Числовые массивы

Объяснение

Ранее в книге уже рассматривались списки (с. 63). В списках могут храниться данные разных типов, включая строки и числа. **Массивы** Python похожи на списки, но используются **только для хранения чисел**. В массиве все данные **должны относиться к одному типу данных** из перечисленных в следующей таблице.

Код типа	Общепринятое название	Описание	Размер в байтах
'i'	Целое число	Целое число в диапазоне от −32768 до 32767	2
'l'	Длинное целое число	Целое число в диапазоне от −2 147 483 648 до 2 147 483 648	4
'f'	Число с плавающей точкой	Числа в диапазоне от −1038 до 1038 с дробной частью (то есть число может содержать до 38 знаков, включая десятичную точку в любой позиции, и может быть как отрицательным, так и положительным)	4
'd'	Число двойной точности	Числа в диапазоне от −10 308 до 10 308 с дробной частью	8

При создании массива необходимо определить тип содержащихся в нем данных. Этот тип не может быть изменен во время работы программы. Следовательно, если вы определяете массив с типом `'i'` (что позволяет использовать в нем целые числа в диапазоне от −32 768 до 32 767), позже вы не сможете добавить десятичную точку в число, потому что это приведет к выдаче сообщения об ошибке и аварийному завершению программы.

Примечание: в других языках программирования термин «массив» обычно используется для хранения данных любого типа, но в массивах Python могут храниться только числа, тогда как списки подходят для хранения любых типов данных. Если вы хотите создать переменную для хранения нескольких строк, в Python для этого придется создать список вместо массива.

Примеры кода

```
from array import *
```

Эта строка должна находиться в самом начале программы, чтобы Python мог использовать библиотеку массивов.

```
nums = array ('i',[45,324,654,45,264])
print(nums)
```

Создает массив с именем **nums**. Массив использует целочисленный тип данных и состоит из пяти элементов. При выводе будет получен следующий результат:

```
array('i', [45, 324, 654, 45, 264])
```

```
for x in nums:
 print(x)
```

Выводит массив, при этом каждый элемент выводится в отдельной строке.

```
newValue = int(input("Enter number: "))
nums.append(newValue)
```

Предлагает пользователю ввести новое число, которое добавляется в конец существующего массива.

```
nums.reverse()
```

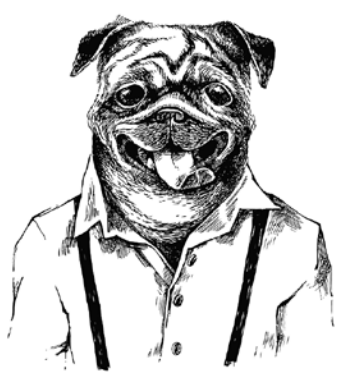
Переставляет элементы массива в обратном порядке.

```
nums = sorted(nums)
```

Сортирует массив по возрастанию.

```
nums.pop()
```

Удаляет последний элемент из массива.



```
newArray = array('i',[])
more = int(input("How many items: "))
for y in range(0,more):
 newValue=int(input("Enter num: "))
 newArray.append(newValue)
nums.extend(newArray)
```

Создает пустой массив с именем **newArray**, использующий целочисленный тип данных. Пользователю предлагается ввести количество элементов, после чего соответствующее количество элементов добавляется в **newArray**. После того как все элементы будут добавлены, содержимое массивов **newArray** и **nums** объединяется.

```
getRid = int(input("Enter item index: "))
nums.remove(getRid)
```

Предлагает пользователю ввести элемент, который удаляется из массива, после чего удаляет первый элемент массива, совпадающий с введенным значением.

```
print(nums.count(45))
```

Показывает, сколько раз значение **45** встречается в массиве.

Задачи

088

Предложите пользователю ввести пять целых чисел и сохраните их в массиве. Отсортируйте список и выведите его содержимое в обратном порядке.

089

Создайте массив для хранения целых чисел. Сгенерируйте пять случайных чисел и сохраните их в массиве. Выведите массив (каждый элемент должен выводиться в отдельной строке).

090

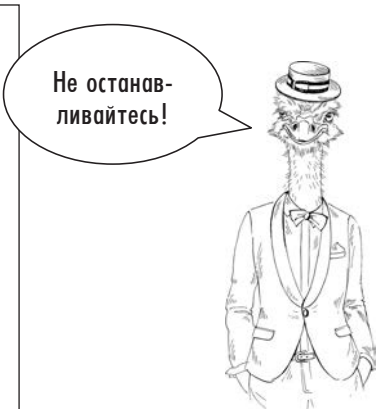
Предложите пользователю вводить целые числа. Если пользователь вводит число от 10 до 20, сохраните его в массиве; в противном случае выведите сообщение «Outside the range». После того как пять чисел будут успешно добавлены в массив, выведите сообщение «Thank you» и выведите массив, каждый элемент которого находился бы на отдельной строке.

091

Создайте массив, содержащий пять чисел (два из которых должны повторяться). Выведите весь массив. Предложите пользователю ввести одно из чисел массива, после чего выведите сообщение, в котором указано, сколько раз число встречается в этом массиве.

092

Создайте два массива: один будет содержать три числа, введенных пользователем, а другой — пять случайных чисел. Объедините эти два массива в один большой. Отсортируйте и выведите его, при этом каждое число должно выводиться в отдельной строке.



093

Предложите пользователю ввести пять чисел. Отсортируйте их и выведите для пользователя. Предложите выбрать одно из чисел. Удалите выбранное число из исходного массива и сохраните его в новом.

094

Выведите массив из пяти чисел. Предложите пользователю выбрать одно из них. После того как число будет выбрано, выведите его позицию в массиве. Если пользователь введет значение, отсутствующее в массиве, предложите ему выбрать снова, пока не будет выбрано допустимое значение.

095

Создайте массив из пяти чисел от 10 до 100, каждое из которых содержит два знака в дробной части. Предложите пользователю ввести целое число от 2 до 5. Если пользователь введет значение, выходящее за границы диапазона, выведите сообщение об ошибке и предложите выбрать снова, пока не будет введено допустимое значение. Разделите каждое из чисел в массиве на число, введенное пользователем, и выведите ответы с точностью до двух знаков.

Ответы

088

```
from array import *

nums = array('i', [])

for i in range(0, 5):
    num = int(input("Enter a number: "))
    nums.append(num)

nums = sorted(nums)
nums.reverse()

print(nums)
```

089

```
from array import *
import random

nums = array('i', [])

for i in range (0, 5):
    num = random.randint(1, 100)
    nums.append(num)

for i in nums:
    print(i)
```

090

```
from array import *

nums = array('i', [])

while len(nums) < 5:
    num = int(input("Enter a number between 10 and 20: "))
    if num >= 10 and num <= 20:
        nums.append(num)
    else:
        print("Outside the range")

for i in nums:
    print(i)
```

091

```
from array import *

nums = array('i', [5, 7, 9, 2, 9])

for i in nums:
    print(i)

num = int(input("Enter a number: "))

if nums.count(num) == 1:
    print(num, "is in the list once")
else:
    print(num, "is in the list ", nums.count(num),"times")
```

092

```
from array import *
import random

num1 = array('i', [])
num2 = array('i', [])

for i in range(0, 3):
    num = int(input("Enter a number: "))
    num1.append(num)

for i in range(0, 5):
    num = random.randint(1, 100)
    num2.append(num)

num1.extend(num2)

num1 = sorted(num1)

for i in num1:
    print(i)
```

093

```
from array import *

nums = array('i', [])

for i in range(0, 5):
    num = int(input("Enter a number: "))
    nums.append(num)

nums = sorted(nums)

for i in nums:
    print(i)

num = int(input("Select a number from the array: "))
if num in nums:
    nums.remove(num)
    num2 = array('i', [])
    num2.append(num)
    print(nums)
    print(num2)
else:
    print("That is not a value in the array")
```

094

```
from array import *

nums = array('i', [4, 6, 8, 2, 5])

for i in nums:
    print(i)

num = int(input("Select one of the numbers: "))

tryagain = True
while tryagain == True:
    if num in nums:
        print("This is in position ", nums.index(num))
        tryagain = False
    else:
        print("Not in array")
        num = int(input("Select one of the numbers: "))
```

095

```
from array import *
import math

num1 = array('f',[34.75, 27.23, 99.58, 45.26, 28.65])
tryagain = True
while tryagain == True:
    num = int(input("Enter a number between 2 and 5: "))
    if num <2 or num > 5:
        print("Incorrect value, try again.")
    else:
        tryagain = False
for i in range(0, 5):
    ans = nums[i] / num
    print(round(ans, 2))
```

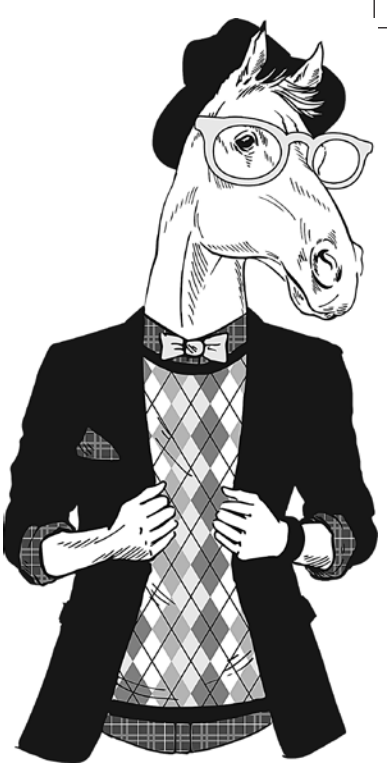
Двумерные списки и словари

Объяснение

С технической точки зрения в Python возможно создать двумерный массив, но так как массивы Python ограничиваются хранением чисел, а большинство программистов Python чувствует себя более уверенно при работе со списками, двумерные массивы используются редко, а **двумерные списки** встречаются гораздо чаще.

Представьте ужасную ситуацию: вы работаете учителем. Понимаю, это не для слабонервных! Также представьте, что у вас есть четыре ученика, и вы преподаете им три разных предмета. Вы как сознательный учитель ведете учет баллов этих учеников по всем предметам. На бумаге подобная таблица может выглядеть так:

	Математика	Английский	Французский
Сьюзен	45	37	54
Питер	62	58	59
Марк	49	47	60
Энди	78	83	62



Двумерные списки работают аналогичным образом:

	0	1	2
0	45	37	54
1	62	58	59
2	49	47	60
3	78	83	62

В Python двумерный список записывается так:

```
grades = [[45, 37, 54], [62, 58, 59], [49, 47, 60], [78, 83, 62]]
```

Если вы не хотите использовать стандартные числовые индексы столбцов Python, можно работать со словарем:

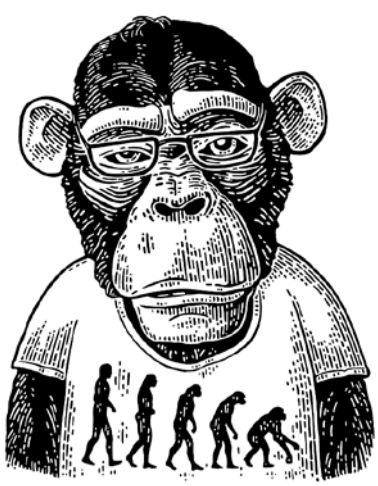
```
grades = [{"Ma":45, "En":37, "Fr":54}, {"Ma":62, "En":58, "Fr":59},
          {"Ma":49, "En":47, "Fr":60}]
print(grades[0]{"En"})
```

Программа выводит значение 37 (оценка для ученика с индексом 0 по предмету **"En"**) и упрощает понимание данных.

Можно пойти еще дальше и добавить символические индексы для строк:

```
grades = {"Susan":{"Ma":45, "En":37, "Fr":54}, "Peter":{"Ma":62, "En":58,
          "Fr":59}}
print(grades["Peter"]["En"])
```

Команда выводит значение 58 (оценка для ученика **"Peter"** по предмету **"En"**).



Примеры кода

```
simple_array = [[2, 5, 8], [3, 7, 4], [1, 6, 9]]
```

Создает двумерный список (изображенный справа) с использованием стандартных индексов Python для строк и столбцов.

	0	1	2
0	2	5	8
1	3	7	4
2	1	6	9

```
print(simple_array)
```

Выводит все данные в виде двумерного списка.

```
print(simple_array [1])
```

Выводит данные из строки 1 — в данном случае [3, 7, 4].

```
simple_array [2] [1] = 5
```

Присваивает элементу, находящемуся в строке 2 и столбце 1, значение 5.

```
print(simple_array [1] [2])
```

Выводит данные из строки 1 и столбца 2 — в данном случае 4.

```
simple_array[1].append(3)
```

Добавляет значение 3 в конец данных строки 1, так что в данном случае строка принимает вид [3, 7, 4, 3].

	x	y	z
A	54	82	91
B	75	29	80

```
data_set = {"A":{"x":54,"y":82,"z":91},"B":{"x":75,"y":29,"z":80}}
```

Создает двумерный список с использованием пользовательских меток для строк и столбцов (см. выше).

```
print(data_set ["A"])
```

Выводит информацию из набора данных **"A"**.

```
print(data_set ["B"] ["y"])
```

Выводит значение элемента из строки **"B"** и столбца **"y"**.

```
for i in data_set:
 print(data_set [i] ["y"])
```

Выводит элемент из столбца **"y"** каждой строки.

```
data_set ["B"] ["y"] = 53
```

Присваивает элементу из строки **"B"** и столбца **"y"** значение 53.

```
Grades [name] = {"Maths":mscore, "English":escore}
```

Добавляет еще одну строку данных в двумерный словарь. В этом случае **name** становится индексом строки, а **"Maths"** и **"English"** — индексами столбцов.

```
for name in grades:
 print((name), grades [name] ["English"])
```

Выводит только значение **name** и оценку по предмету **"English"** для каждого ученика.

```
del list [getRid]
```

Удаляет выбранный элемент.

Задачи

096

Создайте следующий набор данных в виде простого двумерного списка со стандартными индексами Python:

	0	1	2
0	2	5	8
1	3	7	4
2	1	6	9
3	4	2	0

097

Используя двумерный список из задачи 096, предложите пользователю выбрать строку и столбец и выведите выбранное значение.

098

Используя двумерный список из задачи 096, предложите пользователю выбрать строку и выведите только ее. Предложите ввести новое значение, добавьте его в конец строки, после чего снова выведите измененную строку.

099

Измените программу из задачи 098. Предложите пользователю выбрать строку и выведите только ее. Предложите выбрать столбец из выведенной строки и выведите только хранящееся там значение. Спросите, хочет ли пользователь изменить его. Если ответ будет положительным, предложите ввести новое значение и измените данные. Наконец, снова выведите измененную строку.

100

Создайте следующий набор данных, представляющий объемы продаж по регионам, в виде двумерного словаря:

	N	S	E	W
John	3056	8463	8441	2694
Tom	4832	6786	4737	3612
Anna	5239	4802	5820	1859
Fiona	3904	3645	8821	2451

101

Используя программу из задачи 100, запросите у пользователя имя и регион. Выведите соответствующие данные. Запросите у пользователя имя и регион того значения, которое он хочет изменить, и позвольте скорректировать объем продаж. Выведите объемы продаж по всем регионам для имени, выбранного пользователем.

102

Предложите пользователю ввести имя, возраст и размер обуви для четырех человек. Запросите имя одного из них в списке и выведите значения его возраста и размера обуви.

103

Измените программу 102, чтобы она выводила имя и возраст для всех людей в списке, но не их размер обуви.

104

После получения имени, возраста и размера обуви для четырех человек запросите у пользователя имя человека для удаления из списка. Удалите эту строку и выведите остальные данные с разбивкой по строкам.