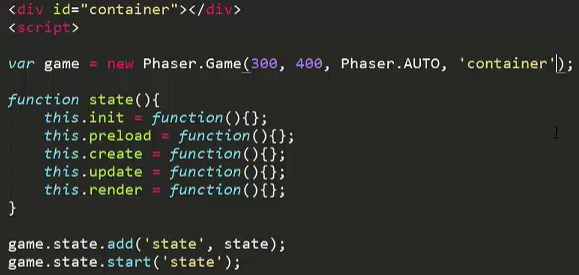
render() 在游戏的每一个渲染周期都会被调用（默认一帧就是一个渲染周期）， 用来做一些自定义渲染工作

这几个方法的执行顺序：



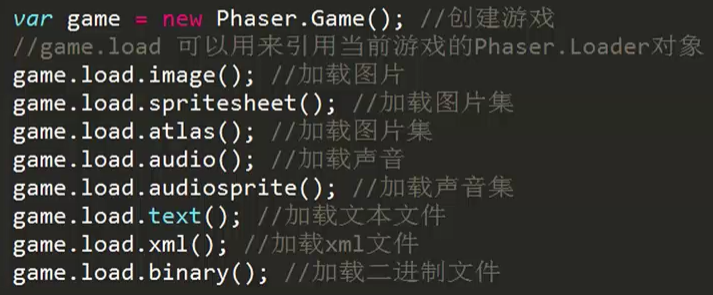
Phaser.StateManager：场景管理对象。

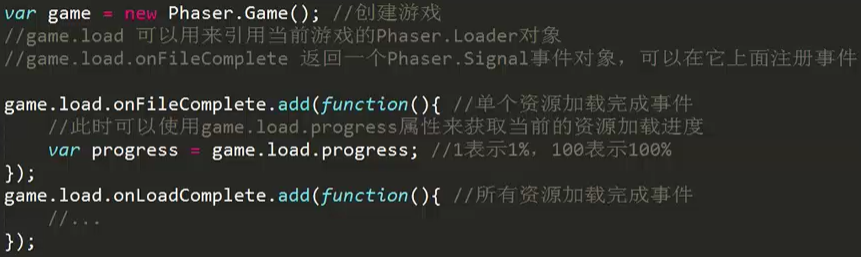




**游戏资源加载**（图片、声音以及其他文件资源）

Phaser中游戏资源的加载使用的是Phaser.Loader对象。





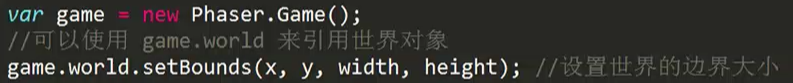


**舞台、世界和摄像机对象**

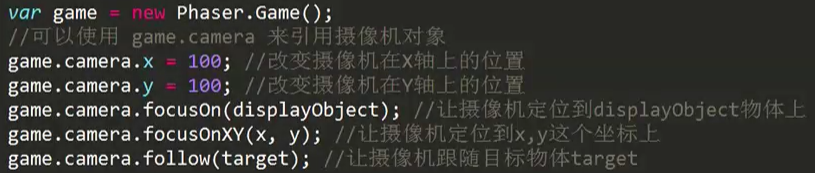
Phaser.Stage舞台对象（不可变大小）



Phaser.World世界对象（可变大小）



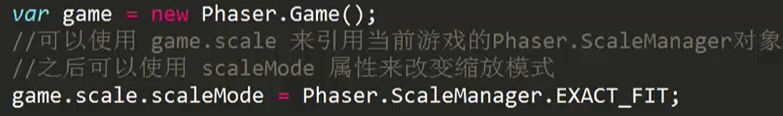
Phaser.Camera摄像机对象

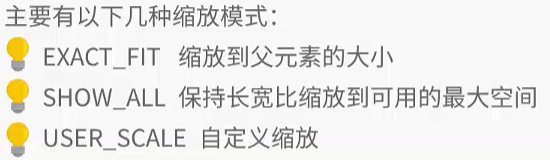




**游戏的缩放控制**

Phaser.ScaleManager是用来控制缩放的对象，通过它我们能进行缩放从而达到适配屏幕的目的。





缩放代码一般写在init方法里面。

