15.20 处理C语言中的可迭代对象¶

问题¶

你想写C扩展代码处理来自任何可迭代对象如列表、元组、文件或生成器中的元素。

解决方案¶

下面是一个C扩展函数例子，演示了怎样处理可迭代对象中的元素：

```
static PyObject *py_consume_iterable(PyObject *self, PyObject *args) {
  PyObject *obj;
  PyObject *iter;
  PyObject *item;

  if (!PyArg_ParseTuple(args, "O", &obj)) {
    return NULL;
  }
  if ((iter = PyObject_GetIter(obj)) == NULL) {
    return NULL;
  }
  while ((item = PyIter_Next(iter)) != NULL) {
    /* Use item */
    ...
    Py_DECREF(item);
  }

  Py_DECREF(iter);
  return Py_BuildValue("");
}
```

讨论¶

本节中的代码和Python中对应代码类似。 `PyObject_GetIter()` 的调用和调用 `iter()` 一样可获得一个迭代器。 `PyIter_Next()` 函数调用 `next` 方法返回下一个元素或NULL(如果没有元素了)。 要注意正确的内存管理——`Py_DECREF()` 需要同时在产生的元素和迭代器对象本身上同时被调用， 以避免出现内存泄露。