

Python计算生态构建

嵩天

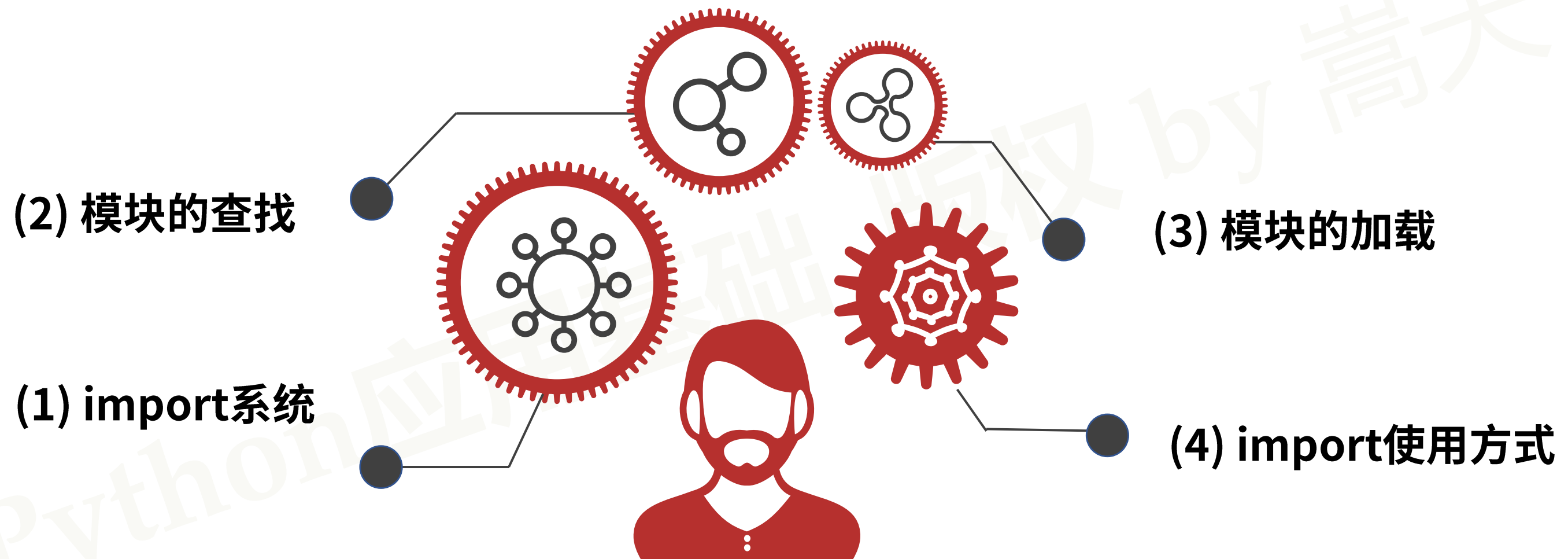
深入理解import系统

高天

深入理解import系统

单元开篇

单元开篇



深入理解import系统

深入理解import系统

import系统

import系统

import系统：扩展命名空间及功能的方法

- 构成：import保留字、__import__()函数和importlib标准库
- 步骤：模块的查找、模块的加载
- 价值：import系统是Python代码复用和命名空间管理的精髓

import系统

import系统：扩展命名空间及功能的方法

- import保留字：调用__import__()进行模块查找，以及模块的加载
- __import__()函数：模块的查找，建立模块对象
- importlib标准库：与import系统相关的丰富API

import系统



模块对象

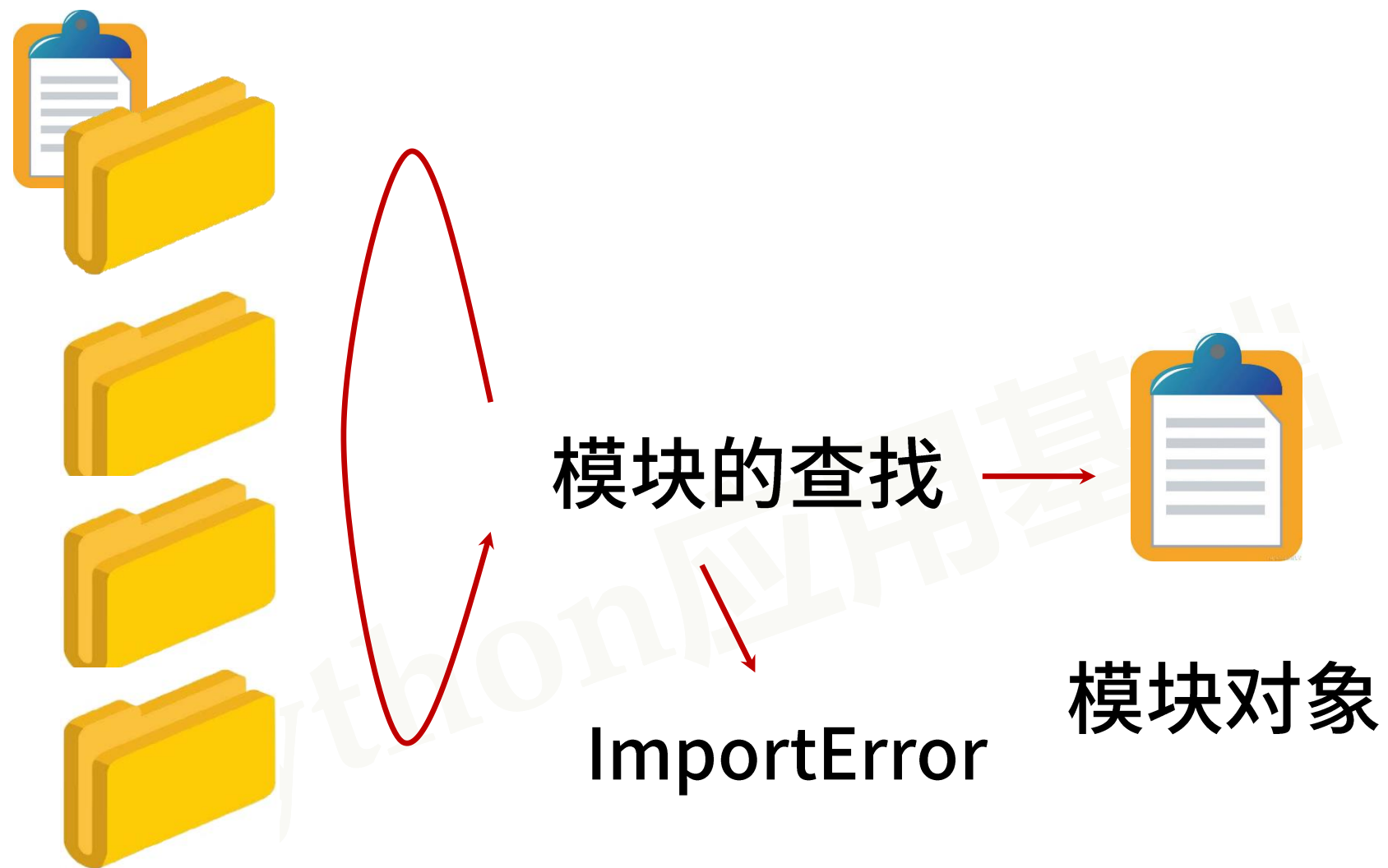
万物皆对象：模块被导入后成为了对象

- 模块的对象形式：模块在程序中使用都是以对象形式体现
- 类似类对象，模块对象只有一个
- 模块对象生成时，模块中代码会被执行，因此会有类对象产生

深入理解import系统

模块的查找

模块的查找



理解模块查找过程

模块的查找

查找的开始

- 输入参数：模块的名字，例如：pkg.pkg1.m1
- 输入方式：import保留字、__import__()、importlib标准库
- 基本行为：按层次结构逐层查找，例如：pkg->pkg1->m1

模块的查找

查找路径：1. `sys.modules`

- 第一步查找`sys.modules`，之前被引入模块的缓存（cache）
- `sys.modules`是一个字典，<模块名/对象引用>:<加载路径>
- 如果模块不在`sys.modules`中，则进入下一步

模块的查找

查找路径：2. 查找策略

- 用户通过import钩子扩展的查找模式
- 内置模块的路径
- `sys.path`（列表变量）提供的加载路径，可以是zip文件或URL

模块的查找

import钩子: import hook

- 扩展查找模块的方式: meta_path方式和import路径方式
- meta_path: 将查找方法增加到sys.meta_path列表变量
- import路径: 将查找方法注册到sys.path_hooks列表变量

模块的查找

模块对象缓存：sys.modules

import钩子的扩展

内置模块的路径

sys.path加载的路径



模块的查找

ImportError

模块对象



深入理解import系统

模块的加载

模块的加载

理解模块加载过程



模块对象



模块的加载



当前命名空间

模块的加载

建立模块对象的过程

- 1. 找到模块后，如果模块对象存在，则使用现有模块对象
- 2. 创建一个新的模块对象，将其加入sys.modules
- 3. 在程序当前命名空间执行模块代码

模块的加载

建立模块对象的过程

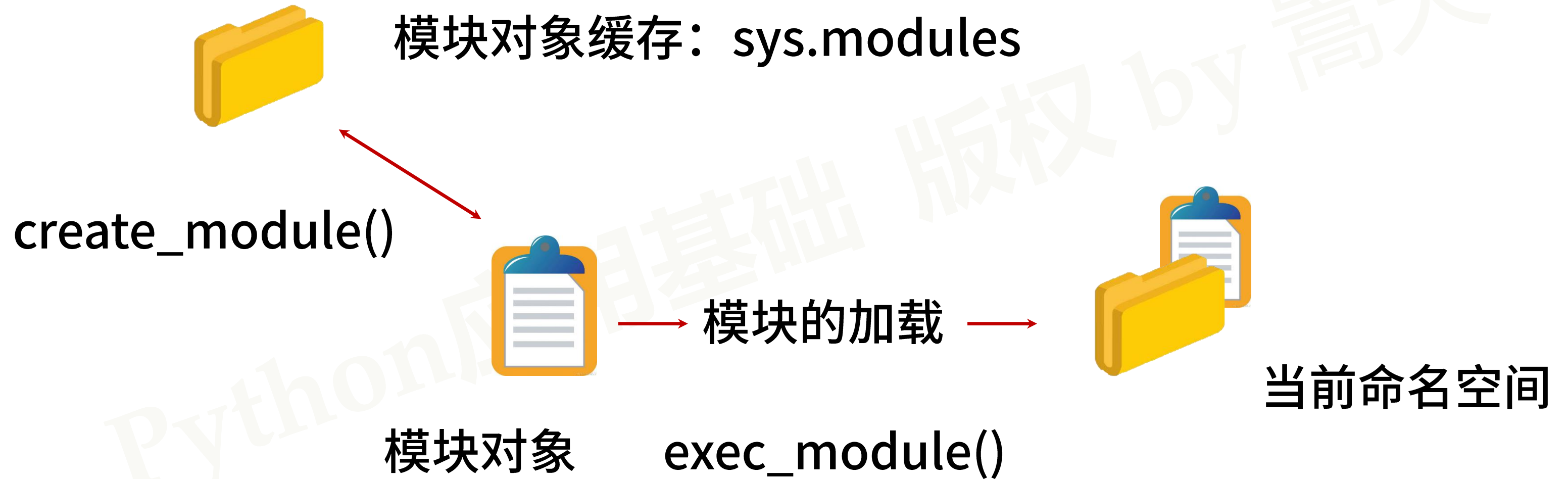
- 创建对象: `importlib.abc.Loader.create_module()`
- 执行对象: `importlib.abc.Loader.exec_module()`

模块的加载

模块对象的命名空间

- 模块的命名空间与引用位置有直接关系
- 如果引用在文件顶层，则使用文件顶层命名空间访问模块
- 如果引用在非顶层，则使用局部命名空间访问模块

模块的加载



深入理解import系统

import使用方式

import使用方式

import的三种使用方式

- `import <模块名>`
- `from <模块名> import <类/函数名/*>`
- `import <模块名> as <别名>`

import使用方式 (1)

import <模块名>

- 当前命名空间下的一个子命名空间
- 成功加载后，产生一个与<模块名>同名的<模块对象名>
- 实际上：<模块对象名>.<子命名空间内元素>方式访问

import使用方式 (2)

from <模块名> import <类/函数/*>

- 将导入元素加载到当前命名空间下
- 成功加载后，产生类对象或函数对象，覆盖同名对象
- 实际上：<类对象名> 或<函数对象名>方式访问

import使用方式 (3)

import <模块名> as <别名>

- 当前命名空间下的一个子命名空间
- 成功加载后，产生一个与<别名>同名的<模块对象名>
- 实际上：<模块对象名>.<子命名空间内元素>方式访问

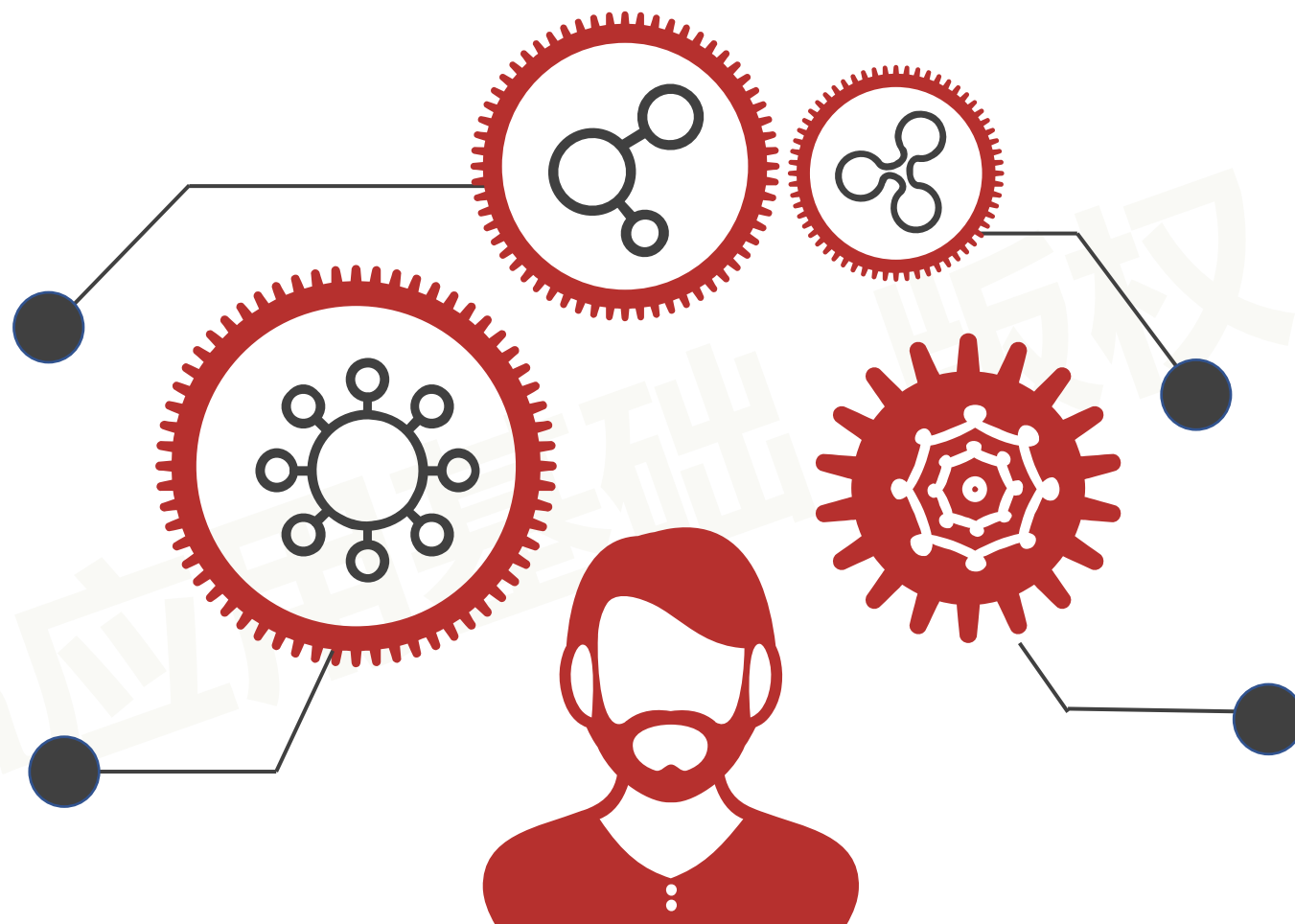
深入理解import系统

单元小结

单元小结

(2) 模块的查找
sys.modules、查找策略、钩子

(1) import系统
模块查找和加载、模块对象



(3) 模块的加载
模块对象的创建和执行

(4) import使用方式
三种使用方式再回顾

深入理解import系统

 Python ▶ 123

Thank you