

黑马程序员

《Python 快速编程入门（第2版）》

教学设计

课程名称：Python 快速编程入门

授课年级：2020 年级

授课学期：2020 学年第二学期

教师姓名：某某老师

2020 年 09 月 09 日

课题名称	第 10 章 Python 计算生态	计划学时	8 课时
内容分析	计算生态学是采用计算理论、计算方法和计算机技术研究现代生态学问题的科学。由于开源的特性，Python 自诞生至今逐步建立起了全球最大的编程计算生态，本章将为大家简单地介绍 Python 计算生态、Python 生态库的构建与发布和常用的 Python 库。		
教学目标及基本要求	1. 了解 Python 计算生态及各应用领域的常用库 2. 了解 Python 生态库的构建与发布 3. 掌握 time、random、turtle 库的基本用法 4. 了解 wordcloud、pygame 库的基本用法		
教学重点	1. 掌握 time、random、turtle 库的基本用法		
教学难点	1. 掌握 time、random、turtle 库的基本用法 2. 了解 wordcloud、pygame 库的基本用法		
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解		
教学过程	<p style="text-align: center;">第一课时</p> <p style="text-align: center;">(Python 计算生态概述)</p> <p>一、创设情境，导入 Python 计算生态的概念</p> <p>(1) 教师根据课件，引出 Python 计算生态的概念。</p> <p>计算生态学是采用计算理论、计算方法和计算机技术研究现代生态学问题的科学。由于开源的特性，Python 自诞生至今逐步建立起了全球最大的编程计算生态。</p> <p>(2) 明确学习目标</p> <ul style="list-style-type: none"> ● 要求学生了解 Python 计算生态及各应用领域的常用库 <p>二、进行重点知识的讲解</p> <p>(1) 教师根据课件，介绍 Python 计算生态涵盖的领域。</p> <p>Python 计算生态涵盖网络爬虫、数据分析、文本处理、数据可视化、图形用户界面、机器学习、Web 开发、网络应用开发、游戏开发、虚拟现实、图形艺术等多个领域，为各个领域的 Python 使用者提供了极大便利。</p>		

(2) 教师根据课件，介绍网络爬虫领域及常用库。

网络爬虫是一种按照一定的规则，自动从网络上抓取信息的程序或者脚本。通过网络爬虫可以代替手工完成很多工作。Python 计算生态通过 Requests、Python-Goose、Re、Beautiful Soup、Scrapy 和 PySpider 等库为这些操作提供了强有力的支持。

(3) 教师根据课件，介绍数据分析领域及常用库。

数据分析指用适当的统计分析方法对收集来的大量数据进行分析，将它们加以汇总、理解与消化，以求最大化地发挥数据的作用。Python 计算生态通过 Numpy、Pandas、SciPy 库为数据分析领域提供支持。

(4) 教师根据课件，介绍文本处理领域及常用库。

文本处理即对文本内容的处理，包括文本内容的分类、文本特征的提取、文本内容的转换等等。Python 计算生态通过 Jieba、PyPDF2、Python-docx、NLTK 等库为文本处理领域提供支持。

(5) 教师根据课件，介绍数据可视化领域及常用库。

数据可视化是一门关于数据视觉表现形式的科学技术研究，它既要有效传达数据信息，也需兼顾信息传达的美学形式，二者缺一不可。Python 计算生态主要通过 Matplotlib、Seaborn、Mayavi 等库为数据可视化领域提供支持。

(6) 教师根据课件，介绍机器学习领域及常用库。

机器学习是一门涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科的多领域交叉学科，该学科旨在研究计算机如何模拟或实现人类的学习行为，以获取新的知识或技能、重新组织已有知识结构并不断改善自身。Python 计算生态主要通过 Scikit-learn、TensorFlow、MXNet 库为机器学习领域提供支持。

(7) 教师根据课件，介绍图形用户界面领域及常用库。

图形用户界面（Graphical User Interface，简称 GUI）指采用图形方式显示的计算机操作用户界面，该界面允许用户使用鼠标、键盘等输入设备操纵屏幕上的图标或菜单选项，以选择命令、调用文件、启动程序或执行一些其他的日常任务。Python 计算生态通过 PyQt5、WxPython、PyGObject 库为图形用户界面领域提供支持。

(8) 教师根据课件，介绍 Web 开发领域及常用库。

Web 开发指基于浏览器而非桌面进行的程序开发。Python 计算生态通过

Django、Tornado、Flask、Twisted 等库为 Web 开发领域提供了支持。

(9) 教师根据课件，介绍网络应用开发领域及常用库。

网络应用开发指以网络为基础的应用程序的开发，Python 计算生态通过 WeRoBot、aip、MyQR 等库为网络应用开发领域提供支持。

(10) 教师根据课件，介绍游戏开发领域及常用库。

Python 计算生态通过 PyGame、Panda3D 库为游戏开发领域提供支持。

(11) 教师根据课件，介绍图形艺术领域及常用库。

图形艺术是一种通过标志来表现意义的艺术。标志是一些单纯、显著、易识别的具有指代性或具有表达意义、情感和指令等作用的物象、图形或文字符号，也是图形艺术的表现手段。Python 计算生态通过 Quads、ascii_art 和 turtle 库为图形艺术领域提供支持。

(12) 教师根据课件，介绍图像处理领域及常用库。

图像处理一般指数字图像处理，图像处理技术一般包括图像压缩、增强和复原、图像匹配、描述和识别。Python 通过 Numpy、Scipy、Pillow、OpenCV-Python 等库为图像处理领域提供支持。

三、归纳总结，布置课后作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要了解的 Python 计算生态及各领域的常用库。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第二课时

(Python 生态库的构建与发布，time 库)

一、回顾上节课的内容，继续讲解本课时的知识

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾上个课时所学习的内容，继续介绍本课时的内容。

上节课主要为大家介绍了计算生态及各应用领域的常用库，接下来，本节课将为大家介绍生态库的构建与发布、time 库。

(3) 明确学习目标

- 要求学生了解生态库的构建与发布
- 要求学生掌握 `time` 库的使用

二、进行重点知识的讲解

(1) 教师根据课件，介绍库的分类及表现形式。

Python 中的库 (Library) 分为 Python 标准库 (Standard Library) 和第三方库 (Third-Party Library)，其中标准库会随 Python 解释器一同安装，可在程序中直接导入与使用；第三方库是由 Python 使用者编写与分享的库，在使用之前需要额外进行安装。事实上 Python 中的库只是一种对特定功能集合的统一说法而非严格定义，它的具体表现形式为模块 (Module) 和包 (Package)。

(2) 教师根据课件，介绍模块的构建与使用，并使用代码进行演示。

(3) 教师根据课件，介绍包的构建与导入，并使用代码进行演示。

将模块放入一个文件夹，并在该文件夹中创建 `__init__.py` 文件，就构建了一个 Python 包。

(4) 教师根据课件，介绍库发布的内容，并使用代码进行演示。

(5) 教师根据课件，介绍 `time` 库的内容，并使用代码进行演示。

三、归纳总结，布置作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要熟悉或掌握的知识点，包括模块的构建与使用、包的构建与导入、`time` 库。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第三课时

(`random` 库, `turtle` 库)

一、回顾上节课内容，继续讲解本节课的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾上个课时所学习的内容，继续介绍本课时内容。

上节课主要为大家介绍了生态库的构建与发布、time 库，接下来，本节课先继续为大家介绍两个内置库——random、turtle 库。

(3) 明确学习目标

- 要求学生掌握 random、turtle 库的使用

二、进行重点知识的讲解

(1) 什么是 random 库？

random 是 Python 内置的标准库，在程序中导入该库，可利用库中的函数生成随机数据。

(2) 教师根据课件，介绍 random 库的用法，并使用代码进行演示。

(3) 什么是 turtle 库？

turtle（海龟）是 Python 内置的一个标准模块，它提供了绘制线、圆以及其他形状的函数，使用该模块可以创建图形窗口，在图形窗口中通过简单重复动作直观地绘制界面与图形。

(4) 教师根据课件，介绍 turtle 库的用法，并使用代码进行演示。

三、归纳总结，布置作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要掌握的内置库：random 和 turtle。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第四课时

（实训案例，jieba 库）

一、回顾上节课内容，继续讲解本节课的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾上个课时所学习的内容，继续介绍本课时内容。

上节课主要为大家介绍了两个内置库——random、turtle 库，接下来，本节课先结合前面的知识点演示两个精彩的实例，再介绍一个第三方库——jieba。

(3) 明确学习目标

- 要求学生掌握 `time`、`turtle` 库的使用
- 要求学生了解 `jieba` 库的使用

二、进行重点知识的讲解

- (1) 教师根据课件，介绍 10.4.1 的实例，并使用代码进行演示。
- (2) 教师根据课件，介绍 10.4.2 的实例，并使用代码进行演示。
- (3) 什么是中文分词？

中文分词是指将一个汉字序列切分成一个一个单独的词，也就是说将连续的字序列按照一定的规范重新组合成词序列的过程，其作用就是将用户输入的中文语句或语段拆成若干汉语词汇。

- (4) 教师根据课件，介绍 `jieba` 库的内容，并使用代码进行演示。

国内常用 `jieba` 实现中文分词，`jieba` 支持三种分词模式：精确模式、全模式、搜索引擎模式，其中精确模式是指试图将句子最精准地切开；全模式是指将句子中所有可以成词的词语都扫描出来；搜索引擎模式是指在精确模式的基础上对长词再次切分。

三、归纳总结，布置作业

- (1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要了解的 `jieba` 库。

- (2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

- (3) 使用博学谷系统下发课后作业。

第五课时

(`wordcloud` 库，`pygame` 库)

一、回顾上节课内容，继续讲解本节课的内容

- (1) 教师对学生们的疑问进行统一答疑。
- (2) 回顾上个课时所学习的内容，继续介绍本课时的内容。

上节课主要为大家演示了两个实例，并介绍了一个第三方库——`jieba`，接下来，本节课将继续为大家介绍 `wordcloud` 库以及 `pygame` 库的部分内容。

- (3) 明确学习目标

- 要求学生了解 wordcloud、pygame 库的使用

二、进行重点知识的讲解

- (1) 教师根据课件，引出词云这一概念。

词云是近些年在网络上兴起的一种图形化信息传递方式，网页浏览者只要短短一瞥即可接收到关键信息。程序在生成词云图之时会过滤掉大量的文本信息，将关键文本组成类似云朵的彩色图形。

- (2) 教师根据课件，介绍 wordcloud 库的用法，并使用代码进行演示。

利用 wordcloud 库生成词云主要包含三个步骤：利用 WordCloud 类的构造方法 WordCloud() 创建词云对象；利用 WordCloud 对象的 generate() 方法加载词云文本；利用 WordCloud 对象的 to_file() 方法生成词云。

- (3) 什么是 pygame？

pygame 是为开发 2D 游戏而设计的 Python 跨平台模块，开发人员利用 pygame 模块中定义的接口，可以方便快捷地实现诸如图形用户界面创建、图形和图像的绘制、用户键盘和鼠标操作的监听以及播放音频等游戏中常用的功能。

- (4) 教师根据课件，介绍 pygame 库的初始化和退出，并使用代码进行演示。

pygame 提供了初始化和退出的函数：init() 和 quit()，其中 init() 函数可以一次性初始化 pygame 的所有模块；quit() 函数可以卸载所有之前被初始化的 pygame 模块。

- (5) 教师根据课件，介绍创建游戏窗口的内容，并使用代码进行演示。

pygame 通过 display 子模块创建图形界面窗口，该子模块中提供了与窗口相关的函数：set_mode()、set_caption() 和 update()，其中 set_mode() 函数用于为游戏创建图形窗口；set_caption() 函数用于设置窗口标题；update() 函数用于刷新窗口，以显示修改后的新窗口。

- (6) 教师根据课件，介绍游戏循环的内容，并使用代码进行演示。

- (7) 教师根据课件提出的需求，引出游戏时钟这一概念。

一般情况下，电脑上一秒绘制 60 帧 (Frame) 便能够达到非常连续、高品质的动画效果，换言之，窗口中刷新图像的频率只要不低于每秒 60 帧，就能够达到我们对动画效果的预期。但循环体的执行频率非常高，远远超出每秒 60 帧。过高的帧率意味着超高的负荷，为了降低循环执行的频率，需要在程序中设置游

戏时钟。

(8) 教师根据课件，介绍设置游戏时钟的内容，并使用代码进行演示。

pygame 的 time 模块专门提供了一个 Clock 类，通过该类的 tick()方法可以方便地设置游戏循环的执行频率。

(9) 教师根据课件，介绍图形和文本绘制的内容，并使用代码进行演示。

pygame 中的图形窗口是一个 Surface 对象，在窗口中进行绘制实质上就是在 Surface 对象之上进行绘制。在 Surface 对象上绘制图形分为加载图片和绘制图片两个步骤；文本绘制实际上也是图片的叠加，只是在绘制之前需要先结合字体，将文本内容制作成图片。

(10) 教师根据课件，介绍 pygame 图形窗口的坐标体系。

坐标原点在游戏窗口的左上角；x 轴与水平方向平行，以向右为正；y 轴与垂直方向平行，以向下为正。

(11) 教师根据课件，介绍元素位置的描述与控制，并使用代码进行演示。

Rect 类用于描述、控制可见对象（文本、图片等）在 pygame 窗口中的位置。Surface 对象在窗口中的位置通过 blit()方法的参数 dest 确定，dest 可接收坐标元组(x,y)，亦可接收矩形对象。

三、归纳总结，布置作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要了解的第三方库：wordcloud 和 pygame(部分)。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第六课时

(pygame 库，实训案例)

一、回顾上节课内容，继续讲解本节课的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾上个课时所学习的内容，继续介绍本课时的内容。

上节课主要为大家介绍了第三方库 wordcloud 以及 pygame 库的部分内容，

接下来，本节课先继续为大家介绍 `pygame` 库的其它内容，再结合前面的知识点演示两个精彩的实例。

(3) 明确学习目标

- 要求学生了解 `pygame` 库的使用

二、进行重点知识的讲解

(1) 教师根据课件，介绍动态效果的内容，并使用代码进行演示。

实现动态效果的原理是文本或图片的更换、位置的改变以及屏幕的刷新。基础的动态效果包括三种：多次修改 `Surface` 对象绘制的位置并连续绘制刷新，实现移动效果；在同一位置绘制不同的 `Surface` 对象，实现动画效果；连续绘制不同 `Surface` 对象的同时，修改绘制的位置，实现移动的动画。

(2) 什么是事件？

程序开发中将玩家会对游戏进行的操作称为事件（Event），根据输入媒介的不同，游戏中的事件分为键盘事件、鼠标事件和手柄事件等等。

(3) 教师根据课件，介绍事件与事件处理的内容，并使用代码进行演示。

(4) 教师根据课件，介绍 10.6.1 的实例，并使用代码进行演示。

(5) 教师根据课件，介绍 10.6.2 的实例，并使用代码进行演示。

三、归纳总结，布置作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要掌握的 `pygame` 库。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

第七课时

（上机练习）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（考察知识点为 `time` 库）

	<p>形式：单独完成</p> <p>题目：</p> <p>请按照 10.4.2 的实例要求，利用 time、turtle 库绘制一个能随本地时间变化的钟表。</p> <p style="text-align: center;">第八课时 （上机练习）</p> <p>上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。</p> <p>上机一：（考察知识点为 pygame 库）</p> <p>形式：单独完成</p> <p>题目：</p> <p>请按照 10.6.2 的实例要求，利用 pygame 库实现一个反应力测试的小游戏，用挡板接住从屏幕顶部随机落下的小球。</p>
思考题 和习题	见教材第 10 章配套的习题
教 学 后 记	