

第三章 第五节 Python 模块

Python 模块

什么是模块？

请思考与代码库类似的模块。

模块是包含一组函数的文件，希望在应用程序中引用。

创建模块

如需创建模块，只需将所需代码保存在文件扩展名为 `.py` 的文件中：

实例

在名为 `mymodule.py` 的文件中保存代码：

```
def greeting(name):  
    print("Hello, " + name)
```

第三章 第五节 Python 模块

Python 模块

使用模块

现在，我们就可以用 `import` 语句来使用我们刚刚创建的模块：

实例

导入名为 `mymodule` 的模块，并调用 `greeting` 函数：

```
import mymodule  
mymodule.greeting("Bill")
```

注释：如果使用模块中的函数时，请使用以下语法：

module_name.function_name

第三章 第五节 Python 模块

Python 模块

模块中的变量

模块可以包含已经描述的函数，但也可以包含各种类型的变量（数组、字典、对象等）：

实例

在文件 `mymodule.py` 中保存代码：

```
person1 = { "name": "Bill",  
            "age": 63,  
            "country": "USA"  
}
```

实例

导入名为 `mymodule` 的模块，并访问 `person1` 字典：

```
import mymodule  
a = mymodule.person1["age"]  
print(a)
```

第三章 第五节 Python 模块

Python 模块

为模块命名

您可以随意对模块文件命名，但是文件扩展名必须是 `.py`。

重命名模块

您可以在导入模块时使用 `as` 关键字创建别名：

实例

为 `mymodule` 创建别名 `mx`：

```
import mymodule as mx
a = mx.person1["age"]
print(a)
```

第三章 第五节 Python 模块

Python 模块

内建模块

Python 中有几个内建模块，您可以随时导入。

实例

导入并使用 `platform` 模块：

```
import platform
x = platform.system()
print(x)
```

Windows

第三章 第五节 Python 模块

Python 模块

使用 dir() 函数

有一个内置函数可以列出模块中的所有函数名（或变量名）。`dir()` 函数：

实例

列出属于 platform 模块的所有已定义名称：

```
import platform
```

```
x = dir(platform)
```

```
print(x)
```

```
['DEV_NULL', '_UNIXCONFDIR', '_WIN32_CLIENT_RELEASES', '_WIN32_SERVER_RELEASES', '__builtins__  
_', '__cached__', '__copyright__', '__doc__', '__file__', '__loader__', '__name__', '__packag  
e__', '__spec__', '__version__', '_comparable_version', '_component_re', '_default_architectu  
re', '_dist_try_harder', '_follow_symlinks', '_ironpython26_sys_version_parser', '_ironpython  
_sys_version_parser', '_java_getprop', '_libc_search', '_linux_distribution', '_lsb_release_v  
ersion', '_mac_ver_xml', '_node', '_norm_version', '_parse_release_file', '_platform', '_plat  
form_cache', '_pypy_sys_version_parser', '_release_filename', '_release_version', '_supported  
_dists', '_sys_version', '_sys_version_cache', '_sys_version_parser', '_syscmd_file', '_syscm  
d_uname', '_syscmd_ver', '_uname_cache', '_ver_output', '_ver_stages', 'architecture', 'colle  
ctions', 'dist', 'java_ver', 'libc_ver', 'linux_distribution', 'mac_ver', 'machine', 'node',  
'os', 'platform', 'popen', 'processor', 'python_branch', 'python_build', 'python_compiler', '  
python_implementation', 'python_revision', 'python_version', 'python_version_tuple', 're', 'r  
elease', 'subprocess', 'sys', 'system', 'system_alias', 'uname', 'uname_result', 'version', '  
warnings', 'win32_ver']
```

注释：dir() 函数可用于所有模块，也可用于您自己创建的模块

第三章 第五节 Python 模块

Python 模块

从模块导入

您可以使用 from 关键字选择仅从模块导入部件。

实例

名为 mymodule 的模块拥有一个函数和一个字典：

```
def greeting(name):  
    print("Hello, " + name)  
person1 = {  
    "name": "Bill",  
    "age": 63,  
    "country": "USA"  
}
```

实例

仅从模块导入 person1 字典：

```
from mymodule import person1  
print (person1["age"])
```

63

第三章 第五节 Python 模块

Python 模块

提示：在使用 from 关键字导入时，请勿在引用模块中的元素时使用模块名称。示例：person1["age"]，而不是 mymodule.person1["age"]。