

Python 3

Краткое введение, часть 2

Хамовники, 2011

План лекции

1. Объявление своих функций
2. Генераторы списков (list comprehension)
3. Файловый ввод-вывод

Объявление своей функции

```
def myRange(start, stop, step = 1):  
    result = []  
    current = start  
    while current < stop:  
        result.append(current)  
        current += step  
    return result
```

```
myRange(0, 10)
```

```
myRange(0, 10, 2)
```

```
myRange(stop=10, start=0)
```

Особенности создания функций

1. Переменные со значением по-умолчанию задаются только после переменных без такого значения.
2. При вызове функции именованные параметры передаются только после неименованных.
3. Сразу после выполнения строки `return` функция завершает свою работу.
4. Ваша функция может принимать на вход и возвращать переменные любого типа. Будьте аккуратны и старайтесь контролировать то, что вы передаете в функцию.

Значение None

None - специальное значение, означающее отсутствие у переменной какого-либо осмысленного значения.

Любая функция в python возвращает некоторое значение. Если в функции нет оператора return, то функция вернёт значение None. То же произойдёт, если использовать return без параметра. Пример:

```
a = print(10)  
print(a)
```

Вывод:

10

None

Генераторы списков

Если вам надо быстро сделать список определенного вида - вам помогут генераторы списков.

```
[x for i in range(10)] == [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
[x * x for i in range(2, 5)] == [4, 9, 16]
```

```
[x for i in range(1, 10, 2)] == [1, 3, 5, 7, 9]
```

```
[x for i in range(1, 5) if x % 2 == 0] == [2, 4]
```

Задачи:

1. `[[i + j for i in range(5)] for j in range(5)]`
2. Как сделать двумерный массив n на m клеток, заполненный нулями?

Файловый ввод-вывод.

Открытие файлов.

```
input_file = open("input.txt", "r")
```

- открыть файл ("input.txt" - название файла, "r" - режим, в котором файл будет использоваться (чтение))

```
output_file = open("output.txt", "w")
```

- открыть файл для записи

```
input_file.close()
```

```
output_file.close()
```

- закрытие открытых файлов

Файловый ввод-вывод.

Чтение из файла

```
input_file.read()
```

- считывает весь файл и записывает его в одну строку (в строку могут входить символы перевода строки).

```
input_file.readline()
```

- считывает одну строку из файла (до перевода строки).

```
input_file.readlines()
```

- считывает все строки из файла и записывает их в список строк (один элемент списка - одна строка).

Файловый ввод-вывод.

Вывод в файл.

```
output_file.write(s)
```

- записывает строку `s` в выходной файл (только строку).

```
print(file=output_file, s, t, 123)
```

- работает, как обычный `print`, но выводит в файл.

```
print(file=output_file, s, t, 123, sep=',', end='\n\n')
```

- выведет `s`, `t` и `123` через запятую, после всего этого выведет два перевода строки (по умолчанию `sep=' ', end='\n'`).