**Строковый тип данных.**

Значением строковой величины (тип **str**) является произвольная последовательность символов, заключенная в одинарные или двойные кавычки. Символьная строка рассматривается как единый объект.

Символом в языке Python является любой из символов, который можно получить на экране нажатием на клавиатуре одной из клавиш или комбинации клавиш, а также некоторых других символов, в том числе и невидимых.

В тексте программы переменную строкового типа можно задать, заключив цепочку символов в одинарные или двойные кавычки:

**d = ‘5’**

**c = ‘Book’**

**c1 = “1\*”**

Новое значение может быть записано в строку с помощью оператора ввода с клавиатуры:

**s = input ()**

Если значение символьной переменной считывается с клавиатуры, то его следует набирать без апострофов. Встроенная функция 1еп определяет длину строки — количество символов в ней:

**n = len(s)**

Прим:

**a = “Привет”**

**b = input() << “Hello”**

**print(len(a)) >> 6**

**print(len(b)) >> 5**

Можно проверить равенство (совпадение) строк (**d == с**) или выяснить, какая из двух строк меньше (при этом используется поочерёдное сравнение кодов символов, образующих слова; меньшим будет то слово, у которого код очередного символа окажется меньше).

Чтобы найти код символа, используют функцию **ord()**, где в качестве параметра указывают символ.

Чтобы по коду узнать символ, используют функцию **chr()**, где в качестве параметра указывают код символа.

В Python (как и в алгоритмическом языке) строки можно сцеплять: а + b (к концу строки а прикрепляется («приписывается») строка Ь).

Пусть

**d = ‘1\*’ + ‘5’ + ‘1\***

Тогда в переменную **d** будет записана следующая строка: **'1\*51\*'**

В результате операции **а \* к**, где **к** — целое число, строка a повторяется к раз. Так, в результате выполнения команды

**print(d \* 3)**

будет выведена строка: **1\*51\*1\*51\*1\*51\***

**Логический тип данных.**

Как известно, величины логического типа принимают всего два значения; в Python это **False** и **True**. Эти константы определены так, что **False < True**.

Логические значения получаются в результате выполнения операций сравнения числовых, строковых и логических выражений. Поэтому в Python логической переменной можно присваивать результат операции сравнения.

Пусть **ans** — логическая переменная, а **N** — переменная целого типа. Тогда в результате выполнения оператора присваивания **ans = N % 2 == 0** переменной **ans** будет присвоено значение **True** при любом чётном **N** и **False** в противном случае.

**print('Определение истинности высказывания о чётности числа')**

**N = int(input('Введите исходное число»1))**

**ans = N % 2 == 0**

**print('Число', N, 'является чётным ans)**

Логическим переменным также можно присваивать значения логических выражений, построенных с помощью известных вам логических операций И, ИЛИ, НЕ, которые в Python обозначаются соответственно **and, or, not**.

Пример. Напишем программу, определяющую истинность высказывания «Треугольник с длинами сторон **а**, **b**, **с** является равнобедренным» для произвольных целых чисел **а**, **b**, **с**.

**print ('Определение истинности высказывания о равнобедренном треугольнике')**

**а = int(input('Введите значение а>>'))**

**b = int(input('Введите значение b>>'))**

**с = int (input ('Введите значение с» ' ) )**

**ans = (а == b) or (а == с) or (b == с)**

**print('Треугольник с длинами сторон', а, ',', b, ', ', с,' является равнобедренным ans)**

**Стр.6.**

Укажите истинное высказывание.

а) 3 > 2 and 5 > 6 == True

б) 'а' < 'b' or 1 > 0 == True

в) 60 < 20 == True

г) 2 // 3 == 1

**Стр.7.**

Выберите вариант, в котором написан результат выполнения программы

a, b, c, d = ‘1’, ‘2’, ‘3’, ‘4’

a += ‘12’ #112

b \*= 3 # 222

c += a + c # 31123

print(a + b + c + d)

1. 112222311234
2. 11222211234
3. 122331234
4. 1122333311234