Python

базовый тренинг

Рузин Алексей
ruzin@me.com

Цель

- Познакомиться с языком
- Научиться самостоятельно разрабатывать приложения

План

- Почему Питон так хорош?
- Среда разработки PyCharm
- Переменные и арифметические выражения
- Условные операторы
- Операции ввода-вывода с файлами
- Строки
- Списки, Кортежи, Множества, Словари
- Итерации и циклы
- Функции
- Объекты и классы
- Исключения
- Модули
- Практическая задача

Дзен

- Красивое лучше уродливого
- Явное лучше неявного
- Простое лучше сложного
- … (import this)

- Style Guide: PEP8

```
file = open('file.csv', 'r')
for line in file:
    print line
file.close()
```

Достоинства Python

- простой и понятный
- выразительный
- поддержка комплексных чисел
- интерпретируемый (+/-)
- динамическая типизация (+/-)
- сборщик мусора (+/-)
- большое количество библиотек
- GIL - ограничение на многопоточность (недостаток)

Где используется

- Научные исследования
- Системное администрирование
- Веб-разработка
- Google, Yandex и пр.

Hello, World

- Создание проекта в PyCharm
- Вызов консоли
- Hello, world
- Перезапуск консоли

Переменные

- Переменные
- Типы данных: целые, вещественные, комплексные, булевы, строки, объекты, None
- type(var), isinstance(var, type)

- type(10)
- type(type(10))
- type(type(type(10)))

Арифметика

- +, -, *, /, **

  2 * 3 + 3
  3 + 3 * 2

- изменение приоритетов с помощью ()

Строки

- a=‘string’ (или "string", """string""")
- Получить символ в n-й позиции: a[n]
- подстрока: a[10:15]
- конкатенация (+)
- format()

Условные операторы

- if a < b:
    print 'a < b'
  elif a > b:
    print 'a > b'
  else:
    print 'a == b'

- if my_var is None or my_var is not None:
    print True

- if 'key' in ‘hockey':
     print ‘key is in hockey’

Цикл while

- a = 0
  while a < 100:
    a += 1

- break, continue

Списки

- a = [1,2,4,’asdf']
- обращение к элементу по индексу a[x]
- a.append(x), a.insert(2, 'Thomas')
- срезы списков a[a:n]
- конкатенация списков (+)
- "двумерный массив" == "список списков"

Итерация по списку

- a = [1, 2, 4, 8]
  for x in a:
      print(x)

- генератор списков [x*x for x in a]

- for x in range(10):
      pass

Tuple (кортеж)

- a = (1, 2, 3)
- обращение к элементу: a[x]
- в отличие от списка не изменяемые
- for i in (1, 2, 3):
  print(i)
- x, y, z = a

Множества

- a = {1, 2, 3}
- Проверка нахождения элемента: x in a
- a.add(x), a.remove(x), a.update({2,3})
- |, &, -, ^
- for i in a:
    print(i)

```
a = {1,2,3}
b = {2,3,4}

print a | b

print a & b

print a - b

print a ^ b
```

Словарь

- a = {‘key_1': 1, 2: ‘value_2’}
- что может быть ключом
- проверка нахождения ключа: ‘key_1’ in a
- for … in
  a.keys(),
  a.values(),
  a.items()

Функции

- определение:
  def func(a):
      return a + 1

- вызов:
  func(3)

- параметры по-умолчанию
  def func(a=3, b=4):
      return a+b

- вызов с параметром по-умолчанию:
  func(b=88)

Классы

- ООП: Инкапсуляция, Наследование, Полиморфизм

- определение:
  class MyClass:
      field1 = 15
      def method(self, a):
          print(a*self.field1)

- создание экземпляра
  a = MyClass()

- обращение к полям и методам:
  a.field1
  a.method(20)

Классы

- Конструктор
```
class MyClass:
    def __init__(self, field1):
        self.field1 = field1
```

- Наследование
```
class MyClass(BaseClass):
    def __init__(self):
        super().__init__()
```

Исключения

- try:
  …
  except Exception as e:
  print(e)
  else:
  print(‘no exception’)
  finally:
  print(‘finally’)

- raise Exception(‘some message’)

Модули и пакеты

- .py файл - модуль
- **import** my_module
- **from** my_module **import** something
- каталог с файлом __init__.py - пакет

Ввод/вывод файлов

```
file = open('file.csv', 'r')
for line in file:
    print line
file.close()

with open('file.txt', 'w') as f:
    a = 0
    while a < 5:
        a += 1
        print(‘this is line #{}’.format(a), file=f)
```

Задача

Дано:
- файл со списком пользователей в формате CSV:
ID, Имя, Возраст
- файл со списком просмотренных страниц (товаров) в формате CSV:
URL, User_ID, длительность просмотра

Составить программу, которая построит распределение пользователей по возрасту для заданного URL в виде:
0-9 лет: 3
10-19 лет: 17
…
90-99 лет: 1

Литература

- Марк Лутц: Изучаем Python
- Дэвид Бизли: Python подробный справочник
- google
- stack overflow
- docs.python.org