

教案二 编写 Python 程序

单元标题	编写 Python 程序		单元教学学时	2 课时
授课场所	一体化实训室		授课形式	线上线下混合模式
在课程中的位置				
学习内容	<ol style="list-style-type: none"> 1. Python 代码规范 2. Python 程序开发流程 3. 创建 Python 项目 4. Python 程序的运行方式 			
学情分析	<p>2020 级软件技术专业大二的学生思维活跃，对软件开发有浓厚的学习兴趣，但学习主动性较差，综合运用知识的能力不足。该学生大一上学期已经开设《面向对象程序设计（Java）》，有一定的编程基础。已经掌握 Python 编程的特点、应用，能够搭建开发环境。但学生的代码编写习惯不好，本次课培养学生养成良好的代码编写习惯，让学生了解程序员应具备的职业素养。</p>			
教学目标	思政目标	知识目标	能力目标	
	<ol style="list-style-type: none"> 1. 养成良好的代码编写习惯； 2. 培养学生良好的职业素养 	<ol style="list-style-type: none"> 1. 掌握 Python 代码规范 2. 掌握 Python 程序开发流程 3. 掌握创建 Python 项目的方法 4. 掌握 Python 程序的运行方式 	<ol style="list-style-type: none"> 1. 能够熟练运用 Python 解释器运行 Python 程序 2. 能够创建 Python 项目并使用 Python 语句进行输入/输出 	
课程思政	融入知识点	<ol style="list-style-type: none"> 1. 计算机软件从业人员应当具备的职业素养 2. Python 编码规范 3. 程序开发流程 		
	融入方式	通过知识点融入思政		

	<p>思政元素</p> <p>1. 计算机软件从业人员应当具备的职业素养：通过介绍软件公司 Python 工程师的招聘条件，让学生了解 Python 程序开发规范的重要性，引导学生养成良好的职业素养。</p> <p>2. Python 编码规范：Python 的命名规范、缩进、注释。</p> <p>3. 程序开发流程：项目需求分析、概要设计、详细设计、测试、发布</p>										
	<p>思政资源</p> <p>网络资源：Python 工程师招聘信息</p> <p>视频：程序员应具备的职业素养</p> <p>课件：Python 编码规范</p> <p>课件、文档：程序开发流程</p>										
教学重点、难点	<p>教学重点：Python 编码规范、Python 程序的运行方式</p> <p>教学难点：Python 程序开发流程</p>										
教学方法与教学手段	<p>1. 教学方法：</p> <p>(1) 通过引例讲解 Python 程序构成</p> <p>(2) 示例说明 Python 对象和引用、标识符、变量、类等的命名规则</p> <p>(3) 示例说明 Python 语句缩进、注释</p> <p>(4) 讨论 Python 代码编写与 Java 语法的不同</p> <p>2. 辅助手段：</p> <p>(1) 多媒体演示</p> <p>(2) 视频讲解</p> <p>(3) 在线教学平台在线测试，大数据分析测试结果</p> <p>3. 对于重点和难点：通过案例讨论讲解、师生互动、在线测试、动画演示等解决和突破。</p>										
课前需掌握内容	<p>1. 什么是编译执行</p> <p>2. 什么是解释执行</p>										
教学内容设计	<p>任务一 Python 编码规范</p> <p>任务二 程序开发流程</p> <p>任务三 新建 Python 项目</p> <p>任务四 运行 Python 程序</p>										
教学资源	<table border="1"> <thead> <tr> <th>资源类型</th> <th>数量</th> </tr> </thead> <tbody> <tr> <td>教学设计/教案/课件/实训指导书/练习题</td> <td>8</td> </tr> <tr> <td>微课/视频/思政视频/音频答疑</td> <td>18</td> </tr> <tr> <td>思维导图/流程图</td> <td>2</td> </tr> <tr> <td>案例源码/推荐学习内容</td> <td>9</td> </tr> </tbody> </table>	资源类型	数量	教学设计/教案/课件/实训指导书/练习题	8	微课/视频/思政视频/音频答疑	18	思维导图/流程图	2	案例源码/推荐学习内容	9
	资源类型	数量									
	教学设计/教案/课件/实训指导书/练习题	8									
	微课/视频/思政视频/音频答疑	18									
思维导图/流程图	2										
案例源码/推荐学习内容	9										
课后拓展作业	<p>1. 在控制台输出个人简介</p> <p>2. 课程平台拓展作业</p>										

教学环节设计

一、课前准备

环节 用时	内容	教师活动	学生活动	教学方法 与手段
学习 微课 15 分钟	学习《Python 中的注释》、《Python 的命名规范》相关微课	1. 发布预习通知； 2. 上传微课	学习微课、 课件、教案	利用教学 平台完成 课前预习
课前 测试 10 分钟	<p>一、填空题</p> <p>1. Python 程序的运行方式有两种分别为_____和_____。</p> <p>2. Python 统一使用_____进行缩进。</p> <p>3. Python 中使用_____表示单行注释。</p> <p>4. Python 中使用_____和_____表示多行注释</p> <p>二、判断题</p> <p>1. Python 缩进可以 tab 和空格混用。 ()</p> <p>2. Python 可以使用三个单引号的多行注释，也可以使用三个双引号的多行注释。 ()</p> <p>3. 注释不是越多越好。对于一目了然的代码，不需要添加注释。 ()</p> <p>4. 变量名、类名取名必须有意义，严禁用单字母。 ()</p> <p>5. 变量名可以用系统关键字。 ()</p> <p>6. 变量名命名尽量体现变量的数据类型和具体意义。 ()</p> <p>7. 无论是解决四则运算的小规模程序，还是航天器使用的复杂控制程序，都遵循输入数据、处理数据和输出数据这一运算模式。 ()</p>	1. 发布测试题； 2. 查看测试结果； 3. 调整教学策略	使用手机 做题	1.利用教学 平台完成 课前测试； 2. 运用大 数据开展 学习行为 分析
课前 作业 10 分钟	搜索资源，了解更多计算机语言。	1. 发布课前作业； 2. 查看学生作业	各小组完 成作业内 容，将作业 上传至教 学平台。	利用教学 平台开展 作业分析

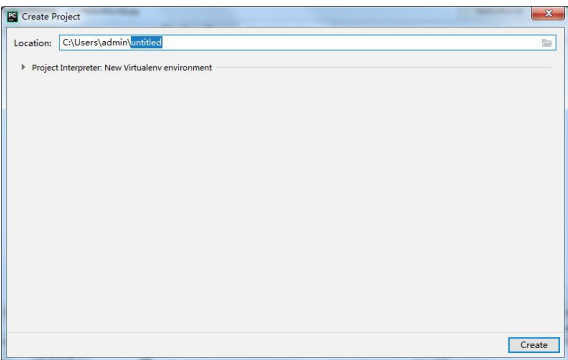
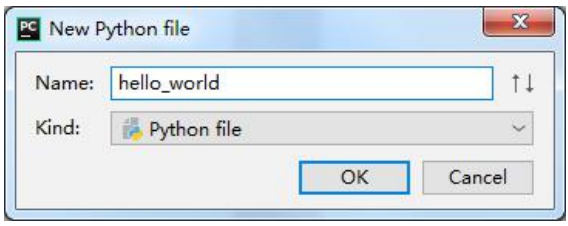

教学环节设计

二、课堂实施（2 课时）

环节 用时	内容	教师活动	学生活动	教学方法 与手段
导入 课程 5 分钟	构造情境，引出计算机软件从业人员应当具备的职业素养： 1. 师生互动，讨论应该具备哪些职业素养 2. 提出问题，编写程序为什么要规范？ 3. 编译执行与解释执行	1. 提出问题； 2. 总结学生的讨论结果 3. 介绍软件公司 Python 工程师的招聘条件	1. 分组讨论； 2. 在教学平台上发表答案； 3. 了解 Python 程序开发规范的重要性	1. 利用多媒体教学； 2. 利用教学平台开展讨论 3. 招聘信息
查看课 前预习 情况 5 分钟	要求学生上台讲解 Python 命名规范和 Java 的区别	1. 提出问题； 2. 总结学生操作情况。	1. 学生代表上台演示； 2. 其他学生观看并提出错误； 3. 改正错误	1. 利用教学平台课前分析测试情况； 2. 多媒体课件；
分析 问题 5 分钟	1. 如何使用 PyCharm 新建项目； 2. Python 的运行方式有哪些？	1. 演示开发环境； 2. 引出学习目的及重点、难点	1. 观看运行效果； 2. 小组讨论 Python 开发环境需要哪些软件； 3. 小组代表发言	1. 多媒体课件； 2. 运用 PyCharm 软件运行效果

环节 用时	内容	教师活动	学生活动	教学方法 与手段
学习 新知 15 分钟	<p align="center">任务一 Python 编码规范</p> <p>1. 代码编码</p> <p>(1) 国际惯例，文件编码和 Python 编码格式全部为 utf-8，例如：在 Python 代码的开头，要统一加上</p> <pre># -- coding: utf-8 --。</pre> <p>(2) Python 代码中，非 ascii 字符的字符串，请需添加 u 前缀</p> <p>(3) 若出现 Python 编码问题，可按照以下操作尝试解决：</p> <pre>import sys reload(sys) sys.setdefaultencoding('utf-8')</pre> <p>2. 命名规范</p> <p>(1) 包名、模块名、局部变量名、函数名 全小写+下划线式驼峰 示例：this_is_var</p> <p>(2) 全局变量 全大写+下划线式驼峰 示例：GLOBAL_VAR</p> <p>(3) 类名 首字母大写式驼峰 示例：ClassName()</p> <p>(4) 变量名命名 尽量体现变量的数据类型和具体意义</p> <p>注意： 变量名、类名取名必须有意义，严禁用单字母 变量名不要用系统关键字，如 dir type str 等等</p> <p>3. 空格</p> <p>统一使用 4 个空格进行缩进。绝对不要用 tab，也不要 tab 和空格混用。对于行连接的情况，建议使用 4 空格的悬挂式缩进。</p> <p>4. import 顺序</p> <p>(1) 标准库 (2) 第三方库 (3) 项目本身(之间用空行分隔)</p> <p>5. 注释</p> <p>(1) 单行注释（行注释）</p>	<p>1. 讲授知识点</p> <p>2. 提出问题： 常用的文件编码有哪些？</p> <p>3. 总结回答情况</p> <p>4. 播放视频“良好的编码习惯”，引导学生平时注意养成良好的职业素养</p>	<p>1. 认真听讲</p> <p>2. 思考问题</p> <p>3. 小组讨论</p> <p>4. 回答问题</p> <p>5. 观看视频，培养良好的职业素养</p>	<p>1. 多媒体教学；</p> <p>2. 分组讨论；</p> <p>3. 教学平台</p> <p>4. PyCharm 开发环境</p>

环节 用时	内容	教师活动	学生活动	教学方法 与手段
学习 新知 15 分钟	<p>Python 中使用#表示单行注释。单行注释可以作为单独的一行放在被注释代码行之上，也可以放在语句或表达式之后。</p> <p># 这是单行注释</p> <p>当单行注释作为单独的一行放在被注释代码行之上时，为了保证代码的可读性，建议在#后面添加一个空格，再添加注释内容。</p> <p>当单行注释放在语句或表达式之后时，同样为了保证代码的可读性，建议注释和语句（或注释和表达式）之间至少要有两个空格。</p> <p>(2) 多行注释</p> <pre>''' '''</pre> <p>这是使用三个单引号的多行注释</p> <pre>''' '''</pre> <p>这是使用三个双引号的多行注释</p> <pre>""" """</pre> <p>(3) 注意</p> <p>注释不是越多越好。对于一目了然的代码，不需要添加注释。</p> <p>对于复杂的操作，应该在操作开始前写上相应的注释。</p> <p>对于不是一目了然的代码，应该在代码之后添加注释。</p> <p>绝对不要描述代码。</p> <p>任务二 程序开发流程</p> <p>1. 程序开发流程</p> <p>为了保证程序与问题统一，也保证程序能长期稳定使用，人们将程序开发过程分为 6 个阶段，分别为：分析问题、划分边界、程序设计、编写程序、测试与调用、升级与维护。</p> <p>2.程序编写的基本方法</p> <p>无论是解决四则运算的小规模程序，还是航天器使用的复杂控制程序，都遵循输入数据、处理数据和输出数据这一运算模式。这一基础的运算模式形成了基本的程序编写方法——IPO 即输入、处理、输出。</p>	<p>1. 讲授知识点</p> <p>2. 提出问题： Python3. x 和 Python2. x 有哪些区别？</p> <p>3. 总结回答情况</p> <p>4. 提问： Python 应用在哪些领域</p> <p>5. 播放视频 -Python 的创始人 Guido von Rossum（吉多·范罗苏姆）的事迹，引导学生正确的人生观</p>	<p>1. 认真听讲</p> <p>2. 思考问题</p> <p>3. 小组讨论</p> <p>4. 回答问题</p> <p>5.观看视频，学习吉多的奉献精神</p>	<p>1.多媒体教学；</p> <p>2.分组讨论；</p> <p>3.教学平台</p> <p>4.PyCharm 开发环境</p>

环节 用时	内容	教师活动	学生活动	教学方法 与手段
<p>分组 实施</p> <p>15 分钟</p>	<p style="text-align: center;">任务三 创建 Python 项目</p> <p>(1) Creat New Project: 创建新项目。</p> <p>(2) Open: 打开已经存在的项目。</p> <p>(3) Check out from Version Control: 从版本控制中检出项目。</p> <p>(4) 创建项目。单击【Create New Project】进入【CreateProject】界面。这里设置项目存储路径为 D:\PythonDemo, 之后单击【Create】进入项目界面。</p>  <p>(5) 此时创建的项目是空项目, 之后还需要在项目中创建 Python 文件。选中项目名称, 单击鼠标右键, 在弹出的快捷菜单中选择【New】→【Python File】, 弹出“New Python file”窗口, 在该窗口的 Name 文本框中设置 Python 文件名为“hello_world”, 单击【OK】按钮后完成文件的创建。</p>  <p>(6) 在 hello_world.py 文件中输入下列代码:</p> <pre>print("Hello World!")</pre> <p>(7) 右键单击 HelloWorld.py 文件, 在弹出的快捷菜单中选择【Run 'hello_world'】运行程序。</p> 	<p>1. 巡回指导, 解疑答难;</p> <p>2. 容易出现的问题:</p> <p>Check out from Version Control: 从版本控制中检出项目</p>	<p>1. 完成创建 Python 项目;</p> <p>2. 小组为单位进行讨论 PyCharm 的用法</p>	<p>1. 运用多媒体课件和视频;</p> <p>2. 利用 API 文档;</p> <p>3. PyCharm 软件、Python3.7</p> <p>4. 分组完成程序</p>

环节 用时	内容	教师活动	学生活动	教学方法 与手段
分组 实施 20 分钟	<p style="text-align: center;">任务四 运行 Python 程序</p> <p>1. Python 程序的运行方式 Python 程序的运行方式有两种分别为交互式和文件式。</p> <p>2. 如何运行 Python 程序 (1) 根据提示录入个人信息，并根据输入将信息打印到终端。</p> <p>交互式</p> <pre>>>> name = input("请输入姓名：") 请输入姓名：李强 >>> spec = input("请输入专业：") 请输入专业：计算机 >>> cls = input("请输入班级：") 请输入班级：3 >>> print("%s 同学来自%s 专业%s 班。" %(name, spec, cls)) 李强同学来自计算机专业 3 班。</pre> <p>文件式</p> <pre>name = input("请输入姓名：") spec = input("请输入专业：") cls = input("请输入班级：") print("%s 同学来自%s 专业%s 班。"%(name, spec, cls))</pre> <p>(2) 计算原型面积</p> <p>交互式</p> <pre>r = 5 s = 3.14 * r * r >>> print(s)</pre> <p>文件式</p> <pre>r = 5 s = 3.14 * r * r print(s)</pre>	<p>1. 巡回指导，解疑答难；</p> <p>2. 容易出现的问题： print("%s 同学来自%s 专业%s 班。"%(name, spec, cls))</p>	<p>1. 学习交互式和文件式运行的特点；</p> <p>2. 小组为单位进行完成案例的运行</p>	<p>1. 运用多媒体课件和视频；</p> <p>2. 利用 API 文档；</p> <p>3. PyCharm 软件、Python3.7</p> <p>4. 分组完成程序</p>
小组 汇报与 总结 10 分钟	<p>各小组汇报代码编写及运行调试情况</p> <p>教师总结</p>	<p>1. 点评小组作品</p> <p>2. 总结本节课</p>	<p>各小组汇报展示、各小组互相评分</p>	<p>1. 利用 PyCharm 软件</p> <p>2. 多媒体课件</p>

教学环节设计

三、课后拓展

环节 用时	内容	教师活动	学生活动	教学方法 与手段
学习 PyCharm 的功能 15 分钟	项目组根据老师的点评，继续搭建开发环境并学习 PyCharm 软件的功能和用法。	在线指导	练习 PyCharm 软件用法 巩固知识	1. 利用教学平台作业分析功能； 2. PyCharm 软件编写代码
学习网络资源 10 分钟	教师推荐优秀的网络学习资源，如：国家数字化学习资源中心、慕课网等，拓宽学生视野。	推荐优秀的网络学习资源	根据需要学习网络资源	利用网络资源拓宽学生视野
拓展 作业 15 分钟	分布使用 Python 解释器和 PyCharm 输出“HelloWorld”	发布作业 解答问题	查找资料 编写代码 运行调试 上传平台	1. 利用模拟实训平台提交代码； 2. 小组讨论完成作业
大数据 开发 Python、 计算机 视觉 1+x 证书 20 分钟	学习《大数据开发（Python）》、《计算机视觉》1+x 证书考试相关资料 项目一 图像采集	提供《大数据开发（Python）》、《计算机视觉》1+x 证书考试相关资料、辅导常见问题	学习提供《大数据开发（Python）》、《计算机视觉》1+x 证书考试相关资料	1. 利用教学平台； 2. 网络资源提供学习内容

任务一

(Python 编码规范)

一、教师通过情景引出本次课的任务

1. 教师对学生们的疑问进行统一答疑。
2. 回顾总结任务二内容，继续介绍本次任务的内容。

任务二我们学习 Python 运行方式：文件式和交互式。良好的编写习惯是程序员的基本素养，本次任务我们将学习 Python 编码规范。

3. 明确学习目标

- (1) 掌握 Python 的代码编码
- (2) 掌握 Python 的命名规范
- (3) 掌握 Python 中的缩进
- (4) 掌握 Python 的注释

二、进行重点知识的讲解

1. 代码编码

(1) 国际惯例，文件编码和 Python 编码格式全部为 utf-8，例如：在 Python 代码的开头，要统一加上

```
# -- coding: utf-8 --。
```

(2) Python 代码中，非 ascii 字符的字符串，请需添加 u 前缀

(3) 若出现 Python 编码问题，可按照以下操作尝试解决：

```
import sys
reload(sys)
sys.setdefaultencoding('utf-8')
```

2. 命名规范

(1) 包名、模块名、局部变量名、函数名
全小写+下划线式驼峰 示例：this_is_var

(2) 全局变量
全大写+下划线式驼峰 示例：GLOBAL_VAR

(3) 类名
首字母大写式驼峰 示例：ClassName()

(6) 变量名命名
尽量体现变量的数据类型和具体意义

注意：

变量名、类名取名必须有意义，严禁用单字母

变量名不要用系统关键字，如 dir type str 等等

3. 空格

统一使用 4 个空格进行缩进。绝对不要用 tab，也不要 tab 和空格混用。对于

行连接的情况，建议使用 4 空格的悬挂式缩进。

4. import 顺序

(1) 标准库 (2) 第三方库 (3) 项目本身(之间用空行分隔)

5. 注释

(1) 单行注释（行注释）

Python 中使用#表示单行注释。单行注释可以作为单独的一行放在被注释代码行之上，也可以放在语句或表达式之后。

```
# 这是单行注释
```

当单行注释作为单独的一行放在被注释代码行之上时，为了保证代码的可读性，建议在#后面添加一个空格，再添加注释内容。

当单行注释放在语句或表达式之后时，同样为了保证代码的可读性，建议注释和语句（或注释和表达式）之间至少要有两个空格。

(2) 多行注释

```
'''
```

这是使用三个单引号的多行注释

```
'''
```

```
"""
```

这是使用三个双引号的多行注释

```
"""
```

(3) 注意

注释不是越多越好。对于一目了然的代码，不需要添加注释。

对于复杂的操作，应该在操作开始前写上相应的注释。

对于不是一目了然的代码，应该在代码之后添加注释。

绝对不要描述代码。一般阅读代码的人都了解 Python 的语法，只是不知道代码要干什么。

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。
2. 布置随堂练习，检测学生掌握情况。

课程思政内容：

良好的编码习惯的重要性：

良好的编码习惯像黑夜中的一盏明灯，指引着迷路的开发者安全靠岸。良好的代码是可预测的，是易于调试、扩展和测试的。良好的编码习惯能够提高你同事的工作效率，同时让你的代码库从整体上给人一种愉快的阅读体验。

- (1) 简洁地给方法和变量命名
- (2) 尽可能减少使用全局变量
- (3) 编写可预测的代码
- (4) 编写可重用的代码
- (5) 写单元测试

任务二

(程序开发流程)

一、回顾上节课内容，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. 回顾总结任务三内容，继续介绍本次任务的内容。

任务三我们学习了 Python 编码规范，了解程序开发流程对后期的 Python 开发有重要的作用，本次任务我们将学习程序开发流程。

3. 明确学习目标
 - (1) 了解程序编写的基本方法
 - (2) 理解程序开发流程
 - (3) 掌握在控制台中输出信息

二、进行重点知识的讲解

1. 程序开发流程

为了保证程序与问题统一，也保证程序能长期稳定使用，人们将程序开发过程分为 6 个阶段，分别为：分析问题、划分边界、程序设计、编写程序、测试与调用、升级与维护。

(1) 首先制定项目计划，最初计划是里程碑性质的。可以先按瀑布模型设置，里程碑点主要为需求评审、设计评审、经过代码开发和单元测试后进行集成测试、部署上线是一个很重要的里程碑，一般用户会期望系统何时能使用进入试运行期(也可以称为用户测试阶段)

(2) 需求开发阶段：怎么样写好需求很关键，如何学会进行需求开发可以去看下经典的《需求工程》这个翻译的书。一般分为《用户需求说明书》和《需求规格说明书》，小项目可以写一个《需求分析报告》，《用户需求说明书》是用用户的语言进行描述，让用户和开发团队对于需求的达成一致的理解，《需求规格说明书》，则是对用户需求的分析，形成系统要具有的功能，这个是真正提供用户可交互操作的文档，也就是后期设计和代码开发的重要基线。

(3) 系统设计阶段：系统总体架构，结合用户对系统环境、开发语言以及运行的网络硬件等要求，确定开发工具等，对应用系统关系进行架构性设计，通过需求阶段对用户的分析归类，用图的方式描述出用户和各子系统或模块的全局视图，以及和其他系统的关系。也就是搞清楚系统的边界问题。

概要设计中除了高层架构设计，还需要设计网络拓扑图，以及系统部署图。概要设计比较重要的还有就是子系统、模块进行合理的划分。模块的名称很大程度上会成为用户的主要菜单，如何用用户的角度去取比较清楚的子系统和模块是很重要的。

(4) 代码开发和单元测试阶段：这个阶段一般来说需要改进瀑布模型，类似迭代开发，把模块进行合理划分，把项目总体计划的代码开发测试阶段划分为多个时间段，每个时间段都包括代码开发、单元测试和集成测试，这个阶段还需要对需求变更进行跟踪控制，如果需求有变更，那么要把需求文档、设计文档都重新跟上。

(5) 测试工作，测试是项目的很重要的环节，怎么测试，怎么准确测试，怎么有效测试，怎么覆盖测试，时间、人手、经验各个方面都会有制约。高级测试人员能够分析系统各测试要点，在需求、设计阶段都要参与，提早了解如何去测试，能写出测试用例。

(6) 文档工作，文档在项目开发中也占有重要位置，代码开发其实也需要文档，代码是成果，代码注释是成果，模块开发卷宗也是重要的成果，因为程序员在开发时候的逻辑是怎么样的，对于今后查问题很有作用。

2. 程序编写的基本方法

无论是解决四则运算的小规模程序，还是航天器使用的复杂控制程序，都遵循输入数据、处理数据和输出数据这一运算模式。这一基础的运算模式形成了基本的程序编写方法——IPO 即输入、处理、输出。

3. 在控制台输出个人信息

教师提出要求，学生根据要求在控制台输出个人信息、爱好等。各小组演示运行结果。

```
name = 'Jim'
age=18
address = '3000, 2nd Ring Road'
love='Table Tennis'
print("name: {}\nage: {}\naddress: {}\nlove: {}".format(name, age, address,love))
```

运行结果：

```
D:\pythonProject1\venv\Scripts\python.exe D:/pythonProject1/hello.py
name:Jim
age:18
address:3000, 2nd Ring Road
love:Table Tennis
```

课程思政：

访谈软件企业优秀员工，了解企业文化、程序开发流程，为以后进行软件开发工作奠定基础。

任务三

（创建 Python 项目）

一、构造情境，引出计算机软件从业人员应当具备的职业素养

1. 师生互动，讨论应该具备哪些职业素养

通过介绍软件公司 Python 工程师的招聘条件，让学生了解 Python 程序开发规范的重要性，引导学生养成良好的职业素养。

2. 提出问题，编写程序为什么要规范？

3. 编译执行与解释执行

高级语言被广泛应用于众多领域，但使用高级语言编写的程序无法被计算机识别，在执行之前需要先将高级语言代码翻译成机器语言代码。根据不同的翻译方式，执行分位编译执行和解释执行。

（1）编译执行

编译执行是指通过编译程序（也称为编译器）将源代码（source code）一次性编译成目标程序，再由计算机运行目标程序的过程，其中源代码指由高级语言编写的代码

（2）解释执行

解释执行与编译执行主要的区别是翻译时的解释程序不产生目标代码，且解释器在翻译源代码的同时执行中间代码。

4. 明确学习目标

要求学生掌握创建 Python 项目的步骤

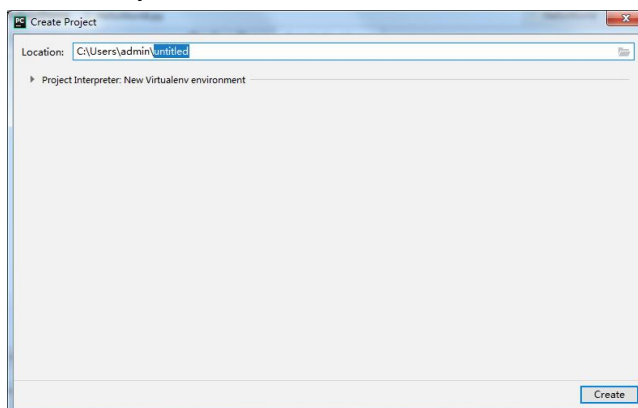
二、进行重点知识的讲解

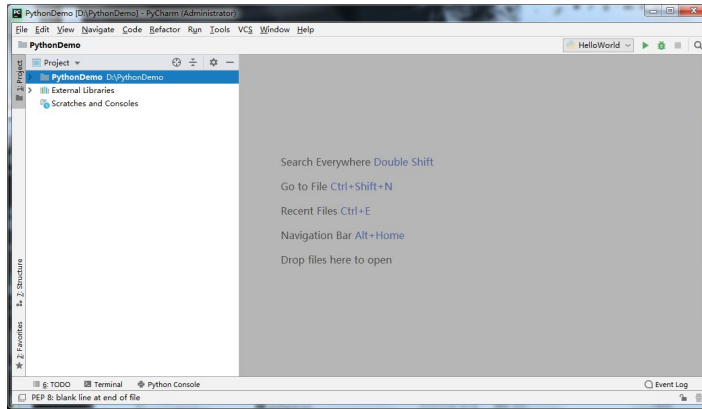
（1）Creat New Project: 创建新项目。

（2）Open: 打开已经存在的项目。

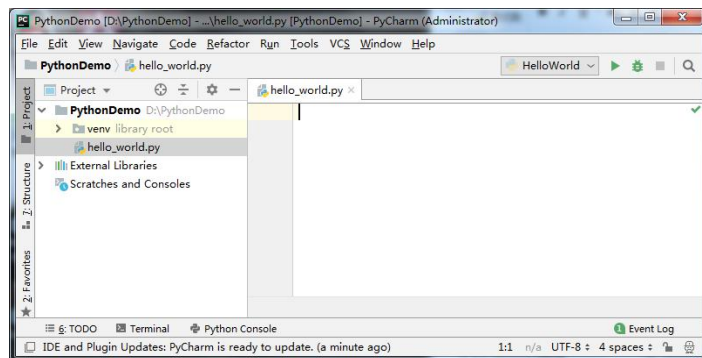
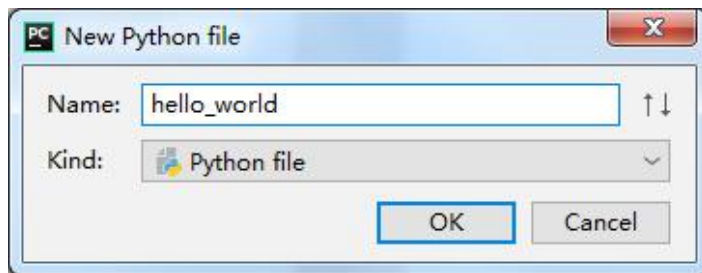
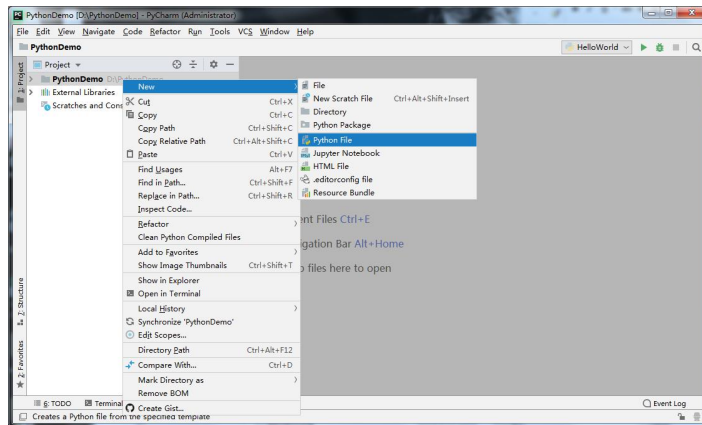
（3）Check out from Version Control: 从版本控制中检出项目。

（4）创建项目。单击【Create New Project】进入【CreateProject】界面。这里设置项目存储路径为 D:\PythonDemo，之后单击【Create】进入项目界面。





(5) 此时创建的项目是空项目，之后还需要在项目中创建 Python 文件。选中项目名称，单击鼠标右键，在弹出的快捷菜单中选择【New】→【Python File】，弹出“New Python file”窗口，在该窗口的 Name 文本框中设置 Python 文件名为“hello_world”，单击【OK】按钮后完成文件的创建。



(6) 在 `hello_world.py` 文件中输入下列代码:

```
print("Hello World!")
```

(7) 右键单击 `HelloWorld.py` 文件, 在弹出的快捷菜单中选择【Run 'hello_world'】运行程序。



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。
2. 布置随堂练习, 检查学生掌握情况。

根据随堂练习资源, 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。

课程思政:

通过介绍软件公司 Python 工程师的招聘条件, 让学生了解 Python 程序开发规范的重要性, 引导学生养成良好的职业素养:

1. 熟悉敏捷开发;
2. 具备良好的团队合作、沟通能力和抗压能力, 具备良好的学习钻研能力;
3. 熟悉常用的数据结构和算法, 有良好的编程习惯;
4. 有责任心, 踏实勤奋, 工作态度认真, 积极向上;
5. 熟悉软件开发的流程, 具有项目交付经验和能力;
6. 有良好的质量意识及代码规范, 能完成设计优化及性能优化;
7. 创新意识, 对技术研究有热情, 敢于面对各种技术挑战。

任务四

(运行 Python 程序)

一、回顾上节课内容，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. 回顾总结任务一内容，继续介绍本次任务的内容。

上机课介绍了新建一个 Python 项目，本此任务一起学习如何运行 Python 程序。

3. 明确学习目标
 - (1) 理解 Python 程序的运行方式
 - (2) 掌握交互式和文件式运行的特点

二、进行重点知识的讲解

1. Python 程序的运行方式

Python 程序的运行方式有两种分别为交互式和文件式。交互式指 Python 解释器逐行接收 Python 代码并批量即时响应；文件式也称批量式，指先将 Python 代码保存在文件中，再启动 Python 解释器批量解释代码。

2. 如何运行 Python 程序

教师根据教材内容 1.3.3 中的示例代码，演示 Python 代码的运行。

- (1) 计算圆的面积

根据圆的半径计算圆的面积。

交互式

```
r = 5
s = 3.14 * r * r
>>> print(s)
```

78.5

文件式

```
r = 5
s = 3.14 * r * r
print(s)
```

- (2) 信息录入

根据提示录入个人信息，并根据输入将信息打印到终端。

交互式

```
>>> name = input("请输入姓名: ")
```

请输入姓名: 李强

```
>>> spec = input("请输入专业: ")
```

请输入专业: 计算机

```
>>> cls = input("请输入班级: ")
请输入班级: 3
>>> print("%s 同学来自%s 专业%s 班。"%(name,spec,cls))
李强同学来自计算机专业 3 班。
文件式
name = input("请输入姓名: ")
spec = input("请输入专业: ")
cls = input("请输入班级: ")
print("%s 同学来自%s 专业%s 班。"%(name, spec, cls))
```

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。
2. 布置随堂练习，检查学生掌握情况。

根据随堂练习资源，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。

练习题：

- (1) Python 程序的运行方式有两种分别为_____和_____。
- (2) Python 统一使用_____进行缩进。
- (3) Python 中使用_____表示单行注释。
- (4) Python 中使用_____和_____表示多行注释
- (5) Python 缩进可以 tab 和空格混用。 ()
- (6) Python 可以使用三个单引号的多行注释，也可以使用三个双引号的多行注释。 ()
- (7) 注释不是越多越好。对于一目了然的代码，不需要添加注释。 ()
- (8) 变量名、类名取名必须有意义，严禁用单字母。 ()
- (9) 变量名可以用系统关键字。 ()
- (10) 变量名命名尽量体现变量的数据类型和具体意义 ()

教学 内容 总结	三、归纳总结，布置课后作业		
	任务	知识目标	思政元素
	任务一 新建 Python 项目	掌握创建 Python 项目的步骤	1. 师生互动, 讨论应该具备哪些职业素养 2. 通过介绍软件公司 Python 工程师的招聘条件, 让学生了解 Python 程序开发规范的重要性, 引导学生养成良好的职业素养。
	任务二 运行 Python 程序	(1) 理解 Python 程序的运行方式 (2) 掌握交互式和文件式运行的特点	
	任务三 Python 编码规范	(1) 掌握 Python 的代码编码 (2) 掌握 Python 的命名规范 (3) 掌握 Python 中的缩进 (4) 掌握 Python 的注释	良好的编码习惯的对软件开发非常重要, 引导学生平时注意养成良好的编码习惯。
任务四 程序开发流程	(1) 了解程序编写的基本方法 (2) 理解程序开发流程 (3) 掌握在控制台中输出信息	访谈软件企业优秀员工, 了解企业文化、程序开发流程, 为以后进行软件开发工作奠定基础。	
教学 反思	经验	1. 多门课程联动, 深入挖掘思政元素 2. 整合教学案例, 有效融入课程思政 3. 融合多种教学手段和方法, 高效完成教学目标	
	不足	1. 教师课程设计能力及课程思政能力有待进一步提高 2. 课程教学材料及思政资料需要进一步完善	
拓展 作业	1. 在控制台输出个人简介 2. 课程平台拓展作业		