**Переменные и арифметические выражения**

**Материалы для самостоятельного изучения: Переменные**

В некоторых задачах вычисления удобно проводить используя вспомогательные переменные. Например, в школьных формулах по физике было удобно вычислять не гигантское выражение целиком, а запоминая результаты вычисления во вспомогательные переменные. Для примера решим задачу вычисления пройденного расстояния по известному времени и скорости:

|  |  |
| --- | --- |
| **1** | **speed = 108** |
| **2** | **time = 12** |
| **3** | **dist = speed \* time** |
| **4** | **print(dist**) |

В этой программе мы создаем три переменные: speed для скорости, time для времени и dist для вычисленного расстояния. При использовании переменых в арифметическом выражении просто используется значение, которое лежит в переменной.

Для присваивания значения переменной используется знак =. Имя переменной должно быть записано слева от знака присваивания, а арифметическое выражение (в котором могут быть использованы числа и другие уже заданные переменные) - справа. Имя переменной должно начинаться с маленькой латинской буквы, должно быть осмысленным (анлийские слова или общеупотребимые сокращения) и не должно превышать по длине 10-15 символов. Если логичное имя переменной состоит из нескольких слов, то нужно записывать его с помощью camelTyping (каждое новое слово кроме первого должно быть записано с большой буквы).

Подбробнее о том, как осуществляется присваивание будет описано ниже.

**Переменные и арифметические выражения**

**Материалы для самостоятельного изучения: Арифметические выражения**

Мы уже использовали арифметические выражения в наших программах, в частности операции + и \*. Также существует ряд других арифметических операций, которые приведены в таблице:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Знак** | **Операция** | **Операнд1** | **Операнд2** | **Результат** |
| **+** | **Сложение** | **11** | **6** | **17** |
| **-** | **Вычитание** | **11** | **6** | **5** |
| **\*** | **Умножение** | **11** | **6** | **66** |
| **//** | **Целочисленное деление** | **11** | **6** | **1** |
| **%** | **Остаток от деления** | **11** | **6** | **5** |
| **\*\*** | **Возведение в степень** | **2** | **3** | **8** |

Все операции инфиксные (записываются между операндами), т.е., например, для возведения 2 в степень 3 нужно писать 2\*\*3.

Особо остановимся на операциях вычисления целой части и остатка от деления от числа.

Пусть заданы два числа A и B, причем B > 0. Обозначим за C целую часть от деления A на B, C = A // B, а за D - остаток от деления A на B, D = A % B.

Тогда должны выполняться следующие утверждения:

A = B × C + D

0 ≤ D < B

Эти утверждения необходимы для понимания процесса взятия остатка от деления отрицательного числа на положительное. Нетрудно убедиться, что если -5 разделить на 2, то целая часть должна быть равна -3, а остаток равен 1. В некоторых других языках программирования остатки в такой ситуации могут быть отрицательными, что неправильно по математическим определениям.

В случае, если B < 0 выполняются следующие утверждения:

A = B × C + D

B < D ≤ 0

Например, при делении 11 на -5 мы получим целую часть равную -3, а остаток будет равен -4.

Если же разделить -11 на -5, то целая часть будет равна 2, а остаток будет равен -1.

Обратите внимание, что целые числа в Питоне не имеют ограничений на длину (кроме объема доступной памяти).