Числа, строки, условия

Операции с числами и строками

Переменные в Python

Переменная хранит определенные данные.

Название переменной в Python должно начинаться с алфавитного символа или со знака подчеркивания и может содержать алфавитно-цифровые символы и знак подчеркивания.

Название переменной не должно совпадать с названием ключевых слов языка Python.

Ключевые слова

```
False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not
```

Задание переменных

name = ‘Tom’

age = 16

x = 15.2

Функция input()

Функция input() в Python отвечает за ввод в программу данных с клавиатуры.

Когда вызывается эта функция, программа останавливает свое выполнение и ждет, когда пользователь введет текст. После этого, когда он нажмет Enter, input() заберет введенный текст и передаст его программе, которая уже будет обрабатывать его согласно своим алгоритмам.

Взаимодействие с input()

Данные, полученные функцией input(), можно присвоить переменной:

```
name = input(‘Enter your name, please’)
```

Типы данных в Python

boolean - логическое значение True или False

int - представляет целое число, для хранения которого использует 4 байта в памяти компьютера.

float - представляет число с плавающей точкой, для хранения которого используется 8 байт, например, 1.2 или 34.76

complex - комплексные числа

Типы данных в Python

str - строки, например "hello". В Python 3.x строки представляют набор символов в кодировке Unicode

bytes - последовательность чисел в диапазоне 0-255

byte array - массив байтов, аналогичен bytes с тем отличием, что может изменяться

list - список

Типы данных в Python

tuple - кортеж

set - неупорядоченная коллекция уникальных объектов

frozen set - то же самое, что и set, только не может изменяться (immutable)

dict - словарь, где каждый элемент имеет ключ и значение

Строка - str

Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации, поэтому с помощью строк можно работать со всем, что может быть представлено в текстовой форме.

s = “It’s a string.”

Операции со строками

- Конкатенация (сложение)
- **>>>** S1 = 'spam‘
- **>>>** S2 = 'eggs‘
- **>>>** **print**(S1 + S2)
- 'spameggs‘
- Важно: в Python производится конкатенация строки только со строкой; при сложении строки с другим типом данных появится ошибка.

- Повторение строки
- `>>> print('spam' * 3)`
- spamspamspam
- Длина строки (функция len)
- `>>> len('spam')`
- 4
- Строку можно умножать только на целое число (int), иначе появляется ошибка.

- Доступ по индексу
- >>> S = 'spam‘
- >>> S[0]
- 's‘
- >>> S[2]
- 'a‘
- >>> S[-2]
- 'a'

- Извлечение среза
- >>> s = 'spameggs‘
- >>> s[3:5]
- 'me‘
- >>> s[2:-2]
- 'ameg‘
- >>> s[:6]
- 'spameg'
- >>> s[1:]
- 'pameggs‘
- >>> s[:]
- 'spameggs'

S1 + S2	Конкатенация (сложение строк)
S1 * 3	Повторение строки
S[i]	Обращение по индексу
S[i:j:step]	Извлечение среза
len(S)	Длина строки
S.find(str, [start],[end])	Поиск подстроки в строке. Возвращает номер первого вхождения или -1
S.rfind(str, [start],[end])	Поиск подстроки в строке. Возвращает номер последнего вхождения или -1
S.replace(шаблон, замена)	Замена шаблона
S.split(символ)	Разбиение строки по разделителю

Числа в Python

- Числа в Python 3 ничем не отличаются от обычных чисел. Они поддерживают набор самых обычных математических операций.
- int – целые числа: 1, 2, 3, 4, 5, …
- float – вещественные числа: 2.25, 3.14, …

Функции str() и int()

- str(input())
- int(input())

Операции с числами

x + y	Сложение
x - y	Вычитание
x * y	Умножение
x / y	Деление
x // y	Получение целой части от деления
x % y	Остаток от деления
-x	Смена знака числа
abs(x)	Модуль числа
x ** y	Возведение в степень

Операторы сравнения

Оператор	Описание	Примеры
==	Проверяет равны ли оба операнда. Если да, то условие становится истинным.	5 == 5 в результате будет True True == False в результате будет False "hello" == "hello" в результате будет True
!=	Проверяет равны ли оба операнда. Если нет, то условие становится истинным.	12 != 5 в результате будет True False != False в результате будет False "hi" != "Hi" в результате будет True
>	Проверяет больше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.	5 > 2 в результате будет True. True > False в результате будет True. "A" > "B" в результате будет False.
<	Проверяет меньше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.	3 < 5 в результате будет True. True < False в результате будет False. "A" < "B" в результате будет True.
>=	Проверяет больше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.	1 >= 1 в результате будет True. 23 >= 3.2 в результате будет True. "C" >= "D" в результате будет False.
<=	Проверяет меньше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.	4 <= 5 в результате будет True. 0 <= 0.0 в результате будет True. -0.001 <= -36 в результате будет False.

Сравнение строк

- При сравнении строк принимается во внимание символы и их регистр. Так, цифровой символ условно меньше, чем любой алфавитный символ. Алфавитный символ в верхнем регистре условно меньше, чем алфавитные символы в нижнем регистре.
- Вначале сравнение идет по первому символу. Если начальные символы обоих строк представляют цифры, то меньшей считается меньшая цифра, например, "1a" меньше, чем "2a".
- Если начальные символы представляют алфавитные символы в одном и том же регистре, то смотрят по алфавиту. Так, "aa" меньше, чем "ba", а "ba" меньше, чем "ca".
- Если первые символы одинаковые, в расчет берутся вторые символы при их наличии.
- Зависимость от регистра не всегда желательна, так как по сути мы имеем дело с одинаковыми строками. В этом случае перед сравнением мы можем привести обе строки к одному из регистров. Функция lower() приводит строку к нижнему регистру, а функция upper() - к верхнему.

Функция print()

```
print(*items, sep=' ', end='\n', file=sys.stdout, flush=False)
```

- Items – объект, который нужно вывести, * обозначает, что объектов может быть несколько;
- sep – разделяет объекты. Значение по умолчанию: ‘ ‘;
- end – ставится после всех объектов;
- file – ожидается объект с методом write (string). Если значение не задано, для вывода объектов используется файл sys.stdout;
- flush – если задано значение True, поток принудительно сбрасывается в файл. Значение по умолчанию: False.

Функция print()

```
print("Python — это весело.")

a = 5
print("a =", a)

b = a
print('a =', a, '= b')
```

Summary

- Оператор присваивания
- input()
- Типы данных в Python (float, int, str)
- Функции str() и int()
- Операции со строками (конкатенация, повторение строки, обращение по индексу, срез, длина строки, замена шаблона, разбиение по разделителю)
- Операции с числами (+, -, *, /, //, %, **)
- Операторы сравнения (==, !=, >, <, >=, <=)

Операторы if, elif, else

Синтаксис

Пример 1

```
if test1:
    state1
elif test2:
    state2
else:
    state3
```

Пример 2

```
if 1:
    print('true')
else:
    print('false')
```

Пример 3

```
a = int(input())
if a < -5:
    print('Low')
elif -5 <= a <= 5:
    print('Mid')
else:
    print('High')
```

Отступы важны!

Логический оператор and

X and Y

! Истина, если оба значения X и Y истинны
!

```
a = 5
b = 7
if a < 7 and b > 5:
    print('True')
else:
        print('False')
```

Логический оператор or

X or Y

! Истина, если хотя бы одно из значений X или Y истинно !

```
a = 5
b = 7
if a == 8 or b == 5:
    print('True')
else:
        print('False')
```

Логический оператор not

not X

! Истина, если X ложно !

```
a = 0
if a not 0:
    print('True')
else:
        print('False')
```

Задачи

1. Написать программу "Калькулятор", которая позволяет пользователю выбрать два числа и узнать результат любой операции над ними: сложение, вычитание, умножение, деление

2. Программа запрашивает у пользователя пароль и в случае совпадения с нужным выдаёт ему некоторую секретную информацию. В случае ошибки прощается с пользователем.

3. С помощью метода S.replace(шаблон, замена) заменить в строке S все двойные пробелы на одинарные. Вывести конечный вариант текста на экран.

4. Метод S.endswith(str) возвращает True, если строка S заканчивается шаблоном str. Написать программу, которая пытается угадать, является введенное пользователем русское имя женским или мужским.