

极客学院  
jikexueyuan.com

# Python模块

# Python模块 — 课程概要

认识 Python 模块

字节编译

`from... import` 详解

认识 `__name__` 属性

自定义模块

`dir()` 函数



# 认识 Python 模块

# 认识 Python 模块

本课时包括以下知识点：

- 什么是模块？
- 如何导入模块
- sys 模块

## 认识 Python 模块 — 什么是模块？

上一门课我们学习过了函数，函数是可以实现一项或多项功能的一段程序。我们的模块是函数功能的扩展，模块是可以实现一项或多项功能的程序块。我们从其定义可以发现，函数是一段程序，模块是一项程序块，也就是说函数和模块都是用来实现功能的，但是模块的范围比函数要广，在模块里面，可以重用多个函数。下面我们通过实例来看一下什么是模块。

## 认识 Python 模块 — 如何导入模块

Python 中在一个程序中使用某个模块，不能直接使用模块里面的功能。在使用模块之前，必须先导入指定模块，只要导入了一个模块，才能使用一个模块。下面我们通过实例来讲解一下如何导入模块。

## 认识 Python 模块 — sys 模块

在 Python 中有一些模块是不用我们自己去定义，Python 官方提供的自带的模块，这些模块伴随的 Python 的安装而产生，我们把这些模块叫做标准库模块。标准库模块是一类模块，而不是特指某一种模块。标准库模块中，也有非常多的模块，有的模块与电子邮件有关，有的模块与操作系统相关，还有的模块与系统功能有关，等等。我们把在标准库中与系统功能有关的这些模块称之为 sys 模块。下面我们通过实例来讲解一下 sys 模块的基本使用。



# 字节编译



# 字节编译

本课时包括以下知识点：

- 什么是 Python 的 .pyc 文件
- 字节编译与编译的区别
- .pyc 文件的产生
- .pyc 文件的使用

## 字节编译 — 什么是 Python 的 .pyc 文件

首先，我们所说的 .pyc 文件是指以 .pyc 为后缀名的这一类文件。我们知道，由于我们计算机只认识二进制的机器语言，所以在执行 Python 模块的时候，有两种执行方式：一种是先将模块里面的内容编译成二进制语言，然后执行这些二进制语言，另一种是直接执行对应模块的二进制语言程序。第二种方式省略了编译这一步，所以执行速度相对来说要快一些。而把模块编译成二进制语言程序的这个过程叫做字节编译，这个过程会产生一个与编译的模块对应的 .pyc 文件。 .pyc 文件就是经过编译后的模块对应的二进制文件。

## 字节编译 — 字节编译与编译的区别

上面我们已经知道，把模块编译成二进制语言程序的这个过程叫做字节编译。记得我们之前提到过，Python是一种解释型语言，而不是一种编译型语言，那么为什么现在又说Python中有字节编译呢？很多人认为，Python中出现字节编译，那么Python就应该是一种编译型语言。其实这种观点是不正确的。Python中虽然出现了编译这个过程，但是Python中编译的过程是在Python的解释器中发生的。换句话说，其实每种程序要想让计算机执行，都只能变成二进制的形式，编译型语言是指在软件中就有一个独立的编译模块去将程序编译，而Python中字节编译这部分功能是由解释器完成的，所以Python仍然是解释型语言。

## 字节编译 — .pyc文件的产生

上面我们已经知道，字节编译这个过程会产生一个与编译的模块对应的 .pyc文件。 .pyc文件就是经过编译后的模块对应的二进制文件。下面我们通过实例来看一下两种 .pyc文件产生的方式。

## 字节编译 — .pyc文件的使用

在 Python 中，.pyc 文件最大的一个作用就是加快了模块的运行速度。那们 .pyc 文件具体来说有什么用呢？如果我们深入学习下去，我们就会知道 .pyc 文件今后还可以做反编译等等高级功能。下面我们通过实例来看一下 .pyc 文件的内容。



from... import 详解

## from... import 详解

本课时包括以下知识点：

- 学会使用 from... import
- 学会使用 from... import \*

## from... import 详解 — 学会使用 from... import

Python中导入一个模块的方法我们可以使用 `import`, 但是 `import` 只是导入了这个模块, 而并没有导入我们模块的某个属性或方法。而我们要不仅导入一个模块, 还有导入模块中对应的一个功能, 我们可以使用 `from... import` 语句。下面我们通过实例来看一下 `from... import` 语句。



## from... import 详解 — 学会使用 from... import\*

在 Python 中使用 from... import 语句只能一次导入一个模块的一个功能，我们如果要想一次性把这个模块的所有功能，也就是所有属性与方法都导入的话，我们可以使用 from... import\* 语句。下面我们通过实例来看一下 from... import\* 语句。

# 认识 `__name__` 属性

# 认识 \_\_name\_\_ 属性

本课时包括以下知识点：

- 认识主模块
- 认识 \_\_name\_\_ 属性
- \_\_name\_\_ 属性的使用

## 认识 `__name__` 属性 — 认识主模块

我们 Python 函数中，如果一个函数调用其他函数完成一项功能，我们称这个函数为主函数，如果一个函数没有调用其他函数，我们这种函数叫做非主函数。模块也类型，如果一个模块是被直接使用的，而没有被别人调用，我们称这个模块为主模块，如果一个模块被别人调用，我们这种模块叫做非主模块。下面我们通过实例认识一下主模块。

## 认识 `__name__` 属性 — 认识 `__name__` 属性

在 Python 中有主模块与非主模块之分，那么如何区分主模块与非主模块呢？是这样的，如果一个模块的 `__name__` 属性的值是 `__main__`，那么就说明这个模块是主模块，反之亦然。其实我们可以把 `__name__` 看作一个变量，这个变量是系统给出的，这个变量的功能是判断一个模块是否是主模块。

## 认识 \_\_name\_\_ 属性 — \_\_name\_\_ 属性的使用

下面我们通过实例来讲解一下 \_\_name\_\_ 属性的使用。



# 自定义模块

# 自定义模块

本课时包括以下知识点：

- 什么是自定义模块
- 自定义模块使用详解



## 自定义模块 — 什么是自定义模块

Python中模块有非常多，有的模块是不需要用户自己去定义的去编写的，Python在安装的时候就自带的模块，这些模块是系统自带模块。而有一些模块刚好跟这种模块不同，他是需要我们自己去定义，自己去编写的模块，这些模块我们把其称之为自定义模块。

## 自定义模块 — 自定义模块使用详解

下面我们通过实例来讲解一下自定义模块如何创建与使用。



## dir()函数

# dir(函数

本课时包括以下知识点：

- 认识 dir(函数
- dir(函数扩展详解

## dir()函数 — 认识 dir()函数

Python中有非常多的模块，但是有的时候我们会忘记一个模块有哪些功能。这个时候，我们可以用 dir()函数来查看一下指定模块的功能列表。下面我们通过实例来讲解一下。

## `dir()`函数 — `dir()`函数扩展详解

Python中 `dir()`函数不仅仅能查看模块的功能列表，还能查看任意指定对象的功能列表，下面我们通过实例来讲解一下。

# Python模块

这本套课程中我们学习了 Python模块的相关内容。你应当掌握了以下知识：

- 1认识 Python模块
- 2字节编译
- 3.from... import详解
- 4认识 \_\_name\_\_ 属性
- 5自定义模块
- 6.dir()函数

如果想继续提高，你可以继续在极客学院学习 Python数据结构这门课程。

# 极客学院

jikexueyuan.com

中国最大的IT职业在线教育平台

