Введение в программирование на языке Python.

Python - интерпретируемый скриптовый язык программирования.

Мы используем пакет Anaconda - дистрибутив Python с большим числом установленных библиотек и удобной средой разработки.

Установка анаконды на свой компьютер/ноутбук см. https://docs.continuum.io/anaconda/

Преимущества Python:

1) Сравнительно простой язык
2) Дает возможность быстро писать небольшие программы
3) Быстро развивается, появляются новые библиотеки
4) Python-сообщество

Недостатки Python:

1) Низкая скорость выполнения кода по сравнению с компилируемыми языками статической типизации (C/C++, Java)

Типы данных в Python (множества с определенными на них операциями)

Немного об ООП (объекто-ориентированное программирование).

1) Класс - описание, каким должен быть объект
2) Объект - конкретный экземпляр класса

В Python все данные являются классами.
Встроенные типы данных:

1. **Логический**, может принимать одно из двух значений — True или False.
2. **Числа**, могут быть целыми (int), с плавающей точкой (float), комплексными.
3. **Строки** — последовательности символов Юникода, например, HTML-документ (str).
4. **Списки** — упорядоченные последовательности значений (list).
5. **Кортежи** — упорядоченные неизменяемые последовательности значений (tuple).
6. **Множества** — неупорядоченные наборы значений (set).
7. **Словари** — неупорядоченные наборы пар вида ключ-значение (dict).
8. **Байты** и **массивы байтов**, например, файл изображения в формате JPEG.

Мы можем сами определять классы и создавать объекты классов.

Операции над встроенными типами данных:

1) операция присвоения (=)
2) логические операции (and, or, not)
3) операции сравнения (<,>, >=, <=, ==, !=, is, is not)
4) арифметические операции (+,-, *, /, //, %, ** и тд)

5) битовые операции (&,|, ^, ~, <<, >>) - применяются для целых чисел (int)

Ветвления и циклы:

Условные конструкции (if-elif-else):

if *условие1*:
    блок1
elif *условие2*:
    блок2
else:
    блок3

Обработка исключений:

try:
    блок 1   # интерпретатор пытается выполнить блок1
except (name1,name2):
    блок 2   # выполняется, если в блоке try возникло исключение name1 или name2
except name3:
    блок 3   # выполняется, если в блоке try возникло исключение name3
except:
    блок 4   # выполняется для всех остальных возникших исключений
else:
    блок 5   # выполняется, если в блоке try не возникло исключения
finally:
    блок 6   # выполнится всегда

Цикл while

while условие:
    блок1
else: # необязательно
    блок2   # выполняется, если выход из цикла был произведён не инструкцией break

ключевые слова:
break - выход из цикла
continue - переход на следующую итерацию цикла

Цикл for:
for <элемент> in <итерируемый объект>:
    блок1
else: # необязательно
    блок2   # выполнится, если выход из цикла не осуществлялся инструкцией break

Литература:
https://wiki.python.org/moin/BeginnersGuide/NonProgrammers