Лекция 13. Введение в программирование на языке Python.

Python представляет собой популярный высокоуровневый язык программирования, который предназначен для создания приложений различных типов. Это и веб-приложения, и игры, и настольные программы, и работа с базами данных. Довольно большое распространение питон получил в области машинного обучения и исследований искусственного интеллекта.

Выполнение программы на Python выглядит следующим образом. Сначала мы пишем в текстовом редакторе скрипт с набором выражений на данном языке программирования. Передаем этот скрипт на выполнение интерпретатору. Интерпретатор транслирует код в промежуточный байткод, а затем виртуальная машина переводит полученный байткод в набор инструкций, которые выполняются операционной системой*.*

Интерпретатор Python сопроваждается достаточным функционалом, который позволяет создавать приложения на этом языке.

**Для создания графических приложений:** Tkinter, PyQt / PySide, wxPython, DearPyGui, EasyGUI

**Для создания мобильных приложений:** Kivy, Toga.

**Для создания веб-приложений:** Django, Flask, FastAPI, Pylons, Bottle, CherryPy ,TurboGears, Nagare

**Для автоматизации процессов:** Selenium (для тестирования веб-приложений), Flask, FastAPI, Pylons, Bottle, CherryPy, TurboGears, Nagare, robotframework, pywinauto, Lettuce, Behave, Requests

**Для работы с различными типами файлов:** OpenPyXL (Excel), lxml (XML), ReportLab / borb (PDF), pdfrw / PyPDF2 (PDF), Pandas (CSV и Excel)

**Для машинного обучения, искусственного интеллекта, Data Science:** Pandas, SciPy, PyTorch, Matplotlib, Theano, Tensorflow, OpenCV, Scikit-Learn, Keras, NumPy

**Для визуализации:** Matplotlib, Seaborn, Plotly, Bokeh, Altair, HoloViews

Введение в написание программ

Программа на языке Python состоит из набора инструкций. Каждая инструкция помещается на новую строку. Большую роль в Python играют отступы. Неправильно поставленный отступ фактически является ошибкой. Однако стоит учитывать, что некоторые конструкции языка могут состоять из нескольких строк.

Python - регистрозависимый язык, поэтому выражения print и Print или PRINT представляют разные выражения. И если вместо метода print для вывода на консоль мы попробуем использовать метод Print

Функция print отвечает за вывод, а функция input отвечает за ввод информации.

Лекция 14. Базовые структуры данных.

1. Переменные

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

И также надо учитывать регистрозависимость, поэтому переменные name и Name будут представлять разные объекты

Два типа наименования переменных:

* **Camel case** подразумевает, что каждое новое подслово в наименовании переменной начинается с большой буквы.
* **Underscore notation** подразумевает, что подслова в наименовании переменной разделяются знаком подчеркивания

Типы данных:

* bool
* int, при использовании двоичной системы обозначается 0b.

Пример: 0b11. Восьмеричная система обозначается 0o. Пример: 0o7. Шестнадцатеричная система обозначается 0x. Пример: 0x0A.

Стоит отметить, что в какой-бы системе мы не передали число в функцию print для вывода на консоль, оно по умолчанию будет выводиться в десятичной системе

* float. Может иметь только 18 значимых символов
* complex (пример: 1+2j)
* str. (’’’ hello ’’’) это и многострочный комментарий и строка в нескольких стоках.

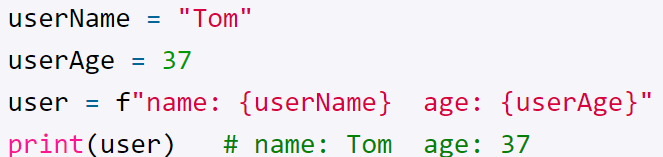
Специальных символов - управляющих последовательностей:

* \\: позволяет добавить внутрь строки слеш
* \': позволяет добавить внутрь строки одинарную кавычку
* \": позволяет добавить внутрь строки двойную кавычку
* \n: осуществляет переход на новую строку
* \t: добавляет табуляцию (4 отступа)

Путь к файлу указывается в виде строки, но требуется поставить **r** перед строкой.

Пример: 

Python позволяет встравивать в строку значения других переменных. Для того что бы вставлять переменные в строки, их следует размещаются в фигурных скобках {} внутри строки, а перед всей строкой ставится символ f:

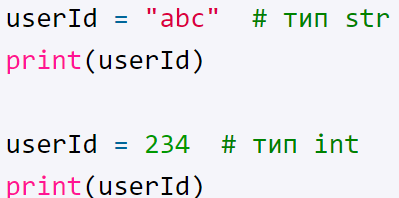
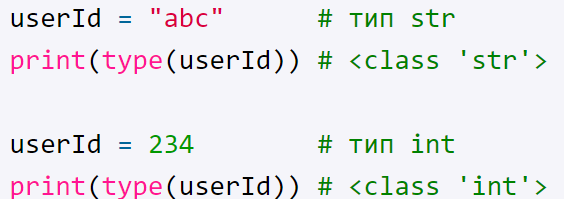


**Динамическая типизация**

Python является языком с динамической типизацией. А это значит, что переменная не привязана жестко к определенному типу.

При этом в процессе работы программы мы можем изменить тип переменной, присвоив ей значение другого типа:

С помощью встроенной функции **type**() динамически можно узнать текущий тип переменной:

**Неявные преобразования**

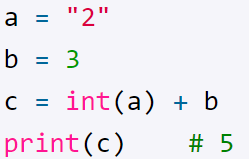
В арифметических операциях оба числа должны представлять один и тот же тип. Если же два операнда операции представляют разные типы данных, то Python пытается автоматически выполнить преобразования к одному из типов. Существуют следующие правила.

* Если один из операндов операции представляет комплексное число, то и второй операнд, то же приводится к complex.
* Если один из операндов операции представляет вещественное число, то и второй операнд, то же приводится к float.
* Иначе, оба операнда должны представлять тип int, и в этом случае преобазование не требуется

**Явные преобразования**

Для преобразования типов Python предоставляет ряд встроенных функций:

* int(): преобразует значение в целое число
* float(): преобразует значение в число с плавающей точкой
* str(): преобразует значение в строку

Пример: 

Если число складывается с число, то это стандартная операция сложения чисел. Если строка складывается со строкой, то это операция объединения строк.