

# Функциональное программирование

---

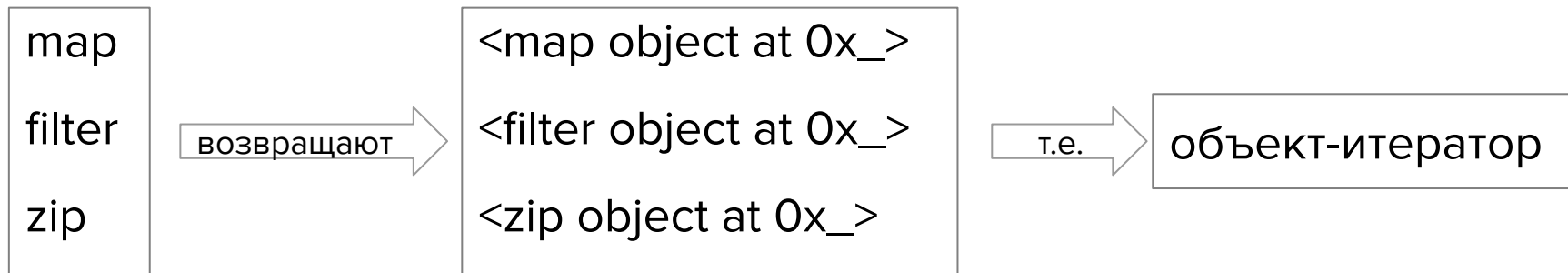
Для вопросов по курсу:

Иванов Дмитрий Владимирович, [dmitry.ivanov@moevm.info](mailto:dmitry.ivanov@moevm.info)

Префикс в теме письма [CS\_23XX]



# Рассматриваемые функции в Python



- Что такое объект-итератор (или просто -- итератор)?
- Что такое итерируемый объект?

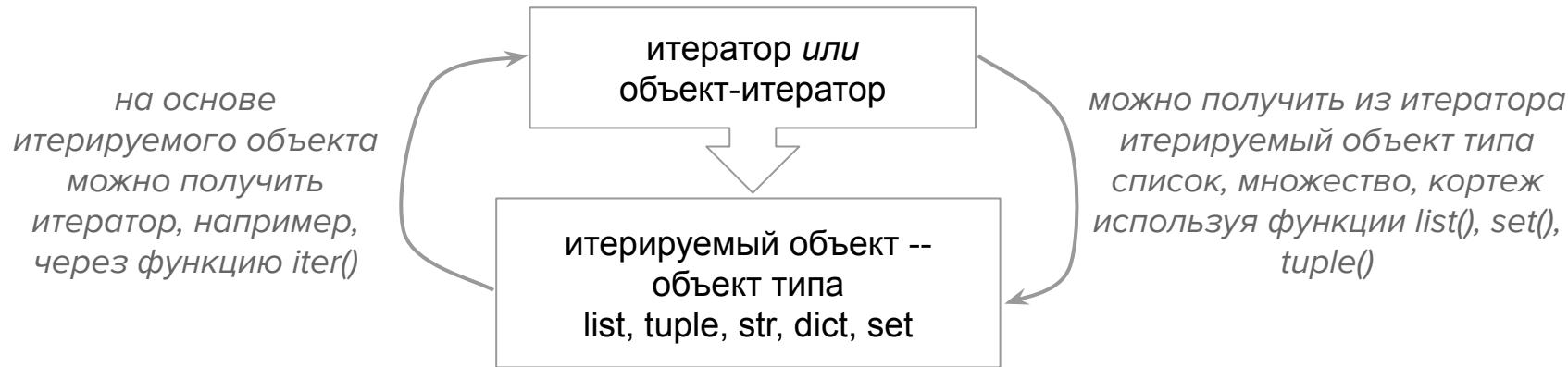


# Итератор и итерируемый объект

Свойство	Итерируемый объект ( <b>iterable</b> или iterable object)	Итератор ( <b>iterator</b> или iterator object)
1) функция iter()	возвращает новый итератор	возвращает текущий итератор
2) функция next()	не работает!	переход к новому элементу
3) обход в цикле	элементы доступны вне цикла	элементы извлекаются и больше не доступны
4) взаимный переход	можно получить итератор	можно создать итерируемый объект



# Итератор и итерируемый объект. Взаимный переход





## Функция map()

`map(<function>, <iterable_1> [, <iterable_2>, ... ,<iterable_N-1> ])`

- применяет функцию **function** к элементам итерируемого объекта(-ов) **iterable\_N**
- формирует **итератор** из измененных элементов объекта(-ов) **iterable\_N**
- **function** обязательно должна возвращать значение
- количество объектов **iterable\_N** равно количеству аргументов в функции **function**



## Функция filter()

`filter(<function>, <iterable>)`

- применяет функцию **function** к элементам итерируемого объекта **iterable**
- формирует **итератор** из тех элементов объекта **iterable**, для которых **function** вернула **True**
- **function** должна возвращать результат, приводимый к **True** и **False**



# Функция zip()

zip(\*iterables)

- формирует **итератор** из пар (кортежей) нулевых, первых и т.д. элементов объектов **iterables**
- полезно в циклах, когда есть необходимо итерироваться по нескольким элементам сразу
- удобна для создания словарей



# lambda-выражения

`lambda argument1, argument2,..., argumentN : expression`

- анонимные (т.е. не имеющие имени) функции, описываемые сразу в том месте, где эту функцию необходимо сразу вызвать
- `argument1, argument2,..., argumentN` - аргументы (как входные аргументы в случае определения функции через **def**)
- **expression** - выражение, в котором участвуют аргументы





# Источники и очень полезные ссылки

- Иерархия исключений:  
<https://pythonworld.ru/ipy-dannyx-v-python/isklyucheniya-v-python-konstrukciya-try-except-dlya-obrabotki-i-sklyuchenij.html>
- Функциональное программирование:  
[https://ru.wikipedia.org/wiki/Функциональное\\_программирование](https://ru.wikipedia.org/wiki/Функциональное_программирование)
- Функция filter: <https://docs.python.org/3/library/functions.html#filter>
- Функция map: <https://docs.python.org/3/library/functions.html#map>
- Функция zip: <https://docs.python.org/3/library/functions.html#zip>
- Функция next(): <https://docs.python.org/3/library/functions.html#next>
- Функция iter(): <https://docs.python.org/3/library/functions.html#iter>
- Ключевое слово lambda: <https://docs.python.org/3/reference/expressions.html#lambda>
- lambda-выражения: <https://docs.python.org/3/tutorial/controlflow.html#lambda-expressions>

# Вопросы по курсу можно задавать:

---

Иванов Дмитрий Владимирович  
[dmitry.ivanov@moevm.info](mailto:dmitry.ivanov@moevm.info)