教案六 定义与调用函数

<table>
<tr><td>单元标题</td><td>定义与调用函数</td><td>单元教学
学时</td><td>8 课时</td></tr>
<tr><td>授课场所</td><td>一体化实训室</td><td>授课形式</td><td>线上线下混合模式</td></tr>
<tr><td>在课程中的位置</td><td colspan="3"> 
</td></tr>
<tr><td>学习内容</td><td colspan="3">1.函数的定义与调用
2.函数的参数传递方式
3.局部变量和全局变量的使用
4.匿名函数与递归函数的使用</td></tr>
<tr><td>学情分析</td><td colspan="3">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。学生已经掌握了流程控制语句、模块与包的用法，能够进行简单的编程。通过课前测试、调查问卷大数据分析，学生对 Python 中的函数的定义与调用掌握不好，部分学生沟通交流与团队合作能力较弱。</td></tr>
<tr><td rowspan="2">教学
目标</td><td>思政目标</td><td>知识目标</td><td>能力目标</td></tr>
<tr><td>1. 培养学生良好的职业素养
2. 培养学生沟通交流与团队合作能力</td><td>1.掌握函数的定义与调用
2.掌握函数的参数传递方式
3.掌握局部变量和全局变量
4.熟悉匿名函数与递归函数
5.了解常用的内置函数</td><td>1. 能够熟练定义函数与调用函数
2. 能够熟练运用函数解决实际问题，简化程序结构</td></tr>
<tr><td rowspan="3">课程思政</td><td>融入
知识点</td><td colspan="2">1. 通过函数的调用引出分工与合作，融入沟通交流与团队合作能力
2.个人与整体的关系</td></tr>
<tr><td>融入方式</td><td colspan="2">通过知识点融入思政</td></tr>
<tr><td>思政元素</td><td colspan="2">1. 日常生活中，要完成一件复杂的功能，我们总是习惯把“大功能”分解为多个“小功能”以实现。在编程的世界里，“功能”可称呼为“函数”，因此“函数”其实就是一段实现了某种功能的代码，并且可以供其它代码调用；
2.通过多人合作完成项目引出分工与合作，培养学生沟通交流与团队合作能力；
3. 正确认识个人与整体的关系。</td></tr>
</table>

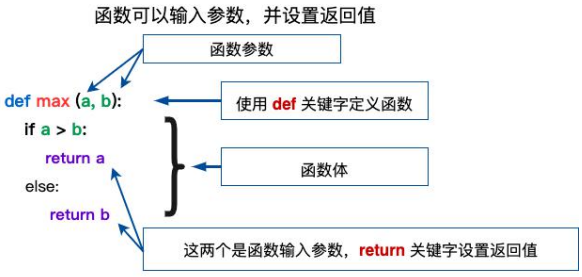
<table>
<tr><td></td><td>思政资源</td><td>视频：沟通能力的培养
视频：团队合作能力的培养
文章：正确认识个人与整体的关系</td></tr>
<tr><td>教学重点、难点</td><td colspan="2">教学重点：位置参数、关键字参数、默认参数
教学难点：全局变量、匿名函数、递归函数</td></tr>
<tr><td>教学方法与教学手段</td><td colspan="2">1.教学方法：
（1）课前让学生学习微课、互联网查找资料熟悉知识点；通过在线测试了解学生对知识点的掌握情况，调整教学策略；
（2）采用教学做一体的教学模式，运用翻转课堂模式，以任务驱动为载体，将公司管理机制运用到教学管理中；
（3）理论联系实际，引入“简易四则运算器”、“汉诺塔”、“斐波那契数列”等案例，充分调动学生的积极性和主动性，夯实学生的基础知识，培养学生探究性学习的能力；
（4）分组讨论“如何实现整数的阶乘”，小组代表汇报；
（5）实例讲解定义函数、调用函数、函数的参数传递、函数的递归调用。
2.辅助手段：
（1）多媒体演示、原理动画展示函数的递归调用、斐波那契数列；
（2）视频讲解；
（3）在线教学平台在线测试，大数据分析测试结果；
（4）利用编程软件 Python3.8、PyCharm 编写调试代码。
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、分析流程图等解决和突破。</td></tr>
<tr><td>课前需掌握内容</td><td colspan="2">Python 中的参数的作用与分类、返回值的含义</td></tr>
<tr><td>教学内容设计</td><td colspan="2">任务一　实现四则运算器
任务二　运用参数传递数据
任务三　定义局部和全局变量
任务四　用递归函数输出斐波那契数列</td></tr>
<tr><td>教学资源</td><td colspan="2"><table>
<tr><th>资源类型</th><th>数量</th></tr>
<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>65</td></tr>
<tr><td>思维导图/流程图</td><td>6</td></tr>
<tr><td>案例源码/推荐学习内容</td><td>4</td></tr>
</table></td></tr>
<tr><td>课后拓展作业</td><td colspan="2">1.登录系统账号检测
2.输出汉诺塔移动过程</td></tr>
</table>

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《函数的定义》、《函数的调用》相关微课	1.将微课、课件、教案上传到教学平台； 2.发布预习通知	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	1. 下列关于函数参数的说法中错误的是。（ ） A. 若无法确定需要传入函数的参数个数,可以为函教设置不定长参数 B. 当使用关键字参数传递实参时,需要为实参关联形参 C. 定义函数时可以为参数设置默认值 D. 不定长参数*args 可传递不定数量的关联形参名的实参 2. 阅读下面程序: num one =1 2 def sum(num two) : global num one num one = 90 return num one + num two print (sum(10)) 运行代码，输出结果是( ) A. 102 B. 100 C. 22 D.12 3.函数可以提高代码的复用性。 ( ) 4.全局变量在所有的函教中都可以访问。 ( ) 5.函数的位置参教有严格的位置关系。 ( ) 6.函数中的跳认参教不能传递实际参数。 ( )	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	查找资料，了解 Python 常用的内置函数	1.发布课前作业； 2.查看学生作业、了解学生对知识点运用情况	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（8 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	教师学提问，如果程序较复杂，并且程序中重复执行某一功能，如何提高代码利用率，使程序结构更加清晰？ 引出函数的意义。函数被指封装起来的、实现某种功能的一段代码，它可以被其他函数调用。当程序实现的功能较为复杂时，开发人员通常会提取其中的功能性代码模块化为一个函数，提高代码复用性、降低代码冗余、使程序结构更加清晰。	1. 提出问题； 2. 启发引导学生思考： 3. 引出学习目的及重点、难点	结合生活实际，积极思考踊跃回答	1. 多媒体课件； 2. 教学平台； 3. PyCharm 软件
查看课前预习情况 10 分钟	1. 小组展示课前作业，教师评价 2. 展示课前测试情况	1. 教师评价 2.讲解错误率高的题目	1. 小组展示作业情况 2. 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 10 分钟	播放视频：如何提升团队合作能力 如何提升沟通能力 函数被指封装起来的、实现某种功能的一段代码，它可以被其他函数调用。当程序实现的功能较为复杂时，开发人员通常会提取其中的功能性代码模块化为一个函数，提高代码复用性、降低代码冗余、使程序结构更加清晰。	1.播放视频 2. 帮助学生理解函数的意义	1. 观看运行效果； 2. 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
 	
45 分钟
	任务一 实现四则运算器
	1．什么是函数
	函数是组织好的、可重复使用的、用于实现单一或相关联功能的代码段，通过函数的名称表示和调用。函数也可以看作是一段有名字的子程序，可以在需要的地方使用函数名调用执行。
	2．函数的定义
	Python 使用 def 关键定义函数，其语法格式如下：
	def 函数名([参数列表]):
	函数体
	[return 语句]。
	3．函数的调用
	函数定义好之后不会立即执行，直到被程序调用时才会生效。通过函数名(参数列表)方式调用。
	def my_absolute(x):
	if x>=0:
	print(x)
	else:
	print(-x)
	my_absolute(-10.3)
	
	课程思政：
	通过函数的意义引出：
	1.日常生活中，要完成一件复杂的功能，我们总是习惯把“大功能”分解为多个“小功能”以实现。在编程的世界里，“功能”可称呼为“函数”，因此“函数”其实就是一段实现了某种功能的代码，并且可以供其它代码调用；
	2. 通过多人合作完成项目引出分工与合作，培养学生沟通交流与团队合作能力。
		1. 讲授知识点；
		2. 提出问题：如何定义与调用函数？总结回答情况；
		3. 播放视频：“如何提高团队合作能力？”、“如何提高沟通能力？”
			1. 认真听讲
			2. 思考问题
			3. 小组讨论
			4. 提交讨论结果
			5. 观看视频
				1.教学平台；
				2.多媒体课件；
				3.PyCharm 软件；
				4.Python3.7 编辑软件；
				5. 分组讨论；

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 90 分钟	**任务二 运用参数传递数据** **1.什么是函数参数传递，为什么用函数参数** 函数的参数传递是指将实参传递给形参的过程，Python 中的函数支持以多种方式传递参数，包括位置传递、关键字传递、默认值传递、包裹传递、解包裹传递以及混合传递。  **2.参数的位置传递** 调用函数时，默认按照位置顺序将对应的实参传递给形参，即将第一个实参分配给第一个形参，第二个实参分配给第二个形参，以此类推。 **3.参数的关键字传递** 关键字传递通过“形参变量名=实参”的形式将形参与实参关联，根据形参的名称进行参数传递，它允许实参和形参的顺序不一致。 **4.参数的默认值传递** 函数在定义时可以给每个参数指定默认值，基本形式为：函数名（参数=默认值），这样在调用时既可以给带有默认值的参数从新赋值，也可以省略相应的实参，使用参数的默认值。 **5.不定长参数传递** 格式：def 函数名([formal_args,]*args,**kwargs): （1）函数传入的参数个数会优先匹配 formal_args 参数的个数 （2）如果传入的参数没有指定名称，那么*args 会以元组的形式存放多余的参数 （3）如果传入的参数指定了名称，如 m=1，那么**kwargs 会以字典的形式存放这些参数。	1.讲授知识点； 2.提出问题：根据任务一的四则运算器运行效果，让学生思考有哪些不足； 3.发起讨论话题：函数的参数传递有几种，各有什么特点	1.认真听讲 2.思考问题 3.小组讨论 4.提交讨论结果	1.教学平台； 2.多媒体课件； 3.PyCharm 软件； 4.Python3.7 编辑软件； 5.分组讨论；

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 45 分钟	**任务三 定义局部和全局变量** **1. 函数返回值的作用** 函数中的 renturn 语句是可选项，可以出现在函数体的任何位置，它的作用是结束当前函数，将程序返回到函数被调用的位置继续执行，同时将函数中的数据返回给主程序。 **2. 变量作用域** Python 变量并不是在哪个位置都可以访问的，具体的访问权限取决于变量定义的位置，其所处的有效范围视为变量的作用域。根据作用域的不同，变量可以划分为局部变量和全局变量。 **（1）局部变量** 在函数内部定义的变量称为局部变量，局部变量只能在定义它的函数内使用。 **（2）全局变量** 全局变量是指在函数之外定义的变量，它在程序的整个运行周期内都用存储单元。默认情况下，函数的内部只能获取全局变量，而不能修改全局变量的值。 **（3）global 关键字** 如果在函数内修改全局变量，需要在函数内使用 global 关键字声明。 **（4）nonlocal 关键字** 使用 nonlocal 关键字可以在函数作用域内修改嵌套作用域中的变量。 课程思政： 从局部变量和全局变量的关系引出个人与集体的关系： 我们应该正确的看待个人与集体的关系，不能把个人脱离于集体之外，个人利益与集体利益总是息息相关。当个人利益和集体利益发生冲突的时候，个体利益应当服从集体利益。 作为学校、班级中的一员，我们应该遵守学校的纪律，服从班集体的决定，承担相应的责任。	1.讲授知识点 2.提出问题：局部变量和全局变量的区别； 3.总结回答情况； 4. 课程思政：从局部变量和全局变量的关系引出个人与集体的关系	1. 认真听讲 2. 思考问题 3. 小组讨论 4. 提交讨论结果 5. 思考问题：个人与集体的关系	1.教学平台； 2.多媒体课件； 3.PyCharm 软件； 4.Python3.7 编辑软件； 5. 分组讨论；

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
 	
45 分钟
	任务一 实现四则运算器
	1.不带参数的四则运算器
	def oper():
	operator = input('请选择要执行的运算符：+、-、*、/ ' + '\n')
	if operator == "+":
	num=6+5
	print("计算结果为:", num)
	elif operator == '-':
	num=6-5
	print("计算结果为:", num)
	elif operator == '*':
	num=6*5
	print("计算结果为:", num)
	elif operator == '/':
	num=6/5
	print("计算结果为:", num)
	oper()
	2.优化四则运算器
	def oper(parm_one, parm_two):
	operator = input('请选择要执行的运算符：+、-、*、/ ' + '\n')
	if operator == "+":
	print("计算结果为:", +parm_one + parm_two)
	elif operator == '-':
	print("计算结果为:", parm_one - parm_two)
	elif operator == '*':
	print("计算结果为:", parm_one * parm_two)
	elif operator == '/':
	print("计算结果为:", parm_one / parm_two)
	num_one = int(input('请输入第一个数:'))
	num_two = int(input('请输入第二个数:'))
	oper(num_one, num_two)
		1.巡回指导；
		2.解疑答难
		常见问题：
		int(input('请输入第一个数:'))
			1.比较不带参数的函数和带参数的函数的区别；
			2.小组为单位进行编写代码、运行测试
				1.API 文档；
				2.网络资源；
				3.教学平台；
				4.多媒体课件；
				5.PyCharm 软件；
				6.Python3.7

<table>
<tr><th>环节
用时</th><th>内容</th><th>教师活动</th><th>学生活动</th><th>教学方法
与手段</th></tr>
<tr>
<td>分组
实施

45 分钟</td>
<td>
任务四　用递归函数输出斐波那契数列
1.递归函数的语法格式
如果一个函数中调用了函数本身，这个函数就是递归函数。
递归函数只需少量代码就可描述出解题过程所需要的多次重复计算，大大地减少了程序的代码量。
2.演示动画，递归函数的原理
3.实现阶乘
代码：
<pre>
def fun_n(count):
    if count==1:
        result=1
    else:
        result=fun_n(count-1)*count
    return result
number=int(input("请输入一个正整数："))
print("%d! ="%number,fun_n(number))
</pre>
运行结果：
<pre>
请输入一个正整数：7
7! = 5040
</pre>
4.用递归函数输出斐波那契数列
代码：
<pre>
def fibonacci(n):
    if n == 1 or n == 2:
        return 1
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)
num = int(input('请输入一个正整数: '))
for i in range(1, num + 1):
    print(fibonacci(i), end=' ')
</pre>
运行结果：
<pre>
请输入一个正整数: 7
1 1 2 3 5 8 13
</pre>
</td>
<td>1.演示动画，讲解递归函数的原理；
2.巡回指导；
3.解疑答难；
容易出现的问题：result=fun_n(count-1)*count</td>
<td>1. 观看动画，学习递归函数的原理；
2.小组为单位进行编写代码、运行测试</td>
<td>1.利用动画演示递归函数的原理；
2.网络资源；
3.教学平台；
4.多媒体课件；
5.PyCharm软件；
6.Python3.7</td>
</tr>
</table>

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
 	
45 分钟
	5．拓展案例：实现汉诺塔游戏
	汉诺塔源于印度一个古老传说：大梵天创造世界的时候做了三根金刚石柱子，一根柱子上从下往上按照从大到小的顺序摞着 64 片黄金圆盘，大梵天命令婆罗门把圆盘从下面开始按照从大到小的顺序重新摆放在另一根柱子上，并规定：小圆盘上不能放大圆盘，三根柱子之间一次只能移动一个圆盘。
	代码：
	`def hanoi(n, ch1, ch2, ch3):`
	`  if n == 1:`
	`  print(ch1, '->', ch3)`
	`  else:`
	`  hanoi(n - 1, ch1, ch3, ch2)`
	`  print(ch1, '->', ch3)`
	`  hanoi(n - 1, ch2, ch1, ch3)`
	`plate_nums = int(input("请输入盘子的数量："))`
	`hanoi(plate_nums, 'A', 'B', 'C')`
	运行结果：
	请输入盘子的数量：3
	A -> C
	A -> B
	C -> B
	A -> C
	B -> A
	B -> C
	A -> C
		1.演示案例运行效果；
		2.动画演示汉诺塔游戏过程；
		3.解疑答难容易出现的问题：hanoi(plate_nums, 'A', 'B', 'C')
			1.学习动画理解汉诺塔游戏过程；
			2.小组为单位进行编写代码、运行测试
				1.利用动画演示诺塔游戏过程；
				2.网络资源；
				3.教学平台；
				4.多媒体课件；
				5.PyCharm 软件；
				6.Python3.7

小组 汇报 15 分钟	各小组汇报代码编写及运行调试情况	点评小组作品	各小组汇报展示、各小组互相评分	PyCharm 软件
课后 拓展 5 分钟	1.登录系统账号检测 2.输出汉诺塔移动过程	1.引发思考 2.布置任务	课后查阅资料，完成作业后上传课程平台	1.教学平台 2.PyCharm 软件

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**
完善 作品 15 分钟	项目组根据老师的点评，继续完善作品，上传至教学平台。	批阅作业	完善作品 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	根据自己需要学习网络资源	利用网络资源拓宽学生视野
拓展 作业 20 分钟	1.登录系统账号检测 2.输出汉诺塔移动过程 3.运用函数实现学生管理系统	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 项目五 可视化图像检测结果	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

教学内容

任务一

（实现四则运算器）

一、创设情境，导入函数的意义

1. 教师通过提问引出函数的意义与应用场景

教师学提问，如果程序较复杂，并且程序中重复执行某一功能，如何提高代码利用率，使程序结构更加清晰？引出函数的意义。函数被指封装起来的、实现某种功能的一段代码，它可以被其他函数调用。当程序实现的功能较为复杂时，开发人员通常会提取其中的功能性代码模块化为一个函数，提高代码复用性、降低代码冗余、使程序结构更加清晰。

2. 明确学习目标

（1）理解函数的含义

（2）掌握函数的定义

（3）掌握函数的调用

二、课前检查

要求在在线教学平台上完成课前测试题。

三、重点知识讲解

1. 什么是函数

函数是组织好的、可重复使用的、用于实现单一或相关联功能的代码段，通过函数的名称表示和调用。函数也可以看作是一段有名字的子程序，可以在需要的地方使用函数名调用执行。

2. 函数的定义

Python 使用 def 关键定义函数，其语法格式如下：

def 函数名([参数列表]):

函数体

[return 语句]。

3. 函数的调用

函数定义好之后不会立即执行，直到被程序调用时才会生效。通过函数名(参数列表)方式调用。

```
def my_absolute(x):
    if x>=0:
        print(x)
    else:
        print(-x)
my_absolute(-10.3)
```

教学内容

课程思政：

通过函数的意义引出：

1.日常生活中，要完成一件复杂的功能，我们总是习惯把“大功能”分解为多个“小功能”以实现。在编程的世界里，“功能”可称呼为“函数”，因此“函数”其实就是一段实现了某种功能的代码，并且可以供其它代码调用；

2. 通过多人合作完成项目引出分工与合作，培养学生沟通交流与团队合作能力。

4. 学生根据教师提示，完成四则运算器

案例源码：

```
def oper():
    operator = input('请选择要执行的运算符：+、-、*、/' + '\n')
    if operator == "+":
        num=6+5
        print("计算结果为:", num)
    elif operator == '-':
        num=6-5
        print("计算结果为:", num)
    elif operator == '*':
        num=6*5
        print("计算结果为:", num)
    elif operator == '/':
        num=6/5
        print("计算结果为:", num)
oper()
```

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。

带领学生回顾函数的意义、函数的定义与函数的调用。

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。

教学内容

任务二

（运用参数传递数据）

一、回顾任务一内容，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

任务一我们完成简易的四则运算，但两个数是固定的，每次调用函数，执行结果是相同的，这种运算器不能解决实际问题。本次任务我们将学习使用参数传递数据，操作数与被操作数是用户输入的数据，每次执行结果会根据用户需求不同。

2.明确学习目标

（1）要求学生掌握位置参数

（2）要求学生掌握关键字参数

（3）要求学生掌握默认参数

（4）要求学生掌握不定长参数

二、进行重点知识的讲解

1.什么是函数参数传递，为什么用函数参数

现要求定义一个函数，这个函数用于计算两个数的和，并把计算的结果打印出来。

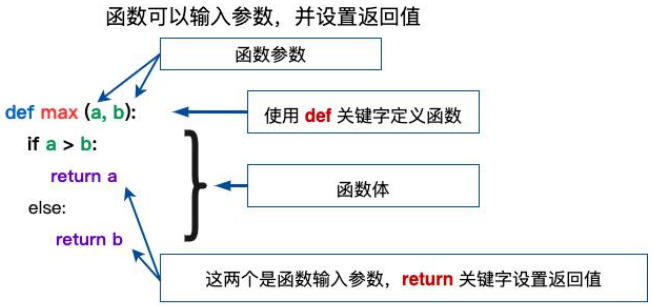
如 def add():

```
    c=11+22
    print(c)
```

这个函数的功能是计算 11 和 22 的和。此时，无论调用这个函数多少次，得到的结果永远都一样，而且只能计算固定的两个数字的和，使得这个函数的局限性很大。为了能让定义的函数更加通用，可以在定义函数的时候添加两个参数，让两个参数来接收传递函数的值。

```
def  add(a,b)
    c=a+b
    print(c)
```

函数的参数传递是指将实参传递给形参的过程，Python 中的函数支持以多种方式传递参数，包括位置传递、关键字传递、默认值传递、包裹传递、解包裹传递以及混合传递。



教学内容	**2.参数的位置传递**
	调用函数时，默认按照位置顺序将对应的实参传递给形参，即将第一个实参分配给第一个形参，第二个实参分配给第二个形参，以此类推。
	案例：比较两个数，并返回较大的数
	def max(a, b):
	if a > b:
	return a
	else:
	return b
	a = 4
	b = 5
	print(max(a, b))
	3.参数的关键字传递
	关键字传递通过“形参变量名=实参”的形式将形参与实参关联，根据形参的名称进行参数传递，它允许实参和形参的顺序不一致。
	def makeup_url(protocal,address):
	print(protocal+'://'+address)
	makeup_url("http","www.163.com")
	makeup_url(address="www.baidu.com",protocal="http")
	makeup_url(protocal="http",address="www.sina.com.cn")
	4.参数的默认值传递
	函数在定义时可以给每个参数指定默认值，基本形式为：函数名（参数=默认值），这样在调用时既可以给带有默认值的参数从新赋值，也可以省略相应的实参，使用参数的默认值。
	def makeup_url(address,protocal="http"):
	print(protocal+'://'+address)
	makeup_url(address="www.163.com")
	makeup_url(protocal="https",address="www.sina.com.cn")
	5.不定长参数传递
	格式：def 函数名([formal_args,] *args,**kwargs):
	（1）函数传入的参数个数会优先匹配 formal_args 参数的个数
	（2）如果传入的参数没有指定名称，那么*args 会以元组的形式存放多余的参数
	（3）如果传入的参数指定了名称，如 m=1，那么**kwargs 会以字典的形式存放这些被命名的参数。

教学内容	案例：
	def func(a,b,c=0,*args,**kw):
	print(a)
	print(b)
	print(c)
	print(args)
	print(kw)
	func(1,2)
	func(1,2,c=3)
	func(1,2,3,'a','b')
	func(1,2,3,'a','b',x=99,y=88)
	6．优化四则运算器
	def oper(parm_one, parm_two):
	operator = input('请选择要执行的运算符：+、-、*、/ ' + '\n')
	if operator == "+":
	print("计算结果为:", +parm_one + parm_two)
	elif operator == '-':
	print("计算结果为:", parm_one - parm_two)
	elif operator == '*':
	print("计算结果为:", parm_one * parm_two)
	elif operator == '/':
	print("计算结果为:", parm_one / parm_two)
	num_one = int(input('请输入第一个数:'))
	num_two = int(input('请输入第二个数:'))
	oper(num_one, num_two)
	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。
	带领学生回顾函数参数的作用，四种参数传递方式。
	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。

教学内容

任务三

（定义局部和全局变量）

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.教师通过提问学生问题，由上一任务引出本课时要讲解的内容。

3.明确学习目标

（1）要求学生掌握什么是变量作用域

（2）要求学生掌握局部变量

（3）要求学生掌握全部变量

二、进行重点知识的讲解

1.函数返回值的作用

函数中的 renturn 语句是可选项，可以出现在函数体的任何位置，它的作用是结束当前函数，将程序返回到函数被调用的位置继续执行，同时将函数中的数据返回给主程序。

```
def is_capital(words):
    if ord("A")<=ord(words[0])<=ord("Z"):  #ord()函数将字符转换为 ASCII 码
        return '首字母是大写的'
    else:
        return '首字母不是大写的'
result = is_capital("Python")   # 将函数返回的结果交给变量
print(result)
```

2.变量作用域

Python 变量并不是在哪个位置都可以访问的，具体的访问权限取决于变量定义的位置，其所处的有效范围视为变量的作用域。根据作用域的不同，变量可以划分为局部变量和全局变量。

（1）局部变量

在函数内部定义的变量称为局部变量，局部变量只能在定义它的函数内使用。

```
def test_one():
    number=100
    print("test_one 中的 number 值为：%d"%number)
def test_two():
    number=200
    print("test_two 中的 number 值为：%d"%number)
test_one()
test_two()
```

教学内容	**（2）全局变量**
	全局变量是指在函数之外定义的变量，它在程序的整个运行周期内都用存储单元。默认情况下，函数的内部只能获取全局变量，而不能修改全局变量的值。
	result=100 *#全局变量*
	def sum(a,b):
	result=a+b *#局部变量*
	print("函数内的 result 的值为：",result) *#result 在是局部变量*
	return result
	#调用 sum 函数
	sum(100,200)
	print("函数外的变量 result 是全局变量，等于",result)
	（3）global 关键字
	如果在函数内修改全局变量，需要在函数内使用 global 关键字声明。
	a=100
	def test():
	a+=100
	print(a)
	test()
	程序报错
	a=100
	def test():
	global a
	a+=100
	print(a)
	test()
	（4）nonlocal 关键字
	使用 nonlocal 关键字可以在函数作用域内修改嵌套作用域中的变量。
	def func():
	count=1
	def func_in():
	count=12
	func_in()
	print(count)
	func()
	内层 func_in 函数没有对外出 count 变量进行修改，而是重新定义了一个同名变量 count，若要修改外层 func 中的变量 count，则需要在变量前使用关键字

教学内容	nonlocal
	def func():
	count=1
	def func_in():
	nonlocal count
	count=12
	func_in()
	print(count)
	func()
	课程思政：
	从局部变量和全局变量的关系引出个人与集体的关系：
	我们应该正确的看待个人与集体的关系，不能把个人脱离于集体之外，个人利益与集体利益总是息息相关。个人只有在集体中，并承担一定的职责，使命，才能使自身价值得以实现，如果脱离了集体，个人就丧失了作为这一集体的成员的资格，也就无须承担这一集体的义务，也无权享受这一集体中的成员能够享受的权利。当个人利益和集体利益发生冲突的时候，个体利益应当服从集体利益。
	作为学校、班级中的一员，我们应该遵守学校的纪律，服从班集体的决定，承担相应的责任。
	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
	带领学生回顾函数返回值的作用、全局变量与局部变量、global 关键字、nonlocal 关键字的用法。
	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。
	（1）Python 中使用关键字_____声明一个函数。
	（2）匿名函数使用关键健字_____声明。
	（3）在函数内部对全局变量进行修改,需要先使用_____关键字声明。
	（4） 函数可以提高代码的复用性。 （ ）
	（5）全局变量在所有的函教中都可以访问。 （ ）
	（6）函数的位置参教有严格的位置关系。 （ ）

教学内容

任务四

（用递归函数输出斐波那契数列）

一、回顾任务三内容，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.教师通过提问学生问题，由任务三引出任务四要完成的内容。

3.明确学习目标

（1）要求学生了解什么是递归函数

（2）要求学生掌握阶乘的实现过程

（3）要求学生掌握斐波那契数列的实现过程

二、进行重点知识的讲解

1. 递归函数的语法格式

递归是一个函数过程在定义中直接调用自身的一种方法，它通常把一个大型的复杂问题层层转化为一个与原问题相似，但规模较小的问题进行求解。

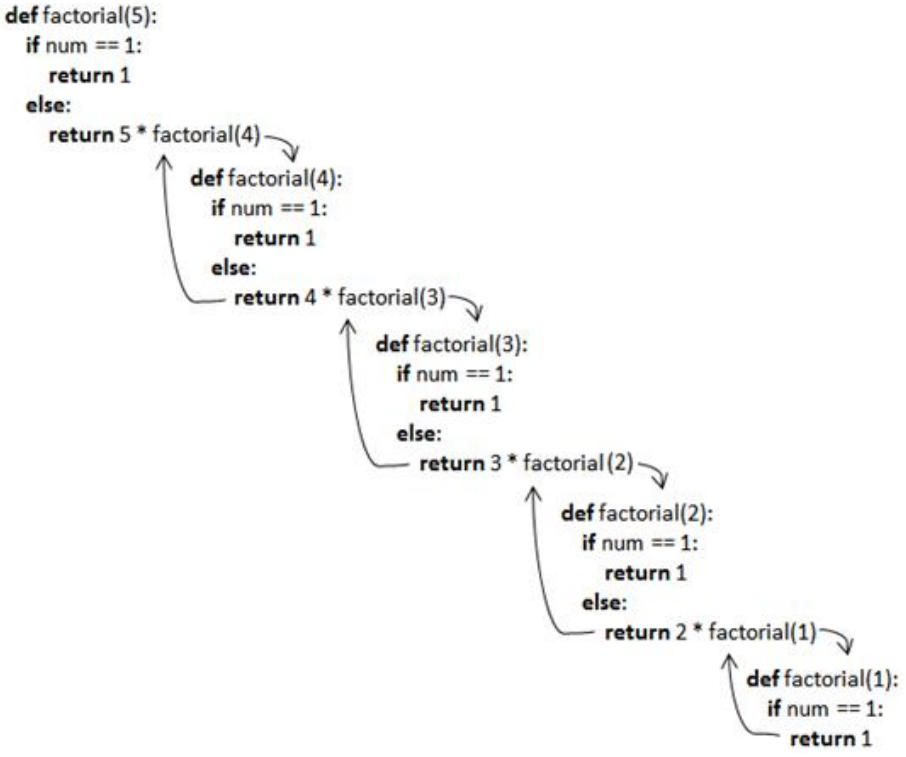
如果一个函数中调用了函数本身，这个函数就是递归函数。

递归函数只需少量代码就可描述出解题过程所需要的多次重复计算，大大地减少了程序的代码量。

2. 演示动画，递归函数的原理

3. 实现阶乘

阶乘是可利用递归方式求解的经典问题。



教学内容

代码：

```
def fun_n(count):
    if count==1:
        result=1
    else:
        result=fun_n(count-1)*count
    return result
number=int(input("请输入一个正整数："))
print("%d! ="%number,fun_n(number))
```

运行结果：

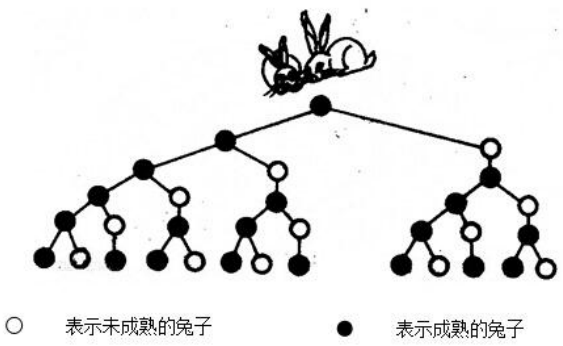
```
请输入一个正整数：7
7! = 5040
```

3．用递归函数输出斐波那契数列

斐波那契数列（Fibonacci sequence），又称黄金分割数列，因数学家莱昂纳多·斐波那契（Leonardoda Fibonacci）以兔子繁殖为例子而引入，故又称为“兔子数列”，指的是这样一个数列：0、1、1、2、3、5、8、13、21、34、……在数学上，斐波那契数列以如下被以递推的方法定义：F(0)=0，F(1)=1， F(n)=F(n－1)+F(n－2)（n ≥ 2，n ∈ N*）在现代物理、准晶体结构、化学等领域，斐波纳契数列都有直接的应用。

斐波那契在《计算之书》中提出了一个有趣的兔子问题：若一对成年兔子每个月恰好生下一对小兔子(一雌一雄)。在年初时，只有一对小兔子。在第一个月结束时，他们成长为成年兔子，并且第二个月结束时，这对成年兔子将生下一对小兔子。这种成长与繁殖的过程会一直持续下去，并假设生下的小兔子都不会死，那么一年之后共可有多少对小兔子？

繁殖的过程可以通过一棵“家族树”来表示：



教学内容

代码：

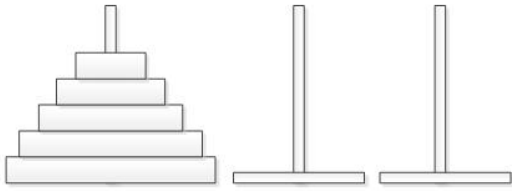
```
"""
使用递归解决 斐波那契数列
"""
def fibonacci(n):
    if n == 1 or n == 2:
        return 1
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)

num = int(input('请输入一个正整数: '))
for i in range(1, num + 1):
    print(fibonacci(i), end=' ')
```

运行结果：

```
请输入一个正整数: 7
1 1 2 3 5 8 13
```

4. 拓展案例：实现汉诺塔游戏

汉诺塔源于印度一个古老传说：大梵天创造世界的时候做了三根金刚石柱子，一根柱子上从下往上按照从大到小的顺序摞着 64 片黄金圆盘，大梵天命令婆罗门把圆盘从下面开始按照从大到小的顺序重新摆放在另一根柱子上，并规定：小圆盘上不能放大圆盘，三根柱子之间一次只能移动一个圆盘。



诺塔问题不管在任何编程语言里都是经典问题，是采用递归算法的经典案例，该问题可以抽象如下：

（1）3 根圆柱 A，B，C，其中 A 上面串了 n 个圆盘

（2）这些圆盘从上到下是按从小到大顺序排列的，大的圆盘任何时刻不得位于小的圆盘上面

（3）每次移动一个圆盘，最终实现将所有圆盘移动到 C 上

利用 Python 语言接近自然语言的特性，开发者可以更容易的将递归算法翻译成程序语句，需要的代码量很小。汉诺塔问题的解决步骤用语言描述很简单，仅三步：

A，B，C 三个圆柱，分别为初始位，过渡位，目标位，设 A 柱为初始位，C 位

教学内容	为最终目标位
	（1）将最上面的 n-1 个圆盘从初始位移动到过渡位
	（2）将初始位的最底下的一个圆盘移动到目标位
	（3）将过渡位的 n-1 个圆盘移动到目标位
	对于递归算法中的嵌套函数 f（n-1）来说，其初始位，过渡位，目标位发生了变化。
	代码：
	`def hanoi(n, ch1, ch2, ch3):`
	`  if n == 1:`
	`  print(ch1, '->', ch3)`
	`  else:`
	`  hanoi(n - 1, ch1, ch3, ch2)`
	`  print(ch1, '->', ch3)`
	`  hanoi(n - 1, ch2, ch1, ch3)`
	
	`plate_nums = int(input("请输入盘子的数量："))`
	`hanoi(plate_nums, 'A', 'B', 'C')`
	运行结果：
	请输入盘子的数量：3
	A -> C
	A -> B
	C -> B
	A -> C
	B -> A
	B -> C
	A -> C
	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
	带领学生回顾函数的递归调用原理。
	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。

<table>
<tr><td>教学内容总结</td><td colspan="2">
<table>
<tr><th>任务</th><th>知识目标</th><th>思政元素</th></tr>
<tr><td>任务一
实现四则运算器</td><td>（1）理解函数的含义
（2）掌握函数的定义
（3）掌握函数的调用</td><td rowspan="2">1.日常生活中，要完成一件复杂的功能，我们总是习惯把“大功能”分解为多个“小功能”以实现。在编程的世界里，“功能”可称呼为“函数”，因此“函数”其实就是一段实现了某种功能的代码，并且可以供其它代码调用；
2. 通过多人合作完成项目引出分工与合作，培养学生沟通交流与团队合作能力。</td></tr>
<tr><td>任务二
运用参数传递数据</td><td>（1）掌握位置参数
（2）掌握关键字参数
（3）掌握默认参数
（4）掌握不定长参数</td></tr>
<tr><td>任务三
定义局部和全局变量</td><td>（1）掌握什么是变量作用域
（2）掌握局部变量
（3）掌握全部变量</td><td rowspan="2">从局部变量和全局变量的关系引出个人与集体的关系：
我们应该正确的看待个人与集体的关系，不能把个人脱离于集体之外，个人利益与集体利益总是息息相关。个人只有在集体中，并承担一定的职责，使命，才能使自身价值得以实现。
作为学校、班级中的一员，我们应该遵守学校的纪律，服从班集体的决定，承担相应的责任。</td></tr>
<tr><td>任务四
用递归函数输出斐波那契数列</td><td>（1）了解什么是递归函数
（2）掌握阶乘的实现过程
（3）掌握斐波那契数列的实现过程</td></tr>
</table>
</td></tr>
<tr><td>教学反思</td><td>经验</td><td>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</td></tr>
<tr><td></td><td>不足</td><td>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</td></tr>
<tr><td>拓展作业</td><td colspan="2">输出汉诺塔移动过程</td></tr>
</table>