python编程从入门到实践

# 第8章 函数

1.函数编写指南

1).编写函数时,需要牢记几个细节。应给函数指定描述性名称,且只在其中使用小写字母和下划线。描述性名称可帮助你和别人明白代码想要做什么。给模块命名时也应遵循上述约定。

2).每个函数都应包含简要地阐述其功能的注释,该注释应紧跟在函数定义后面,并采用文档字符串格式。文档良好的函数让其他程序员只需阅读文档字符串中的描述就能够使用它:他们完全可以相信代码如描述的那样运行;只要知道函数的名称、需要的实参以及返回值的类型,就能在自己的程序中使用它。

3).给形参指定默认值时,等号两边不要有空格:

4).PEP 8 ( https://www.python.org/dev/peps/pep-0008/ )建议代码行的长度不要超过 79 字符,这样只要编辑器窗口适中,就能看到整行代码。如果形参很多,导致函数定义的长度超过了79 字符,可在函数定义中输入左括号后按回车键,并在下一行按两次 Tab 键,从而将形参列表和只缩进一层的函数体区分开来。

5)如果程序或模块包含多个函数,可使用两个空行将相邻的函数分开,这样将更容易知道前一个函数在什么地方结束,下一个函数从什么地方开始。

6)所有的 import 语句都应放在文件开头,唯一例外的情形是,在文件开头使用了注释来描述整个程序

# 第9章 类

1.❹处的 super() 是一个特殊函数,帮助 Python 将父类和子类关联起来。这行代码让 Python 调用 ElectricCar 的父类的方法 \_\_init\_\_() ,让 ElectricCar 实例包含父类的所有属性。父类也称为超类 ( superclass ),名称 super 因此而得名。

为测试继承是否能够正确地发挥作用,我们尝试创建一辆电动汽车,但提供的信息与创建普通汽车时相同。在❺处,我们创建 ElectricCar 类的一个实例,并将其存储在变量 my\_tesla 中。这行代码调用 ElectricCar 类中定义的方法 \_\_init\_\_() ,后者让 Python 调用父类 Car 中定义的方法 \_\_init\_\_() 。我们提供了实参 'tesla' 、 'models' 和 2016

2.在 Python 2.7 中,继承语法稍有不同, ElectricCar 类的定义类似于下面这样:

class Car(object):

def \_\_init\_\_(self, make, model, year):

--snip--

class ElectricCar(Car):

def \_\_init\_\_(self, make, model, year):

super(ElectricCar, self).\_\_init\_\_(make, model, year)

--snip--

函数 super() 需要两个实参:子类名和对象 self 。为帮助 Python 将父类和子类关联起来,这些实参必不可少。另外,在 Python 2.7 中使用继承时,务必在定义父类时在括号内指定 object 。

# 第10章 文件和异常

pass

# 第11章 测试代码

1. 单元测试和测试用例:

Python 标准库中的模块 unittest 提供了代码测试工具。**单元测试** 用于核实函数的某个方面没有问题;**测试用例** 是一组单元测试,这些单元测试一起核实函数在各种情形下的行为都符合要求。良好的测试用例考虑到了函数可能收到的各种输入,包含针对所有这些情形的测试。**全覆盖式测试** 用例包含一整套单元测试,涵盖了各种可能的函数使用方式。对于大型项目,要实现全覆盖可能很难。通常,最初只要针对代码的重要行为编写测试即可,等项目被广泛使用时再考虑全覆盖。

Pass

# 项目1 外星人入侵

# 第12章 武装飞船

创建游戏是趣学语言的理想方式。看别人玩你编写的游戏让你很有满足感,而编写简单的游戏有助于你明白专业级游戏是怎么编写出来的。在阅读本章的过程中,请动手输入并运行代码,以明白各个代码块对整个游戏所做的贡献,并尝试不同的值和设置,这样你将对如何改进游戏的交互性有更深入的认识。

12.1规划项目

在游戏《外星人入侵》中,玩家控制着一艘最初出现在屏幕底部中央的飞船。玩家可以使用箭头键左右移动飞船,还可使用空格键进行射击。游戏开始时,一群外星人出现在天空中,他们在屏幕中向下移动。玩家的任务是射杀这些外星人。玩家将所有外星人都消灭干净后,将出现一群新的外星人,他们移动的速度更快。只要有外星人撞到了玩家的飞船或到达了屏幕底部,玩家就损失一艘飞船。玩家损失三艘飞船后,游戏结束。

12.2安装pygame

1.在Linux系统中检查是否安装了pip:在终端中执行:pip3 –version

2. 在Linux系统中检查是否安装了pip: 在终端中执行 sudo python3 get-pip.py

3.如果你使用的是 Python 3 ,就需要执行两个步骤:安装 Pygame 依赖的库;下载并安装 Pygame 。

执行下面的命令来安装 Pygame 依赖的库(如果你开始终端会话时使用的是命令 python3.5 ,请将 python3-dev 替换为 python3.5-dev ):

$ sudo apt-get install python3-dev mercurial

$ sudo apt-get install libsdl-image1.2-dev libsdl2-dev libsdl-ttf2.0-dev

这将安装运行《外星人入侵》时需要的库。如果你要启用 Pygame 的一些高级功能,如添加声音的功能,可安装下面这些额外的库:

$ sudo apt-get install libsdl-mixer1.2-dev libportmidi-dev

$ sudo apt-get install libswscale-dev libsmpeg-dev libavformat-dev libavcode-dev

$ sudo apt-get install python-numpy

接下来,执行下面的命令来安装 Pygame (如有必要,将 pip 替换为 pip3 ):

$ pip install --user hg+http://bitbucket.org/pygame/pygame

告知你 Pygame 找到了哪些库后,输出将暂停一段时间。请按回车键,即便有一些库没有找到。你将看到一条消息,说明成功地安装了 Pygame 。

要确认安装成功,请启动一个 Python 终端会话,并尝试执行下面的命令来导入 Pygame :

$ python3

>>> import pygame

>>>

12.3.2在 Pygame 中,颜色是以 RGB 值指定的。这种颜色由红色、绿色和蓝色值组成,其中每个值的可能取值范围都为 0~255 。颜色值 (255, 0, 0) 表示红色, (0, 255, 0) 表示绿色,而 (0, 0,255) 表示蓝色。通过组合不同的 RGB 值,可创建 1600 万种颜色。在颜色值 (230, 230, 230) 中,红色、蓝色和绿色量相同,它将背景设置为一种浅灰色

12.3.3创建设置类

每次给游戏添加新功能时,通常也将引入一些新设置。下面来编写一个名为 settings 的模块,其中包含一个名为 Settings 的类,用于将所有设置存储在一个地方,以免在代

码中到处添加设置。这样,我们就能传递一个设置对象,而不是众多不同的设置。另外,这让函数调用更简单,且在项目增大时修改游戏的外观更容易:要修改游戏,只需修改

settings.py 中的一些值,而无需查找散布在文件中的不同设置。

12.4添加飞船图像

在游戏中几乎可以使用任何类型的图像文件,但使用位图( .bmp )文件最为简单,因为 Pygame 默认加载位图。虽然可配置 Pygame 以使用其他文件类型,但有些文件类型要求你在

计算机上安装相应的图像库。大多数图像都为 .jpg 、 .png 或 .gif 格式,但可使用 Photoshop 、 GIMP 和 Paint 等工具将其转换为位图

12.4.1选择用于表示飞船的图像后,需要将其显示到屏幕上。我们将创建一个名为 ship 的模块,其中包含 Ship 类,它负责管理飞船的大部分行为。

使用 get\_rect() 获取相应 surface 的属性 rect：Pygame 的效率之所以如此高,一个原因是它让你能够像处理矩形( rect 对象)一样处理游戏元

素,即便它们的形状并非矩形。像处理矩形一样处理游戏元素之所以高效,是因为矩形是简单的几何形状。这种做法的效果通常很好,游戏玩家几乎注意不到我们处理的不是游

戏元素的实际形状。将游戏元素居中,可设置相应 rect 对象的属性 center 、 centerx 或 centery 。要让游戏元素与屏幕边缘对齐,可使用属性 top 、 bottom 、 left 或 right ;要调整游

戏元素的水平或垂直位置,可使用属性 x 和 y ,它们分别是相应矩形左上角的 x 和 y 坐标

注意

在 Pygame 中,原点 (0, 0) 位于屏幕左上角,向右下方移动时,坐标值将增大。在 1200×800 的屏幕上,原点位于左上角,而右下角的坐标为 (1200, 800) 。

12.5 重构:模块 game\_functions

创建一个名为 game\_functions 的新模块,它将

存储大量让游戏《外星人入侵》运行的函数。通过创建模块 game\_functions ,可避免 alien\_invasion.py 太长,并使其逻辑更容易理解

12.6驾驶飞船

12.6.2允许不断移动

玩家按住右箭头键不放时,我们希望飞船不断地向右移动,直到玩家松开为止。我们将让游戏检测 pygame.KEYUP 事件,以便玩家松开右箭头键时我们能够知道这一点;然

后,我们将结合使用 KEYDOWN 和 KEYUP 事件,以及一个名为 moving\_right 的标志来实现持续移动。

飞船不动时,标志 moving\_right 将为 False 。玩家按下右箭头键时,我们将这个标志设置为 True ;而玩家松开时,我们将这个标志重新设置为 False 。

飞船的属性都由 Ship 类控制,因此我们将给这个类添加一个名为 moving\_right 的属性和一个名为 update() 的方法。方法 update() 检查标志 moving\_right 的状态,

如果这个标志为 True ,就调整飞船的位置。每当需要调整飞船的位置时,我们都调用这个方法。

12.6.4调整飞船速度：