**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования  
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ имени Н.Э.БАУМАНА  
(национальный исследовательский университет)»**

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

**Конспект по языку Python**

**(дисциплина «Основы информатики»)**

Работу выполнил

студент группы ИУ9-11Б

Окутин Денис

Москва, 2021

**1) Типизация и система типов языка.**

Python — это язык с **сильной динамической типизацией**.

В Python типы данных делятся на встроенные в интерпретатор и не встроенные, которые можно использовать при импортировании модулей. (ниже представлена классификация встроенных типов)

**Типы данных Python:**

1. Числа *(int, float,complex)*
2. Cтроки
3. Bool *(true/false)*
4. None
5. Cписки *(list – список, turple – кортеж (последовательность неизменяемых объектов), range – диапазон)*
6. Множества *(set – множество, frozenset – неизменяемое множество)*
7. Словари *(== связанные списки в Scheme)*

Типы данных в Python делятся на **изменяемые/неизменяемые**.

Изменяемые: числа *(int, float,complex)*, логические *(bool),* кортежи *(turple),* строки *(str),* неизменяемые множества*(frozen set)*

Неизменяемые: списки *(list),* множества *(set),* словари *(dict)*

**Тип переменной** можно определить с помощью функции *type():*



**2) Основные управляющие конструкции.**

**Оператор if-elif-else**

Синтаксис:

****

Части в скобках опциональны, часть, помеченная \* может повторяться.

Принцип работы таков, что если <условие1> истина, то выполнится <оператор1>, остальные ветки игнорируются. Если <условие1> ложно, то проверяется <условие2> по аналогии, если все условия оказались ложными, то выполняется ветка else.

**Оператор for**

Синтаксис:



Значение <переменная> пробегает все значения из элемента <список>, и пока это выполнено выполняется <оператор1>, который находится в теле цикла. Когда цикл завершается, выполняется код, расположенный ниже цикла.

**Оператор while**

Синтаксис:



Пока <условие> истина будет выполняться <оператор1>, иначе цикл завершиться и будет выполняться участок кода, расположенный ниже.

**Операторы break и continue**

Break осуществляет немедленный выход из циклов for и while.

Continue осуществляет немедленный переход на следующую итерацию цикла

**Оператор def**

Синтаксис



Может вызываться в любой части кода, при вызове выполняется тело функции, вызов осуществляется строго по шаблону, заданному в начале функции

**Операторы return и yield**

Синтаксис

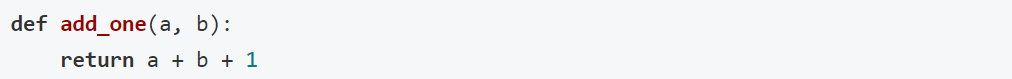
****

Осуществляют возврат из функции, возвращая значение <результат>. Отличие в том, что при повторном вызове функции a) в случае return функция начинает работать с самого начала b) в случае yield функция продолжается с момента возврата из функции.

**3)Подмножество языка для функционального программирования**

Функции в Python являются объектами первого класса (могут употребляться в программе наравне с объектами других типов данных).

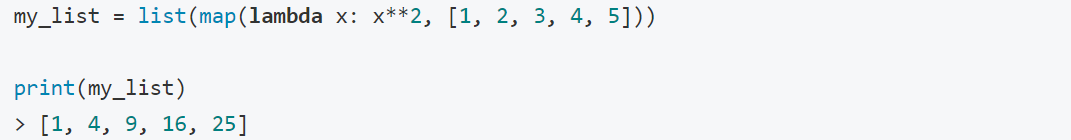
Функция может быть определена как оператором *def* так и *lambda* выражением

**Встроенные функции высших порядков**

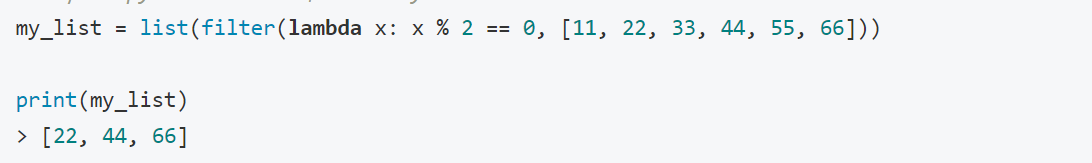
**1) map()**

Принимает функцию-аргумент и применяет её ко всем элементам входящей последовательности.

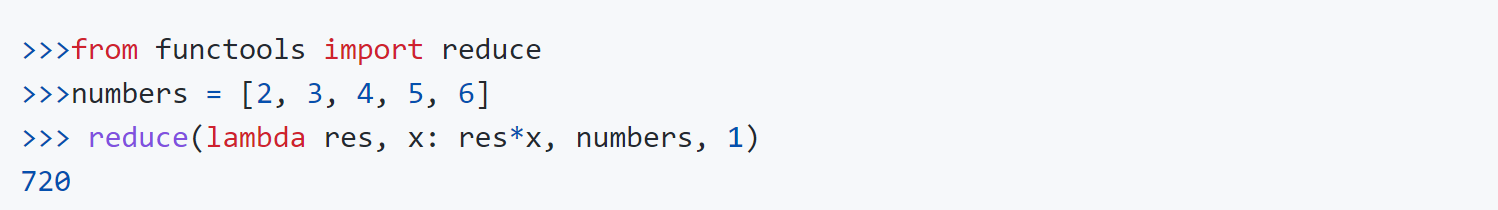
****

**2) filter()**

Фильтрует последовательность по заданному условию

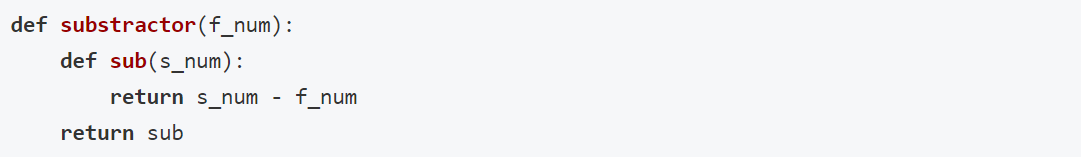


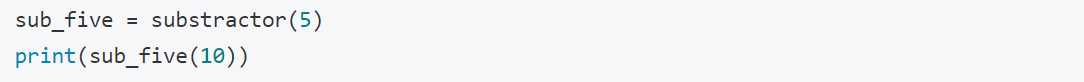
**3) reduce()**



**Замыкания**

Замыкания неразрывно связаны с концепцией вложенных функций. Это такие функции, в теле которых есть ссылки на переменные, объявленные вне определения этой функции и не являющиеся её параметрами.







**4) Важнейшие функции для работы с потоками ввода/вывода, строками, регулярными выражениями**

**Ввод/вывод**

**Стандартные:**

input() возвращает считанное значение ввиде строки

print() выводит строку в stdout

**Импортируемые из модуля sys:**

*sys.stdin*возвращает считанное значение из консоли

*sys.stdout*отображает результаты команды в консоли

*sys.stderr*отображает ошибки возникшие при работе программы в специальный поток вывода ошибок

**Работа со строками**

Конкатенация (сложение)

****

Дублирование строки

****

Нахождение длины

****

Доступ по индексу (отрицательные индексы идут с конца строки)

****

Извлечение среза

Оператор извлечения среза: [X:Y:a], где X- начало среза, Y – конец среза, a- шаг с которым берутся элементы среза (по умолчанию a=1)

****

****

**Регулярные выражения**

В Python доступны при импортировании модуля *re*

**

Наиболее часто используемые методы модуля:

