

§ 1.2

Организация ввода и вывода данных

Ключевые слова:

- оператор вывода `print()`
- формат вывода
- оператор ввода `input()`

1.2.1. Вывод данных

В предыдущем параграфе мы познакомились с типами данных и рассмотрели оператор присваивания. Этого достаточно для того, чтобы записать программу преобразования данных. Но результат этих преобразований нам виден не будет.

Для вывода данных из оперативной памяти на экран компьютера используется **оператор (функция) вывода** `print()`:

```
print(<выражение 1>, <выражение 2>, ..., <выражение N>)
```

Здесь в круглых скобках помещается список вывода — список выражений, значения которых выводятся на экран. Это могут быть числовые, символьные и логические выражения, в том числе константы и переменные.

Пример. Оператор `print('s=', s)` выполняется так:

- 1) на экран выводятся символы, заключённые в одинарные кавычки: `s=`
- 2) на экран выводится значение переменной `s` с именем `s`.

Если значение переменной `s` равно `15`, и она имеет целочисленный тип, то на экране появится: `s=15`

Обратите внимание: по умолчанию выводимые выражения разделяются одним пробелом, иначе говоря, разделителем между ними является пробел. Оператор `print()` вставляет между выводимыми значениями так называемый разделитель (сепаратор, от англ. *separator*). Мы можем его изменять, указывая новый разделитель после слова `sep`.

Вариант организации вывода	Оператор (функция) вывода	Результат
По умолчанию	<code>print(1, 20, 300)</code>	1 20 300
Убрать разделители-пробелы	<code>print(1, 20, 300, sep='')</code>	120300
Добавить разделитель-запятую	<code>print(1, 20, 300, sep=',')</code>	1, 20, 300
Вывод каждого значения с новой строки	<code>print(1, 20, 300, sep='\n')</code>	1 20 30

Предположим, что мы работаем с натуральными числами, каждое из которых меньше 100. Тогда на одно число на экране достаточно выделить 3 позиции: две позиции на запись самого числа и ещё одну позицию на пробел слева, разделяющий числа. Записывается это так:

```
print("{:3}{:3}{:3}".format(a, b, c))
```

Это **форматный вывод**: строка для вывода строится с помощью функции `format`. Аргументы этой функции (`a`, `b` и `c`) — это те величины, которые выводятся; они указываются в круглых скобках.

Символьная строка слева от точки — это форматная строка, которая определяет **формат вывода**, т. е. как именно величины будут представлены на экране. Фигурные скобки обозначают место для вывода очередного элемента; число после двоеточия — количество позиций, которые отводятся на число; на первом месте выводится значение `a`, на втором — значение `b`, на третьем — `c`. Если цифр в числе меньше, чем зарезервированных под него позиций на экране, то свободные позиции дополняются пробелами слева от числа. Если указанное в формате вывода число меньше, чем необходимо, то оно автоматически будет увеличено до минимально необходимого.

Например, числа 12, 5 и 15 будут выведены так:

```
12 5 15
```

Числа 12, 5 и 1500 будут выведены следующим образом:

```
12 51500
```

Для вывода вещественного числа в списке вывода для каждого выражения указываются два параметра:

- 1) общее количество позиций, отводимых на число;
 - 2) количество позиций в дробной части числа:
- d — целые числа (int);
 - f — вещественные (float);
 - e — экспоненциальный формат.

Фрагмент программы	Результат выполнения оператора вывода
<pre>a = 4 print ("a=", "{:5d}{:5d}".format(a, a * a))</pre>	a= 4 16
<pre>a = 1 / 3; b = 1 / 9 print ("{:7.3f}{:7.4f}".format(a, b))</pre>	0.333 0.1111
<pre>a = 1 / 3 print ("{:10.3e}".format(a))</pre>	3.333e-01

При выполнении очередного оператора `print()` по умолчанию вывод продолжается в новой строке. Чтобы убрать переход к новой строке, используется параметр `end`:

```
print(a, end="") # убран переход на новую строку
print(b)
```

1.2.2. Первая программа на языке Python

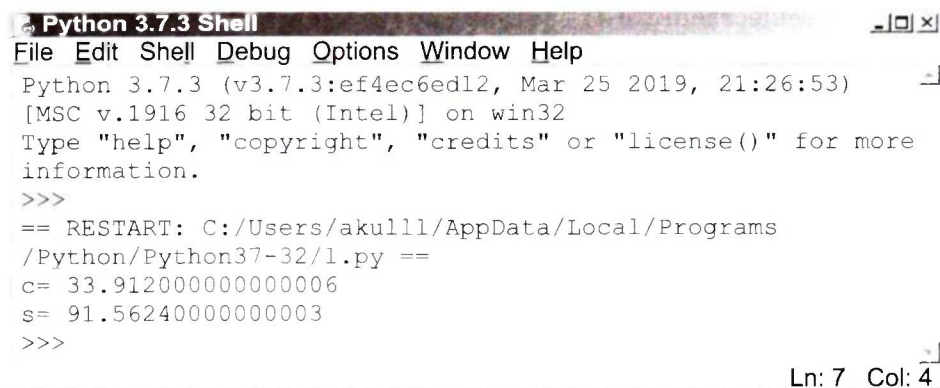
Пользуясь рассмотренными операторами, составим программу, вычисляющую длину окружности и площадь круга с радиусом 5,4 см.

Исходным данным в этой задаче является радиус: $r = 5,4$ см. Результатом работы программы должны быть величины c и s : c — длина окружности и s — площадь круга. c , s и r — величины вещественного типа.

Исходное данное и результаты связаны соотношениями, известными из курса математики: $c = 2\pi r$, $s = \pi r^2$. Программа, реализующая вычисления по этим формулам, будет иметь вид:

```
# Программа 1
r = 5.4
c = 2 * 3.14 * r
s = 3.14 * r ** 2
print ('c=', c)
print ('s=', s)
```

Эта программа верна и решает поставленную задачу. Запустив её на выполнение, мы получим следующий результат (рис. 1.2).



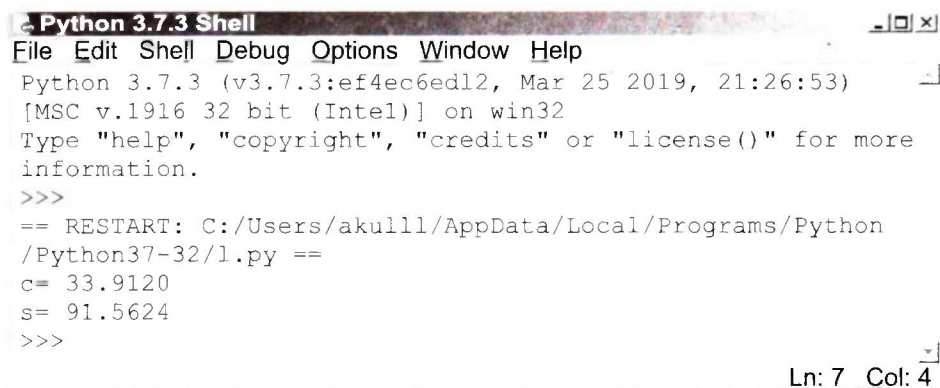
```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53)
[MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>>
== RESTART: C:/Users/akull11/AppData/Local/Programs
/Python/Python37-32/1.py ==
c= 33.912000000000006
s= 91.562400000000003
>>>
```

Ln: 7 Col: 4

Рис. 1.2. Результат вычисления длины окружности и площади круга

Улучшим внешний вид результата, используя вывод по формату (рис. 1.3).

```
print("c=", "{:7.4f}".format(c))
print("s=", "{:7.4f}".format(s))
```



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53)
[MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>>
== RESTART: C:/Users/akull11/AppData/Local/Programs/Python
/Python37-32/1.py ==
c= 33.9120
s= 91.5624
>>>
```

Ln: 7 Col: 4

Рис. 1.3. Форматный вывод

И всё-таки составленная нами программа имеет существенный недостаток: она находит длину окружности и площадь круга для единственного значения радиуса (5,4 см).

Для того чтобы вычислить длину окружности и площадь круга для другого значения радиуса, потребуется вносить изменения