Теоретический материал

Теоретический материал

Сайт: Дистанционная подготовка
Курс: Д. П. Кириенко. Программирование на языке Python (школа 179 г. Москвы)
Book: Теоретический материал
Printed by: maung myo
Date: Воскресенье 4 Март 2018, 01:37

Table of Contents

Запуск простейшей программы

В предыдущем задании мы использовали Питон для простых разовых вычислений, используя интерактивный режим. Например, было задание вычислить длину гипотенузы прямоугольного треугольника по ее катетам. Запустите текстовый редактор и напишите следующий текст:

```
a = 179
b = 197
c = (a ** 2 + b ** 2) ** 0.5
print (c)
```

Здесь мы используем **переменные** — объекты, в которых можно сохранять различные (числовые, строковые и прочие) значения. В первой строке переменной a присваивается значение 179, затем переменной b присваивается значение 971, затем переменной c присваивается значение арифметического выражения, равному длине гипотенузы.

После этого значение переменной c выводится на экран.

Сохраните этот текст в файле с именем hypot.py. Запустите терминал, перейдите в каталог, где лежит этот файл и выполните эту программу:

```
$ python3 hypot.py
```

Интерпретатор языка Питон, запущенный с указанием имени файла, запускается не в интерактивном режиме, а выполняет ту последовательность команд, которая сохранена в файле. При этом значения вычисленных выражений не выводятся на экран (в отличии от интерактивного режима), поэтому для того, чтобы вывести результат работы программы, то есть значение переменной c, нам понадобится специальная функция print.

Ввод данных: функция input()

Пример выше неудобен тем, что исходные данные для программы заданы в тексте программы, и для того, чтобы использовать программу для другого треугольника необходимо исправлять текст программы. Это неудобно, лучше, чтобы текст программы не менялся, а программа запрашивала бы у пользователя данные, необходимые для решения задачи, то есть запрашивала бы значения двух исходных переменных a и b. Для этого будем использовать функцию input(), которая считывает строку с клавиатуры и возвращает значение считанной строки, которое сразу же присвоим переменым a и b:

```
a = input()
b = input()
```

Правда, функция input возвращает текстовую строку, а нам нужно сделать так, чтобы переменные имели целочисленные значения. Поэтому сразу же после считывания выполним преобразование типов при помощи фунцкии int, и запишем новые значения в переменные a и b.

```
a = int(a)
b = int(b)
```

Можно объединить считывание строк и преобразование типов, если вызывать функцию int для того значения, которое вернет функция input:

```
a = int(input())
b = int(input())
```

Далее в программе вычислим значение переменной c и выведем результат на экран.

Теперь мы можем не меняя исходного кода программы многократно использовать ее для решения различных задач. Для того нужно запустить программу и после запуска программы ввести с клавиатуры два числа, нажимая после кажого числа клавишу Enter. Затем программа сама выведет результат.

Вывод данных: функция print()

Функция print может выводить не только значения переменных, но и значения любых выражений. Например, допустима запись print(2 + 2 ** 2). Также при помощи функции print можно выводить значение не одного, а нескольких выражений, для этого нужно перечислить их через запятую:

```
a = 1
b = 2
print(a, '+', b, '=', a + b)
```

В данном случае будет напечатан текст 1 + 2 = 3: сначала выводится зание переменной a, затем строка из знака “+”, затем значение переменной b, затем строка из знака “=”, наконец, значение суммы a + b.

Обратите внимание, выводимые значение разделяются одним пробелом. Но такое поведение можно изменить: можно разделять выводимые значения двумя пробелами, любым другим символом, любой другой строкой, выводить их в отдельных строках или не разделять никак. Для этого нужно функции print передать специальный именованный параметр, называемый sep, равный строке, используемый в качестве разделителя (sep — аббревиатура от слова separator, т.е. разделитель). По умолчанию параметр sep равен строке из одного пробела и между значениями выводится пробел. Чтобы использовать в качестве разделителя, например, символ двоеточия нужно передать параметр sep, равный строке ':':

```
print(a, b, c, sep = ':')
```

Аналогично, для того, чтобы совсем убрать разделитель при выводе нужно передать параметр sep, равный пустой строке:

```
print(a, '+', b, '=', a + b, sep = '')
```

Для того, чтобы значения выводились с новой строке, нужно в качестве параметра sep передать строку, состоящую из специального символа новой строки, которая задается так:

```
print(a, b, sep = '\n')
```

Символ обратного слэша в текстовых строках является указанием на обозначение специального символа, в зависимости от того, какой символ записан после него. Наиболее часто употребляется символ новой строки '\n'. А для того, чтобы вставить в строку сам символ обратного слэша, нужно повторить его два раза: '\\'.

Вторым полезным именованным параметром функции print является параметр end, который указывает на то, что выводится после вывода всех значений, перечисленных в функции print. По умолчанию параметр end равен '\n', то есть следующий вывод будет происходить с новой строки. Этот параметр также можно исправить, например, для того, чтобы убрать все дополнительные выводимые символы можно вызывать функцию print так:

```
print(a, b, c, sep = '', end = '')
```

Целочисленная арифметика

Для целых чисел определены ранее рассматривавшиеся операции +, -, * и **. Операция деления / для целых чисел возвращает значение типа float. Также функция возведения в степень возвращает значение типа float, если показатель степени — отрицательное число.

Но есть и специальная операция целочисленного деления, выполняющегося с отбрасыванием дробной части, которая обозначается //. Она возвращает целое число: целую часть частного. Например:

```
>>> 17 // 3
5
>>> -17 // 3
-6
```

Другая близкая ей операция: это операция взятия остатка от деления, обозначаемая %:

```
>>> 17 % 3
2
>>> -17 % 3
1
```