在这个文档中:

按CTRL+1 键，可以变为“章”的格式。

按CTRL+2 ，可以变为“节”的格式。

按CTRL+3，可以变为“小节”的格式。

建议：

先把章列出来，等“章”确定差不多了。

然后章后的“节”标题，然后列“小节”标题。

这样一层层的列，可以保证“先总体，后细节”。

## [我的邮箱是wuhongwei@phei.com.cn](mailto:我的邮箱是wuhongwei@phei.com.cn)

以后咱们沟通时，重要的文档可以发邮箱，这样有一个备份。

建议

按照下面列出的四篇来组织内容，下一页是写作模板。

第1篇 准备篇，包括了解语言或软件、配置机器及搭建开发环境等；

第2篇 入门篇，介绍了基础操作；

第3篇 进阶篇，介绍了更高级的知识及应用；

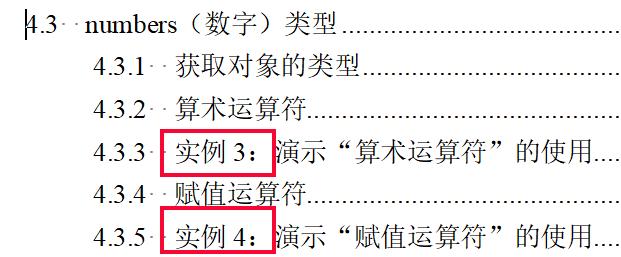
第4篇，是对前面知识的综合及实际应用。

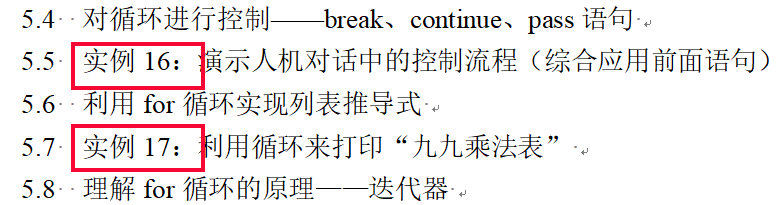
特色说明：

### （1）目录中有实例字眼

案例有三个种，

* 第一种是针对某个知识点的小案例（这在目录层次无法感知），就是大家正常用的那种。
* 第二种是分布在二、三标题，可以看到的案例，这种案例，共计不于30个。 如：





* 第三种是从章名就可以感知的案例。

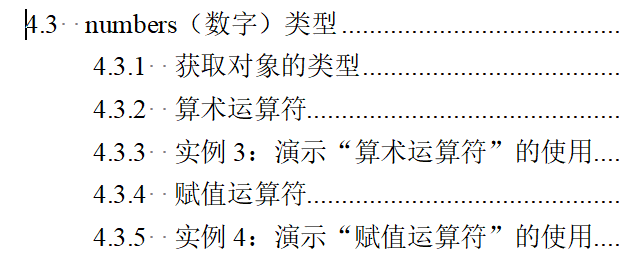
第11章 实战42：Python爬虫：批量采集股票数据，并持久化到excel中

第12章 实战43：自动化：读取excel数据文件，并用可视化分析

第13章 实战44：机器学习：从一组看似混乱的数据中找出y≈2x的规律

### （2）实例是什么回事

比如讲算述运算符，得先讲一些理论知识，然后通过一个实例。这时，可以把实例直接分出一个三级标题来。如下方这样。掌握这个原则，可以在目录中列出很多实例。如果理论很少，也可以直接列出一个实例的标题，以实例为主线，中间串插着讲一点理论。



（3）章标题，尽量体现出数据处理的逻辑，一般都是 数据的基本处理 →查询→索引→… 。

### （4）标题尽量用动词短语, 体现“解决什么问题”“能干什么”，类如下面这些标题，就像咱们正常说话一样即可。

* 准备Spark基础安装包
* 用Flume采集游戏日志到Kafka集群
* 新建Python文件，并编写代码
* 实现过滤并保存来访记录的函数
* 安装及使用opencv模块

### 避免只是一个名词的罗列，特别是纯英文的标题文字，如下面这些标题

请用下面这个目录来修改、增加目录。

第 1 篇 准备

本篇主要包括搭建环境等；

# 第1章 了解Web 3D

## 1.1 了解Web 3D的历史

## 1.2 准备预备知识

## 1.3 介绍3D编程

# 第2章 搭建开发环境

## 2.1 Reason版本及开发环境介绍

## 2.2 下载及安装VSCode

## 2.3 在VSCode中搭建Reason的开发环境

## 2.4 实例：写第一个Reason程序

# 第3章 搭建测试环境

## 3.1 介绍自动化测试

### 3.1.1 实例：单元测试

### 3.1.2 实例：集成测试

### 3.1.1 介绍端对端测试

## 3.2 介绍运行测试

### 3.2.1 通过log测试

### 3.2.2 通过WebGL-Inspector测试WebGL

### 3.2.3 通过Spector测试WebGL

## 3.3 介绍性能测试

### 3.3.1 使用Profile测试

### 3.3.2 使用Timeline测试

### 3.3.3 使用Console.profile和Performance测试

### 3.3.4 使用Memory测试

第 2 篇 入门

本篇主要介绍了基础知识；

# 第4章 函数式编程

## 4.1 学习函数式编程思想

## 4.2 实例：函数式编程的示例

# 第5章 函数反应式编程

## 5.1 学习函数反应式编程思想

## 5.2 学习Most库

### 5.2.1 介绍Most API

### 5.2.2 下载并安装Most

### 5.2.3 实例：函数反应式编程的示例

# 第6章 Reason语言基础

## 6.1 介绍Reason

## 6.2 介绍Reason语法

## 6.3 实例：应用了Reason主要特性的示例

# 第7章 WebGL基础

## 7.1 介绍渲染管线

## 7.2 坐标系变换

# 第8章 学习Reason-React

## 8.1 介绍Reason-React

## 8.2 学习基础知识

## 8.3 学习Redux

## 8.4 实例：Reason-React + Redux示例

第 3 篇 雏形

这一篇主要给出引擎和编辑器的基本架构。

# 第9章 快速写最小的Demo

## 9.1 需求分析

## 9.2 设计

## 9.3 实现

### 9.3.1 给出模块代码

### 9.3.2 给出每个模块的具体代码，具体分析

# 第10章 引擎和编辑器需求分析

## 10.1 引擎需求分析

## 10.2 编辑器需求分析

# 第11章 提出3D引擎

## 11.1 设计边界

## 11.2 提出数据

## 11.3 分析用户逻辑

## 11.4 识别领域模型

## 11.5 使用引擎

# 第12章 提出编辑器

## 12.1 设计边界

## 12.2 提出数据

### 12.2.1 引入Redux

## 12.3 引入引擎

## 12.4 使用编辑器

# 第13章 优化

## 13.1 测试性能

### 13.1.1 基准测试

### 13.1.2 分析

## 13.2 优化initAll

## 13.3 优化loop

### 13.3.1 缓存location

### 13.3.2 只设置一次相机uniform数据

### 13.3.3 缓存uniform数据

### 13.3.4 缓存program

### 13.3.5 使用vao

### 13.3.6 缓存vao

## 13.4 最终优化结果分析

# 第14章 改进引擎：提出GameObject+组件架构

## 14.1 分析现有架构

### 14.1.1 提出新需求

### 14.1.2 提出解决方案

## 14.2 初步设计

## 14.3 实现组件

### 14.3.1 设计

### 14.3.2 实现Transform组件

### 14.3.3 实现Geometry组件

### 14.3.4 实现BasicMaterial组件

### 14.3.5 实现MeshRenderer组件

## 14.4 修改GameObject

### 14.4.1 设计

### 14.4.2 实现

## 14.5 修改Render

### 14.5.1 设计

### 14.5.2 实现

## 14.6 实例：扩展组件

## 14.7 实现新需求

# 第15章 改进引擎：提出Data Oriented架构

## 15.1 分析现有架构

## 15.2 设计

## 15.3 实现

## 15.4 实例：扩展组件

## 15.5 实现新需求

# 第16章 改进引擎：改进Shader

## 16.1 分析现有架构

## 16.2 设计

## 16.3 实现

## 16.4 实例：扩展Shader

## 16.5 实现新需求

# 第17章 改进引擎：提出渲染队列

## 17.1 分析现有架构

## 17.2 设计

## 17.3 实现

## 17.4 实现新需求

# 第18章 改进引擎：支持各种应用场景

## 18.1 分析现有架构

## 18.2 设计

## 18.3 实现

## 18.4 实例：支持移动端

## 18.5 实现新需求

# 第19章 改进引擎：支持多线程

## 19.1 分析现有架构

## 19.2 设计

## 19.3 实现

## 19.4 实例：增加线程

# 第20章 改进引擎：其它改进

## 20.1 提出子State

## 20.2 使用函数式编程实现loop

## 20.3 契约检查

## 20.4 错误处理

# 第21章 改进编辑器

## 21.1 更新引擎

## 21.2 分析现有架构

## 21.3 重构文件夹

## 21.4 契约检查

## 21.5 错误处理

第 4篇 进阶

# 第22章 编辑器增加Redo/Undo功能

# 第23章 编辑器增加场景管理功能

## 23.1 需求分析

## 23.2 查看场景所有GameObject

## 23.3 操作GameObject

# 第24章 编辑器增加Inspector功能

## 24.1 需求分析

## 24.2 显示选中GameObject的所有组件

## 24.3 操作组件

# 第25章 编辑器增加控制台功能

# 第26章 增加相机

## 26.1 需求分析

## 26.2 引擎增加相机组件

## 26.3 更新编辑器

# 第27章 编辑器增加运行/停止功能

## 27.1 需求分析

## 27.2 初步设计

## 27.3 重构，提出新设计

## 27.4 实现

# 第28章 编辑器增加网格平面

# 第29章 编辑器增加Icon Gizmo

## 29.1 需求分析

## 29.2 实现IMGUI

## 29.3 实现相机Gizmo

# 第30章 支持事件

## 30.1 需求分析

## 30.2 引擎支持事件

## 30.3 编辑器支持事件

# 第31章 增加FlyCameraController

## 31.1 需求分析

## 31.2 引擎增加FlyCameraController组件

## 31.3 编辑器支持FlyCameraController组件

## 31.4 编辑器在Scene View中使用FlyCameraController组件

# 第32章 编辑器增加资产

## 32.1 需求分析

## 32.2 增加Material资产

## 32.3 重构：使用Recursive Type

## 32.2 实现

### 32.2.1 实现Asset Tree

### 32.2.2 实现Asset Children

### 32.2.2 实现Asset Inspector

# 第33章 支持光照

## 33.1 需求分析

## 33.2 更新引擎

### 33.2.1 引擎增加DirectionLight组件

### 33.2.2 引擎增加LightMaterial组件

## 33.3 更新编辑器

### 33.2.1 更新Inspector

### 33.2.2 更新Asset

### 33.2.3 更新Icon Gizmo

# 第34章 增加纹理

## 34.1 需求分析

## 34.2 更新引擎

### 34.2.1 引擎增加BasicSourceTexture组件

### 34.2.2 LightMaterial组件增加Diffuse Map

## 34.3 更新编辑器

### 34.3.1 更新Inspector

### 34.3.2 更新Asset

# 第35章 编辑器增加导入／导出功能

## 35.1 需求分析

## 35.2 更新引擎

### 35.2.1 增加Scene Graph文件(.wdb)

## 35.3 编辑器增加导出功能

### 35.3.1 增加导出场景wdb文件功能

### 35.3.1 增加导出包wpk文件功能

## 35.4 编辑器增加导入功能

### 35.4.1 增加导入模型wdb资产

### 35.4.2 增加导入包wpk文件

# 第36章 编辑器增加发布本地包功能

# 第37章 编辑器支持PWA

第 5篇 商业实战

这一篇主要讲，编辑器的应用实例。

# 第38章 制作3D静态场景

## 38.1 本章教学视频、素材、代码说明

## 38.2 准备美术素材

## 38.3 实战：使用编辑器制作3D静态场景

## 38.4 小结