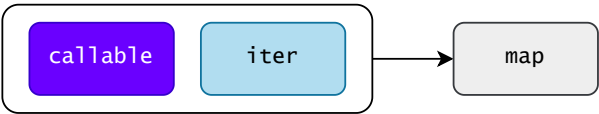
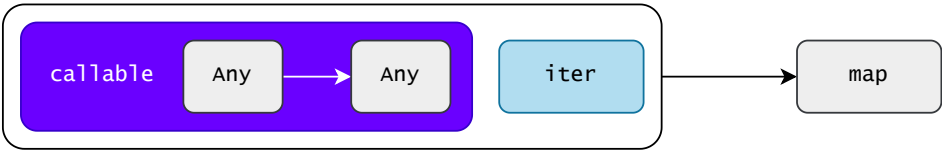


MAP

map(callable, iterable)



Первый аргумент это вызываемый тип, второй - итерируемый



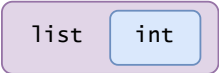
Вызываемый тип должен принимать на вход только 1 аргумент
Но стоит обратить внимание что на вход callable будут подаваться элементы из iter

Пример 1:

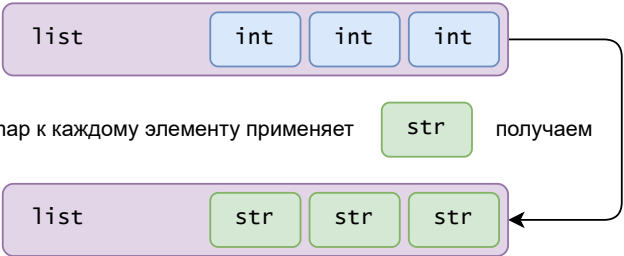
```
list1 = [1, 2, 3, 4, 5]
list2 = list(map(str, list1))
print(list2) # ['1', '2', '3', '4', '5']
```

Список является итерируемым объектом

Исходный тип списка

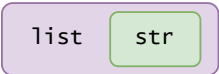


```
list1 = [1, 2, 3, 4, 5]
```



```
list2 = list(map(str, list1))
```

Тип нового списка



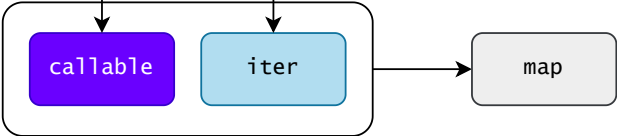
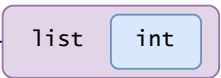
```
['1', '2', '3', '4', '5']
```

Рассмотрим на типах

`str` Любой объект преобразует в строку



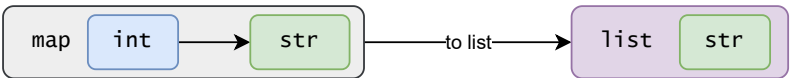
Исходный тип списка
Элементы только целые числа (int)



Получаем объект map, который будет должен преобразовать целые числа списка в строки



После преобразования в список map выполнит данные действия над каждым элементом исходного списка и мы получим новый

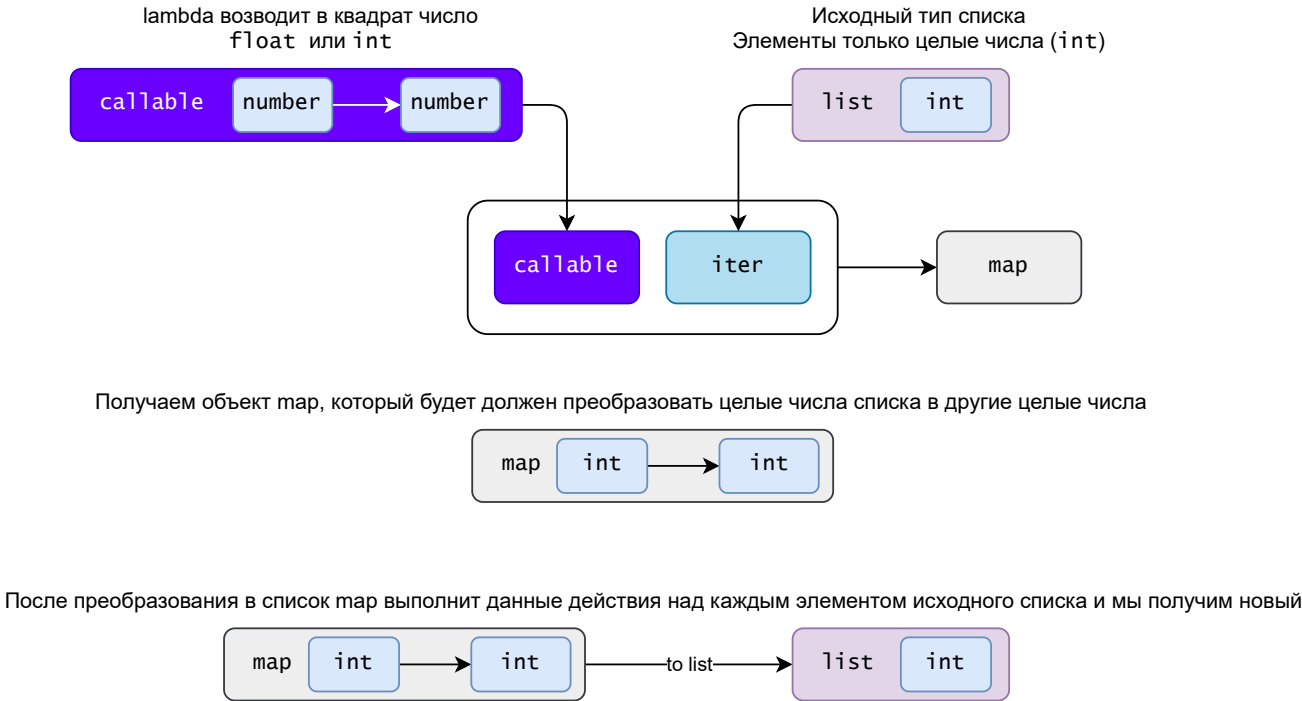


Пример 2:

```
list1 = [1, 2, 3, 4, 5]
list2 = list(map(lambda x: x ** 2, list1))
print(list2)  # [1, 4, 9, 16, 25]
```



Рассмотрим на типах



Пример 3:

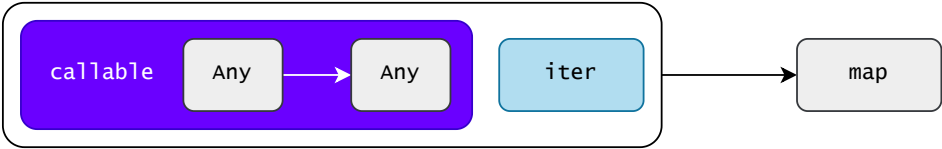
```
list1 = [1, 2, 3, 4, 5.0]

def power(x, n):
    return x ** n

list2 = list(map(power, list1))
```

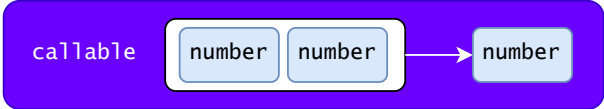
Ошибка выполнения map
power() missing 1 required positional argument: 'n'

Рассмотрим на типах



Вызываемый тип должен принимать на вход только 1 аргумент, любые
Но стоит обратить внимание что на вход callable будут подаваться элементы из iter

Наша функция требует 2 аргумента на вход



Можно исправить, если укажем второй аргумент, как необязательный:

```
list1 = [1, 2, 3, 4, 5]

def power(x, n = 2):
    return x ** n

list2 = list(map(power, list1))
```

Теперь функция требует 1 аргумент на вход, второй необязательный,
так как у него есть значение по умолчанию

