Иногда программам необходимо взаимодействовать с пользователями, либо для получения некоторых данных, либо для получения какого-либо результата. И вот тогда input()функция крадет представление.

**Чтение входных данных от пользователя**

Входные данные, которые мы хотим получить, - это не что иное, как некоторое значение, введенное пользователем. input()Функция считывает это значение и возвращает его в программе в виде строки. Например, следующая программа считывает имя пользователя и печатает приветствие.

user\_name = input()

print('Hello, ' + user\_name)

В первой строке программа будет ждать, пока пользователь введет что-то в качестве входных данных. Мы присвоим этот ввод переменной, чтобы сохранить его на потом. Во второй строке программа добавляет введенное имя в конец 'Hello, 'строки и в результате выводит всю фразу целиком.

Если пользователь вводитSauron, эта программа печатает:

Hello, Sauron

Итак, ваша программа выводит результат, который зависит от ввода пользователя (имя).

**Очистка сообщений**

Настоятельно рекомендуется четко указать, какого типа ввода мы ожидаем от наших пользователей. Для этого input()функция может принимать необязательный аргумент, то есть сообщение:

user\_name = input('Please, enter your name: ')

print('Hello, ' + user\_name)

Программа запускается, пользователь видит сообщение, вводит свое имя и получает следующий результат:

Please, enter your name: Sauron

Hello, Sauron

Другой способ сделать это - распечатать сообщение отдельно:

print('Enter your name: ')

user\_name = input()

print('Hello, ' + user\_name)

На самом деле большой разницы нет: в предыдущем примере ввод будет напечатан в той же строке, что и сообщение, в то время как в этом случае он будет записан в следующей строке. Итак, вы можете выбрать все, что вам нравится.

Хотя рекомендуется печатать сообщения для пользователей, избегайте их в наших образовательных задачах по программированию, иначе ваш код может не пройти наши тесты.

**Важные детали**

Давайте углубимся в некоторые детали.

Прежде всего, как долго пользователь может вводить данные и как программа понимает, что человек ввел все, что хотел? Вот что касается input() функции: как только программа начала выполнять эту функцию, она останавливается и ожидает, пока пользователь введет некоторое значение и **нажмет Ввод**. Это также означает, что если пользователь не вводит данные, программа больше не будет выполняться.

Что еще вы должны запомнить? Ну, это: любое введенное вами значение функция воспринимает как **строку**. Не имеет значения, вводите ли вы цифры или буквы, ввод будет преобразован в строку.

Если вы хотите, чтобы число было **числом**, вы должны написать его явно:

print("What's your favorite number?")

value = int(input()) # now value keeps an integer number

Однако будьте осторожны: в этих обстоятельствах, если пользователь вводит нецелочисленное значение, Error появится.

Чтобы прочитать несколько входных данных, вы должны вызвать функцию более одного раза:

day = int(input()) # 4

month = input() # October

Блестяще! Почему эта дата? Это просто:

print('Cinnamon roll day is celebrated on', month, day)

# Cinnamon roll day is celebrated on October 4

**Заключение**

Поздравляем, теперь вы знаете, как работать с input(), то есть с функцией, которая помогает вам взаимодействовать с пользователем. Поверьте нам, это то, что вы определенно оцените при программировании. Вот что вы узнали:

* нет ограничений на длину входных данных, функция будет ждать, пока пользователь не нажмет Enter;
* вы можете добавить сообщение для пользователя вместе с запросом на ввод;
* функция интерпретирует любое введенное значение как строку;
* входные данные могут быть позже преобразованы в нужный вам тип данных.