

- ☑ **functionbody**: 可选参数, 用于指定函数体, 即该函数被调用后, 要执行的功能代码。如果函数有返回值, 可以使用 `return` 语句返回。

注意

- ① 函数体 “`functionbody`” 和注释 “`”comments”`” 相对于 `def` 关键字必须保持一定的缩进。
- ② 如果定义的函数暂时什么也不做, 那么需要使用 `pass` 语句作为点位符, 或者添加 Docstrings, 但不能直接添加一行单行注释。

例如, 定义一个过滤危险字符的函数 `filterchar()`, 代码如下:

```
01 def filterchar(string):
02     """功能: 过滤危险字符 (如黑客), 并将过滤后的结果输出
03         about: 要过滤的字符串
04         没有返回值
05     """
06     import re # 导入 Python 的 re 模块
07     pattern = r'(黑客)|(抓包)|(监听)|(Trojan)' # 模式字符串
08     sub = re.sub(pattern, '@_@', string) # 进行模式替换
09     print(sub)
```

运行上述代码, 将不显示任何内容, 也不会抛出异常, 因为 `filterchar()` 函数还没有被调用。

9.1.2 调用函数

调用函数也就是执行函数。如果把创建的函数理解为创建一个具有某种用途的工具, 那么调用函数就相当于使用该工具。调用函数的基本语法格式如下:

```
functionname([parametersvalue])
```

参数说明如下。

- ☑ **functionname**: 函数名称, 要调用的函数名称必须是已经创建好的。
- ☑ **parametersvalue**: 可选参数, 用于指定各个参数的值。如果需要传递多个参数值, 则各参数值间使用逗号 “,” 分隔; 如果该函数没有参数, 则直接写一对小括号即可。

例如, 调用 9.1.1 节创建的 `filterchar()` 函数, 可以使用下列代码:

```
01 about = '我是一名程序员, 喜欢看黑客方面的图书, 想研究一下 Trojan。'
02 filterchar(about)
```

调用 `filterchar()` 函数后, 将显示如图 9.3 所示的结果。

场景模拟: 第 5 章的例 5.1 实现了每日一帖功能, 但是这段代码只能执行一次, 如果想要再次输出, 还需要再重新写一遍。如果把这段代码定义为一个函数, 那么就可以多次显示每日一帖。

【例 9.1】 输出每日一帖 (共享版)。(实例位置: 资源包\TM\sl\09\01)

在 IDLE 中创建一个名称为 `function_tips.py` 的文件, 然后在该文件中创建一个名称为 `function_tips` 的函数, 在该函数中, 从励志文字列表中获取一条励志文字并输出, 最后调用函数 `function_tips()`, 代码如下: