

实用·好玩·参与

# 零基础入门学习



# Python

微课版

30 HOURS

30小时  
视频教程

© 小甲鱼 编著

# 目录

[前言](#)

[第1章 就这么愉快的开始吧](#)

[1.1 获得Python](#)

[1.2 从IDLE启动Python](#)

[1.3 失败的尝试](#)

[1.4 尝试点儿新的东西](#)

[1.5 为什么会这样](#)

[第2章 用Python设计第一个游戏](#)

[2.1 第一个小游戏](#)

[2.2 缩进](#)

[2.3 BIF](#)

[第3章 成为高手前必须知道的一些基础知识](#)

[3.1 变量](#)

[3.2 字符串](#)

[3.3 原始字符串](#)

[3.4 长字符串](#)

[3.5 改进我们的小游戏](#)

## [3.6 条件分支](#)

## [3.7 while循环](#)

## [3.8 引入外援](#)

## [3.9 闲聊数据类型](#)

### [3.9.1 整型](#)

### [3.9.2 浮点型](#)

### [3.9.3 布尔类型](#)

### [3.9.4 类型转换](#)

### [3.9.5 获得关于类型的信息](#)

## [3.10 常用操作符](#)

### [3.10.1 算术操作符](#)

### [3.10.2 优先级问题](#)

### [3.10.3 比较操作符](#)

### [3.10.4 逻辑操作符](#)

## [第4章 了不起的分支和循环](#)

### [4.1 分支和循环](#)

### [4.2 课堂小练习](#)

### [4.3 结果分析](#)

### [4.4 Python可以有效避免“悬挂else”](#)

## 4.5 条件表达式 (三元操作符)

## 4.6 断言

## 4.7 while循环语句

## 4.8 for循环语句

## 4.9 range()

## 4.10 break语句

## 4.11 continue语句

# 第5章 列表、元组合字符串

## 5.1 列表：一个“打了激素”的数组

### 5.1.1 创建列表

### 5.1.2 向列表添加元素

### 5.1.3 从列表中获取元素

### 5.1.4 从列表删除元素

### 5.1.5 列表分片

### 5.1.6 列表分片的进阶玩法

### 5.1.7 一些常用操作符

### 5.1.8 列表的小伙伴们

### 5.1.9 关于分片“拷贝”概念的补充

## 5.2 元组：戴上了枷锁的列表

### [5.2.1 创建和访问一个元组](#)

### [5.2.2 更新和删除元组](#)

## [5.3 字符串](#)

### [5.3.1 各种内置方法](#)

### [5.3.2 格式化](#)

## [5.4 序列](#)

# [第6章 函数](#)

## [6.1 Python的乐高积木](#)

### [6.1.1 创建和调用函数](#)

### [6.1.2 函数的参数](#)

### [6.1.3 函数的返回值](#)

## [6.2 灵活即强大](#)

### [6.2.1 形参和实参](#)

### [6.2.2 函数文档](#)

### [6.2.3 关键字参数](#)

### [6.2.4 默认参数](#)

### [6.2.5 收集参数](#)

## [6.3 我的地盘听我的](#)

### [6.3.1 函数和过程](#)

### [6.3.2 再谈谈返回值](#)

### [6.3.3 函数变量的作用域](#)

## [6.4 内嵌函数和闭包](#)

### [6.4.1 global关键字](#)

### [6.4.2 内嵌函数](#)

### [6.4.3 闭包 \(closure\)](#)

## [6.5 lambda表达式](#)

## [6.6 递归](#)

### [6.6.1 递归是“神马”](#)

### [6.6.2 写一个求阶乘的函数](#)

### [6.6.3 这帮小兔崽子](#)

### [6.6.4 汉诺塔](#)

## [第7章 字典和集合](#)

### [7.1 字典：当索引不好用时](#)

#### [7.1.1 创建和访问字典](#)

#### [7.1.2 各种内置方法](#)

### [7.2 集合：在我的世界里，你就是唯一](#)

#### [7.2.1 创建集合](#)

#### [7.2.2 访问集合](#)

### [7.2.3 不可变集合](#)

## [第8章 永久存储](#)

### [8.1 文件：因为懂你，所以永恒](#)

#### [8.1.1 打开文件](#)

#### [8.1.2 文件对象的方法](#)

#### [8.1.3 文件的关闭](#)

#### [8.1.4 文件的读取和定位](#)

#### [8.1.5 文件的写入](#)

#### [8.1.6 一个任务](#)

### [8.2 文件系统：介绍一个高大上的东西](#)

### [8.3 pickle：腌制一缸美味的泡菜](#)

## [第9章 异常处理](#)

### [9.1 你不可能总是对的](#)

### [9.2 try-except语句](#)

#### [9.2.1 针对不同异常设置多个except](#)

#### [9.2.2 对多个异常统一处理](#)

#### [9.2.3 捕获所有异常](#)

### [9.3 try-finally语句](#)

### [9.4 raise语句](#)

[9.5 丰富的else语句](#)

[9.6 简洁的with语句](#)

[第10章 图形用户界面入门](#)

[10.1 导入EasyGui](#)

[10.2 使用EasyGui](#)

[10.3 修改默认设置](#)

[第11章 类和对象](#)

[11.1 给大家介绍对象](#)

[11.2 对象 = 属性 + 方法](#)

[11.3 面向对象编程](#)

[11.3.1 self是什么](#)

[11.3.2 你听说过Python的魔法方法吗](#)

[11.3.3 公有和私有](#)

[11.4 继承](#)

[11.4.1 调用未绑定的父类方法](#)

[11.4.2 使用super函数](#)

[11.5 多重继承](#)

[11.6 组合](#)

[11.7 类、类对象和实例对象](#)



## [11.8 到底什么是绑定](#)

## [11.9 一些相关的BIF](#)

## [第12章 魔法方法](#)

### [12.1 构造和析构](#)

#### [12.1.1 \\_\\_init\\_\\_\(self\[, ...\]\)](#)

#### [12.1.2 \\_\\_new\\_\\_\(cls\[, ...\]\)](#)

#### [12.1.3 \\_\\_del\\_\\_\(self\)](#)

### [12.2 算术运算](#)

#### [12.2.1 算术操作符](#)

#### [12.2.2 反运算](#)

#### [12.2.3 增量赋值运算](#)

#### [12.2.4 一元操作符](#)

### [12.3 简单定制](#)

### [12.4 属性访问](#)

### [12.5 描述符 \(property的原理\)](#)

### [12.6 定制序列](#)

### [12.7 迭代器](#)

### [12.8 生成器 \(乱入\)](#)

## [第13章 模块](#)

## [13.1 模块就是程序](#)

## [13.2 命名空间](#)

## [13.3 导入模块](#)

## [13.4 `\_\_name\_\_ = '\_\_main\_\_'`](#)

## [13.5 搜索路径](#)

## [13.6 包](#)

## [13.7 像个极客一样去思考](#)

# [第14章 论一只爬虫的自我修养](#)

## [14.1 入门](#)

### [什么是编码](#)

## [14.2 实战](#)

### [14.2.1 下载一只猫](#)

### [14.2.2 翻译文本](#)

## [14.3 隐藏](#)

### [14.3.1 修改User-Agent](#)

### [14.3.2 延迟提交数据](#)

### [14.3.3 使用代理](#)

## [14.4 Beautiful Soup](#)

## [14.5 正则表达式](#)

欢迎访问：电子书学习和下载网站 (<https://www.shgis.com>)

文档名称：《零基础入门学习Python》小甲鱼.pdf

请登录 <https://shgis.com/post/4079.html> 下载完整文档。

手机端请扫码查看：

