Лекция 12

Язык программирования Python.

Хайрулин Сергей Сергеевич

email: s.khairulin@g.nsu.ru, s.khayrulin@gmail.com

Ссылка на материалы

План

- Лекции/практические занятия
  - Тест
- Дифференцированный зачет в конце семестра
  - Защита задания

Литература

Начальный уровень

- Mark Pilgrim. Dive into Python - http://www.diveintopython.net/
- Марк Лутц. Изучаем Python, 4-е издание // Символ-Плюс 2011.
- ...

Стандарт/Документация

- PEP-8 - https://www.python.org/dev/peps/pep-0008/
- https://www.python.org/
- https://github.com/python/cpython

Экспертный уровень

- Лучано Рамальо: Python. К вершинам мастерства
- Mitchell L. Model. Bioinformatics Programming Using Python // O'Reilly 2010.

Версии Python

- Python 2 вышел 2010 году последняя версия 2.7.16 - исправлялись только баги(ошибки) с января 2020 года поддержка прекращена.
- Python 3 в появился в 2008, является актуальной версией языка. Текущая стабильная версия 3.9, в разработке 3.10
  - Python 3 не гарантирует совместимости кода с Python 2

Summary

- Форматирование строк.
- Магические методы.
- Функции первого класса. Элементы функционального программирования
  - lambda функции.
  - Замыкания.
- Освобождение ресурсов - garbage collector.

Форматирование строк.

```
In [2]: var = "world"

In [3]: "Hello %s"%(var)
Out[3]: 'Hello world'

In [4]: var2 = "It's me"

In [5]: "Hello %s %s"%(var, var2)
Out[5]: "Hello world It's me"
```

Форматирование строк (format).

```
In [9]: "Hello {0} It's {1}".format("world", "me")
Out[9]: "Hello world It's me"
```

```
In [10]: "Hello {k1} It's {k2}".format(k1="world", k2="me")
Out[10]: "Hello world It's me"
```

Форматирование строк (f string).

fstring появились с релизом python 3.4 - позволяют форматировать строки на лету подставляя текущее значение переменной.

```
In [13]: who = "me"

In [14]: f"Hello {var} It's {who}"
Out[14]: "Hello world It's me"
```

!Переменные должны быть определены на момент формирования строки.
Иначе это приведет к ошибкам.

Магические методы.

Магические методы - это подход в python к *перегрузке операторов*, позволяющий классам определять свое поведение в *отношении операторов языка*. Подобные методы добавляются в реализацию класса и должны называться определенным образом

def __<meth_name>__(args....):

Со всем списков методов и описанием можно ознакомиться в этой статье.

Магические методы.

```
In [5]: class A:
   ...:     def __init__(self, data):
   ...:         self.__acum = data
   ...:     def __str__(self):
   ...:         return str(self.__acum)
   ...:     def __add__(self, other):
   ...:         self.__acum += other
   ...:

In [6]: a1 = A(10)

In [7]: a1 + 5

In [8]: print(a1)
15
```

Магические методы.

matrix.py > Matrix > __mul__

```python
class Matrix:
    def __init__(self, data):
        """
        :param data: list[list[float]]
        """

        self.matrix = data

    def __str__(self):
        result = ""
        for row in self.matrix:
            tmp = ""
            for item in row:
                tmp += str(item) + " "
            result += tmp + "\n"
        return result

    def mult_scalar(self, scalar):
        for i in range(len(self.matrix)):
            for j in range(len(self.matrix[i])):
                self.matrix[i][j] *= scalar

    def __mul__(self, m):
        if isinstance(m, (float, int)):
            for i in range(len(self.matrix)):
                for j in range(len(self.matrix[i])):
                    self.matrix[i][j] *= m
            return self

        elif isinstance(m, Matrix):
            raise NotImplementedError("It's en exircise!")
```

Магические методы.

```
In [10]: class A:
    ...:     def __init__(self, data):
    ...:         self.__acum = data
    ...:     def __str__(self):
    ...:         return str(self.__acum)
    ...:     def __int__(self):
    ...:         return self.__acum
    ...:

In [11]: a = A(10)

In [12]: print(a)
10

In [13]: int(a)
Out[13]: 10
```

Функции первого класса.

В информатике язык программирования имеет функции первого класса, если он рассматривает функции как объекты первого класса. В частности, это означает, что язык поддерживает передачу функций в качестве аргументов другим функциям, возврат их как результат других функций, присваивание их переменным или сохранение в структурах данных.

Функции первого класса.

```
In [29]: def foo():
    ...:     print("Function Foo is running")
    ...:

In [30]: bar = foo

In [31]: bar()
Function Foo is running
```

```
In [33]: def func(f):
    ...:     f()
    ...:

In [34]: func(bar)
Function Foo is running
```

Lambda - функции

Lambda function - однострочные анонимные функции

```
In [39]: (lambda x: x+1)(2)
Out[39]: 3

In [40]: foo = lambda x: x+1

In [41]: foo(2)
Out[41]: 3
```

Замыкание

```
In [50]: def foo():
    ...:     x = 1
    ...:     def inner_f():
    ...:         nonlocal x
    ...:         x += 1
    ...:         return x
    ...:     return inner_f
    ...:

In [51]: a = foo()

In [52]: a()
Out[52]: 2

In [53]: a()
Out[53]: 3
```

Освобождение ресурсов - garbage collector.

В python реализован алгоритм сборки мусора, который удаляет объекты(освобождает память занятую этими объектами) из памяти. Такими образом вам не нужно заботиться об утечках памяти, до тех пор **пока вы сами ее не сделаете.**

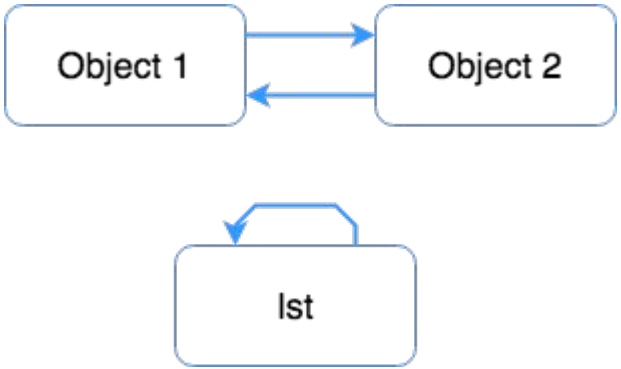
Об алгоритме можно почитать здесь

Освобождение ресурсов - garbage collector.

На самом деле алгоритма 2

1. Подсчет ссылок

2. Сканирование на наличее циклических ссылок



Спасибо за внимание!