# Лекция 1

## Установка

Для начала нам потребуется скачать питон с официального сайта. Важно (!) поставить галочку add python \_\_ to PATH. Затем нажать Install now. После чего запустить скаченный файл.

Можно работать с питоновскими файлами и без среды разработки (IDE), но это не совсем удобно, так как для запуска файла с расширение .py в терминале (это командная строка или же cmd) придется после перехода в директорию с файлом написать python \_\_\_.py, после чего в консоли появится результат выполнения программы.

Более легкий способ потребует для запуска нажать на одну кнопку. И вариант этот называется PyCharm. Скачать можно с сайта https://www.jetbrains.com/pycharm/download/ . Нужна Community версия, она полностью бесплатна и почти не имеет обрезанного функционала, так как программировать мы будем только на Питоне. А профессиональная версия включает в себя еще расширенные инструменты для разработки на HTML, JS и баз данных SQL.

## Арифметические операции

Рассмотрим простейшие арифметические операции и как они выглядят на Питоне:

Сложение + (+=): 3 + 2 = 5

Вычитание - (-=): 3 – 2 = 1

Умножение \* (\*=): 3 \* 2 = 6

Возведение в степень \*\*: 3 \*\* 2 = 9

Деление / (/=): 7 / 3 = 2.333333…

Целая часть от деления //: 7 // 3 = 2

Остаток от деления %: 7 % 3 = 1

Переменные и синтаксис

Python может определить тип переменной: для x = 1.35 будет выведен тип ***FLOAT***.

Что будет выведено, если

x = input()

y = input()

print(x+y)

при введении чисел 123 и 354. Должно быть 477? Нет, 123354. Всё это связанно с динамической типизацией и тем, что input выдает строковое значение. Чтобы преобразовать переменную одного типа в другой, мы можем использовать следующие функции:

str(х) - переводит переменную х в строковую

int(х) – переводит переменную х в целочисленную

float(х) – переводит переменную х в переменную с плавающей точкой

Тип bool – это логические переменные, то есть принимающие значение True (истина) или False (ложь).

Ко всем переменным можно просто обращаться по имени и получать их значение. Но строковые переменные – особенные, и к ним можно обращаться по индексу символа. Индексация в строке происходит с нуля. Вывести длину строковой переменной можно командой len(s). Пример обращения к строке по индексу:

S=’spam’

print(S[0]) >> s

print(S[2]) >> a

print(S[-2]) >>a

Строки можно умножать на числа, складывать между собой.

S1 = 'spam'

S2 = 'eggs'

print(S1 + S2) >>'spameggs'

print('spam' \* 3) >>spamspamspam

Строковые переменные можно доставать «срезами», делается это так:

название\_переменной[X:Y], где X – начало среза, а Y – окончание;

символ с номером Y в срез не входит. По умолчанию первый индекс равен 0, а второй - длине строки.

s = 'spameggs'

>>> s[3:5] >>'me'

>>> s[2:-2] >>'ameg'

>>> s[:6] >>'spameg'

>>> s[1:] >>'pameggs'

Функции для перевода всей строки в верхний регистр \_.upper(), а для перевода всей строки в нижний регистр \_.lower(), \_.title() это перевод начальных символов всех слов в верхний регистр, \_.capitalize() только первый символ заглавный.

Сбор строки из массива осуществляется так

переменная = “способ связи”.join (название массива) или вместо массива можно указать название строковой переменной для разделения между каждым символом указанным способом связи.

## Условные операторы

Синтаксис условных операторов if elif else:

if условия 1:

действие 1

elif условие 2:

действие 2

else:

действие 3

Блок elif является необязательным и конструкцию if elif else можно построить обычным образом если иначе – if else. Elif может быть сколько угодно.

Укороченная версия if-else:

if X:

A=Y

else:

A=Z

Можно также написать A=Y if X else Z. В литературе этот прием называется трехместное выражение if-else.

## Логические операции

Сравнение возвращает логическое значение.

Рассмотрим основные логические операторы: X and Y – логическое И (выдает истину только когда Х И Y истинны), X or Y – логическое ИЛИ (выдает истину, если хотя бы 1 из двух истинно), X xor Y – взаимоисключающее ИЛИ (является истиной, если только одно из двух истина), not X – логическое отрицание (инверсия).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| А | В | not A | A and B | A or B | A xor B |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 |