# Урок 06.01. Сравнение строк

Последнее время мы постоянно работает с числами, и совсем забыли про строки, а ведь также могут встречаться нам при программировании.

Давайте возьмем нашу старую задачу: ввод пароля в виде числа, и если он не равен 235, то в цикле while продолжаем его запрашивать. Давайте вспомним эту задачу.

password = int(input(‘Введите пароль: ’))

while password != 235:

print(‘Не верный пароль’)

password = int(input(‘Введите пароль ещё раз: ’))

print(‘Пароль верный!’)

По правде говоря, мало у кого из нас пароль состоит только из цифр. Более того при регистрации на некоторых сайтах просят чтобы пароль состоял из букв и цифр. Давайте попробуем так сделать. Для начала, если мы просто уберем преобразование строки в число (в строчке программы, где стоит input), то мы вообще никак не сможем угадать пароль, даже «235» перестанет действовать. А всё потому, что число 235 и строка «235» это разные вещи и они не равны друг другу. Чтобы доработать нашу программу, необходимо «235» просто взять в кавычки. Теперь мы можем добавить буквы в наш пароль. Например, «Python235». Теперь у нас есть, и маленькие, и большие буквы, и даже цифры. Обращаю внимание, что регистр имеет значение.

А теперь еще одну задачу. К уроку литературы нужно было прочитать произведение «Евгений Онегин». И Мария Ивановна задает вопрос пяти случайным ученикам: «Кто написал произведение?». Так вот пока ученики не угадают автора Мария Ивановна ставит двойки, а как только первый ученик угадал, то Мария Ивановна перестает задавать вопрос и продолжает урок. Наша задача посчитать количество двоечников. Вот как будет выглядеть наша программа.

badGradeCount = 0

for student in range(5):

answer = input(‘Кто написал произведение: ’)

if answer == ‘Пушкин’:

break

print(‘Не правильно! Два!’)

badGradeCount += 1

print(‘Кол-во двоек: ’, badGradeCount)

Запустим программу. Введем два раза не правильно и потом правильно. Отлично. Получили ответ. А что если ввести правильный ответ с маленькой буквы. Наша программа не сработает – исправим это.

badGradeCount = 0

for student in range(5):

answer = input(‘Кто написал произведение: ’)

if (answer == ‘Пушкин’) or (answer == ‘пушкин’):

break

print(‘Не правильно! Два!’)

badGradeCount += 1

print(‘Кол-во двоек: ’, badGradeCount)

Вот теперь у нас программа работает получше.

Хорошо. Двигаемся дальше.

Может показаться не логичным изучение строковых переменных после изучения цикла for. Но это немного не так и сейчас мы поймем почему.

Рассмотрим такую простую с виду задачу. Наш любимый учитель захотел, чтобы символы слова «Python!» выводились в отдельной строке. И вот тут нам может понадобиться цикл «for». Если вспомнить наше первое знакомство с циклом «for», то мы просто писали перечисление необходимых чисел. Так вот строка точно такой же список, по которому цикл «for» будет идти. Тогда наша программа будет выглядеть так:

for symbol in ‘Python!’:

print(symbol)

Вот и вся программа. Заметим, что нам не нужно писать каждый символ в кавычках. А если нам нужно выводить строчку введенную пользователем, то:

phrase = input(‘Введите фразу: ’)

for symbol in phrase:

print(symbol)

А теперь наш учитель хочет, чтобы каждый символ в строке выводился 5 раз подряд. Конечно мы можем просто сделать конкатенацию символов. Но есть способ проще – нужно всего лишь умножить наш символ на 5.

phrase = input(‘Введите фразу: ’)

for symbol in phrase:

print(symbol \* 5)

И! Всё получилось.

Дальше решим другую задачу. Мы работаем в лингвистическом отделе и нам нужно посчитать частоту, с которой встречаются буквы в тексте. Нас будет интересовать всего лишь две буквы в тексте. Давайте напишем программу.

text = input(‘Введите текст: ’)

first\_symbol = input(‘Введите первый символ: ’)

second\_symbol = input(‘Введите второй символ:’)

first\_symbol\_count = 0

second\_symbol\_count = 0

for symbol in text:

if first\_symbol == smbol:

first\_symbol\_count += 1

if second\_symbol == smbol:

second\_symbol\_count += 1

print(‘Символ’, first\_symbol, ‘встречается’, first\_symbol\_count)

print(‘Символ’, second\_symbol, ‘встречается’, second\_symbol\_count)

Давайте запустим программу и проверим. Хорошо. Всё работает.

А теперь выполните задания.