Методические указания для тренера  
по курсу «Основы алгоритмизации и программирования для разработки ботов для телеграмм»

# Часть 1. Элементарное программирование

Цель: Знакомство с Python и его базовым синтаксисом

## Базовый синтаксис языка Python

* Расскажите слушателям историю языка Python (слайд 4)
* Покажите слушателям пример «hello world», объясните структуру программы
* Расскажите слушателям, что программа состоит из команд, которые работают с данными. Обратите их внимание, что программа работает с переменными, через которые мы получаем доступ к разным структурам данных, например к набору символов, называемых строками. К данным мы получаем доступ через переменные. Объясните слушателям требования к именам переменных в Python
* Покажите слушателям более сложный пример – решение квадратного уравнения. Объясните необходимость оператора условного перехода. Сделайте ссылку, что он подробно будет обсуждаться далее
* Попросите слушателей запрограммировать расчет физических формул, дополнив файл Module1\Exercises\physics.py своими решениями и проверьте их
* Объясните слушателям важность отступов на примере задачи «Квадратное уравнение». Обратите внимание, что они должны использовать или только пробелы или только табуляции.

## Среды разработки Python

* Расскажите слушателям об интерпретаторе командной строки Python. Обратите внимание слушателей, что интерпретатор удобен, когда вы изучаете какую-то библиотеку python, когда пишите несложный скрипт автоматизации, но программы лучше разрабатывать в специализированных инструментальных средах
* Покажите слушателям ключи командной строки интерпретатора Python
* Расскажите слушателям о инструментальной среде PyCharm. Обратите внимание на использование инструмента для полноценной разработки приложений. Если вы работаете в этой среде, продемонстрируйте ее слушателям, покажите основные действия, связанные
* Расскажите слушателям об инструменте Visual Studio Code. Продемонстрируйте, как открыть каталог, создать файлы и запустить приложение на Python. Эта среда является рекомендованная в этом тренинге
* Расскажите слушателям о рекомендованных расширения к Visual Studio Code:
  + Python – основное расширение для запуска Python-программ
  + Python Preview – расширение, позволяющее разобраться как устроена внутренняя память Python-программ
  + Python Indent – расширение, которое автоматически корректирует необходимые отступы
  + Python Coding Conventions – расширение, позволяющее с помощью горячих клавиш отформатировать код программы, так чтобы он соответствовал общепринятым правилам в Python

## 1.3. Элементарные структуры данных

* В этом модуле мы изучим основные типы данных в языке Python и операции над этими типами данных
* Расскажите слушателям о том, что такое тип данных. Обратите их внимание, что операции с типами данных не обязательно обозначаются традиционно знаками +, - и т.д., но и могут быть обозначены вполне понятными именами, например append, count и т.д.
* Обратите внимание слушателей, что тип переменной определяется автоматически на основе типа выражения, стоящего в правой части операции присваивания
* Расскажите о численных типах данных:
  + int – целые числа
  + float - числа с плавающей точкой
  + complex – комплексные числа
* Обратите внимание слушателей, что численные типы данных поддерживают большую арифметику
* Продемонстрируйте слушателям операции с числами
* Расскажите слушателям о битовых операциях. Обратите их внимание на то, что данные операции используются в алгоритмах шифрования, работы с внешними устройствами. Продемонстрируйте эти операции. Если аудитория не знает данных операций, пожалуйста, расскажите о них.
* Расскажите слушателям о комплексных числах. Покажите пример программы и опишите как можно создать комплексное число и как можно получить доступ к его элементам
* Объясните слушателям, что такое строка. Обратите их внимание на то, что строка – неизменяемый тип данных, а также, что по умолчанию строка хранит символы в формате Unicode
* Расскажите слушателям о байтах и байтовых массивах. Обратите внимание, что байтовый массив, в отличие от строк можно менять, но указывать нужно код символа в виде целого числа
* Продемонстрируйте типовые операции со строками. Обратите внимание слушателей на разницу между операциями index() и find(), а также rindex() и rfind()
* Попросите слушателей выполнить упражнение «Элементарные операции со строками»
* Расскажите слушателям про оператор slice. Объясните как работает оператор slice, если параметры оператора отрицательные, используя слайд
* Обсудите со слушателями форматированный вывод. Подробно объясните программу, приведенную на слайде
* Расскажите слушателям про списки и кортежи. Обратите их внимание, на необходимость использования списков только тогда, когда нам необходимо менять их содержимое. Если же нам содержимое контейнеров необходимо только для чтения, тогда следует использовать кортежи – они занимают меньше памяти и более производительны
* Продемонстрируете слушателям основные операции над списками
* Обратите внимание слушателей, что кортежи поддерживают такие же операции как и в списках, кроме изменяющих содержимое кортежа. В любом случае, операции сложения списков со списками и кортежа с кортежами создают новый список или кортеж соответственно
* Объясните слушателям как будет работать код, в котором к концу списка добавляют в качестве элемента этот же список
* Расскажите слушателям про операцию присваивания значений кортежа или списка переменным, находящимся слева от знака «=»
* Попросите слушателей выполнить упражнение на операции со списками и кортежами. Обратите их внимание, на то что часть операций, которые удалось выполнить со списком не будут доступны для кортежей. Обсудите почему
* Расскажите слушателям о словарях. Продемонстрируйте работу со словарем
* Расскажите слушателям о множестве. Обратите их внимание, что множество не содержит дубликатов
* Расскажите слушателям об операциях над множествами. Если будет необходимость, напомните им основные идеи из теории множеств (<https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2#%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%8B%D0%B5_%D0%BF%D0%BE%D0%BD%D1%8F%D1%82%D0%B8%D1%8F>)
* Продемонстрируйте операции над множествами слушателям
* Попросите слушателей выполнить элементарные операции над множествами
* Расскажите слушателям о возможности преобразования типа данных. Продемонстрируйте возможности преобразования типа данных

## Операторы Python

* Расскажите слушателям о необходимости операторов логики ветвления
* Объясните слушателям логику работу оператора if используя схему его алгоритма на слайде
* Объясните слушателям, что оператор if и далее оператор while в качестве условия рассматривают значение True или False. Зная как трактуется то или иное значение выражения можно упросить запись. Продемонстрируйте слушателям примеры применения оператора if (Module3\Live\if.py)
* Расскажите слушателям про условное выражение. Это выражение используется в ситуации, когда в зависимости от условия одной переменной надо присвоить или одно или другое значение. В таких простых ситуациях оператор if будет выглядеть очень громоздко
* Расскажите слушателям для чего применяется оператор цикла while. В следующем разделе тренинга он будет использоваться в реализации алгоритмов. Объясните работу этого оператора с использованием схемы на слайде
* Напомните слушателям про структуры данных. Структуры данных хранят другие данные в виде последовательности, называемой контейнером. К таким последовательностям в Python обычно относят списки и кортежи
* Расскажите слушателям как работает оператор for, используя схему на слайде
* Расскажите слушателям об операторе break. Этот оператор используется для досрочного завершения цикла без участия условия его продолжения
* Расскажите слушателям об операторе continue. Этот оператор используется для досрочного завершения блока цикла и перехода к следующей итерации

## Функции

* Дайте слушателям определение функции, расскажите о задачах, которые они решают
* Расскажите о синтаксисе определения функции. Обратите внимание слушателей, что в в одном файле Python может быть только одна функция с одним и тем же именем. Если в файле несколько функций с одинаковым именем, то кроме последней определенной, все остальные одноименные функции игнорируются
* Объясните слайд, в котором показан пример использования функций. Обратите внимание слушателей на механизм передачи значений параметров, которые получают начальное значение в момент вызова функции
* Расскажите слушателям про лямбда функцию. Не следует давать много теории по функциональному программированию, можно лишь ограничиться фразой, что лямбда-функция – это небольшая функция, которая используется для указания критерия сравнения в некоторых функциях Python.
* Объясните слушателям пример с функцией filter на слайде. Обратите их внимание, что благодаря лямбда-функции, функция filter получается универсальной и не зависит от способа фильтрации, который как раз и определяется этой лямбда функцией.
* Расскажите слушателям о типовом алгоритме обработки данных «Map-Filter-Reduce» (<https://codecamp.ru/blog/map-filter-reduce/>). Здесь участвует три функции – map(), filter() и reduce(). Функция map преобразует список в некоторый другой список. Функция filter отбирает список, согласно критерию, а reduce() выполняет некоторую операцию над всеми элементами списка, в результате чего получается некоторое значение. Здесь лямбда-функция играет роль функции трансформации элемента списка, функции, определяющий критерий отбора и функции, выполняющей накапливающую операцию. Такой подход часто используется при естественной обработке текста.
* Объясните слушателям пример реализации алгоритма «Map-Filter-Reduce». В этом примере, исходный список содержит кортежи из двух элементов – строки и числа. С помощью функции map() мы создаем список строк, содержащий ровно то количество символов, которое было задано во втором элементе кортежей исходного списка. Далее, с помощью функции filter() мы оставляем в списке только те строки, длина которых больше трех. И наконец, функция reduce суммирует длины строк.