Урок 2.  
Арифметические операции с числами и строками.

короткая линия

# План урока

1. Целые и вещественные числа
2. Знаки математический действий
3. Операции над строками
4. Сравнение чисел и строк
5. Простейшие функции

# Целые и вещественные числа

Для представления чисел в Python используется три типа данных: целые числа (обозначаются int), вещественные, или дробные, числа (обозначаются float) и комплéксные числа (в данном курсе рассматриваться не будут). Важно помнить, что дробные числа записываются через точку, а не через запятую. Также Python позволяет преобразовывать строки в числа. Это делается одноименными командами: int(), float() и complex(). Например, вы можете перевести строку “5” в целое число и дальше применять к нему арифметические действия.

print(int(**"5"**))

“5” теперь будет восприниматься интерпретатором как целое число. Теперь попробуйте преобразовать другое число. Поэкспериментируйте с этими командами. Например, что получится, если написать

print(int(**"a"**))

print(int(995.23))

Или

|  |
| --- |
| print(float(42)) |

Все символы, введенные с клавиатуры, изначально воспринимаются как строки. Поэтому, для того чтобы программа корректно работала с введенными числами, необходимо пользоваться командами выше. Например, так:

|  |
| --- |
| n = int(input())  print(n) |

# Знаки математических действий

Python, как и любой язык программирования, поддерживает арифметические операции, хорошо вам известные. Ниже представлены их обозначения.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Знак | Арифметическая операция | Первое число | Второе число | Результат |
| + | Сложение | 3 | 2 | 5 |
| - | Вычитание | 3 | 2 | 1 |
| \* | Умножение | 3 | 2 | 6 |
| / | Деление | 3 | 2 | 1.5 |
| \*\* | Возведение в степень | 3 | 2 | 9 |
| % | Остаток от деления | 3 | 2 | 1 |
| // | Целочисленное деление | 3 | 2 | 1 |

Данные операции применимы ко всем числам в Python: как к целым (тип данных int), так и к дробным (тип данных float). Вы можете проверить каждую операцию из таблицы, взяв любое количество чисел. Python поддерживает “длинную арифметику”, поэтому вы с легкостью можете проверить значение такого выражения как, например:

print(2 \*\* 2018)

Поэкспериментируйте с вычислениями и приступайте к выполнению задач (одна задача — на повторение предыдущей темы):

**Задача “Обновленный словарь”**

**Задача “Сложение чисел”**

**Задача “Площадь прямоугольника”**

**Задача “Перевод в метры”**

**Задача “Пример из физики”**

**Задача “Цени свое время”**

Вполне возможно, что данные задачи не вызвали у вас никаких трудностей (в таком случае, будем считать, что уроки математики не прошли даром). Однако в таблице есть операции, которые могут быть не совсем привычны для вас: взятие остатка и целой части от деления. Если вы знакомы с языком Pascal, то вы поймете, что эти обозначения эквивалентны словам mod и div в этом языке.

Ниже несколько прикладных задач, на примерах которых можно удостовериться, что эти действия часто могут пригодиться.

**Задача “Последняя цифра числа”**

**Задача “Который час?”**

После того, как вы потренировались использовать % и //, решите следующую задачу:

**Задача “Последние n цифр числа”**

# Операции над строками

Вернемся к строкам. Оказывается, строки тоже поддерживают некоторые арифметические операции. Применение одной из них — умножения — вы уже видели в примере из предыдущего урока. Поговорим об этом подробнее.

В отличие от чисел, в Python строки можно только складывать (сложение строк называется операцией конкатенации), а также умножать строки на число. Попробуйте выполнить следующий код:

print(**"Yin"** + **" "** + **"Yang"**)

print(**"because"** \* 42)

Как вы, возможно, поняли, последние задачи предыдущей домашней работы можно было также решить с помощью конкатенации строк, передав print() лишь одну строку, полученную из нескольких.

Теперь попробуйте сложить строку с числом (в некоторых языках программирования — например, Java, — такое бы получилось, и число автоматически привелось бы к строке) или умножить две строки. Посмотрите, что выведется в консоль.

Чтобы получить строковое представление числа, аналогично команде int() используется str(). Так, код

print(int((str(2) \* 100)) \*\* 2)

возведет в квадрат число, состоящее из ста двоек.

int() и str() во многих задачах используются в связке.

**Задача “Сложное выражение”**

**Задача “Сумма кубов цифр”**

# Сравнение чисел и строк

Кроме вычислений, над числами, как и над строками, можно производить операции сравнения.

|  |  |
| --- | --- |
| Знак | Операция сравнения |
| == | Равно |
| != | Не равно |
| > | Больше |
| < | Меньше |
| >= | Больше или равно |
| <= | Меньше или равно |

Если вы выведете результат сравнения, в консоль выведется один из двух вариантов: слово True или False, означающие инстинность или ложность написанного вами выражения. Подробнее о bool переменных мы будем говорить позже.

Допустим, что применимо к числам все вышеперечисленные действия интуитивно понятны. Но как работают операции >, <, >=, <= для строк?

При сравнении строк принимаются во внимание как символы, так и их регистр. Цифровой символ считается меньше любого алфавитного символа. Алфавитный символ в верхнем регистре меньше алфавитных символов в нижнем регистре. Python начинает сравнивать строки с первого слева символа. Большей считается та строка, первый несовпадающий символ которой больше. Соответственно, строки равны, если их длины равны и их символы совпадают.

**Задача “Простое сравнение”**

# Простейшие функции

Настало время ввести новое важное понятие — функция. В языках программирования функция похожа на функцию в математике: грубо говоря, функция — это команда языка, которая принимает “внутрь” какие-то параметры (аргументы), выполняет какие-то действия над аргументами и возвращает результат этого действия. Все команды, которые мы выучили до сих пор — print(), input(), int(), float(), str() — являются стандартными функциями языка Python. Заметьте, что они похожи по синтаксису: все они принимают значения в круглые скобки и выполняют определенные действия.

Познакомимся с тремя важными стандартными функциями:

1. len() — принимает на вход строку, возвращает число — длину строки.
2. max() — принимает на вход набор чисел, возвращает число — максимальный элемент из набора.
3. min() — принимает на вход набор чисел, возвращает число — минимальный элемент из набора.

Потренируйтесь с использованием данных функций (попробуйте передать разные параметры) и сдайте следующие задачи:

**Задача “Простая длина”**

**Задача “Простой максимум”**

**Задача “Простой минимум”**

Данный урок ввел множество новых концептов. Важно понять и научиться использовать каждый из них. Помните, что в сети есть множество ресурсов, способных объяснить любую тему, а также не забывайте задавать интересующие вас вопросы преподавателю.

**Домашнее задание:**

**Задача “Площадь треугольника”**

**Задача “Сколько прошло времени”**

**Задача “Симулятор столовой”**

**Задача “Цена пирожка”**

**Задача “18!”**

**Сложная задача “Улитка”**