map(fun_name, iterable)

```
def sqr(n)
    return n**2
list(map(sqr, [1,2,3])) = [1,4,9]
```

```
def identity(n)
    return n        [1,2,3]
```

1 ⟶ sqr ⟶ 1

2 ⟶ sqr ⟶ 4

3 ⟶ sqr ⟶ 9

sum([1,2,3]) = 6

list(map(sum, [[1,1,1], [2,2,2], [3,3,3]]))

[1,1,1], ⟶ sum ⟶ 3

[2,2,2], ⟶ sum ⟶ 6

[3,3,3] ⟶ sum ⟹ 9

[a, b, c]
a → fun → a'
b → fun → b'
c → fun → c'
map

list1 = ["python", "java", "C"]

```
def fun1(string1)
    return string1[0]
```

map(fun1, list1)

python → fun1 → p
java → fun1 → j
C → fun1 → C

```
list1 = ['python','java','c']
def fun1(string1):
    return (string1,'okay')

mapped_obj = map(fun1,list1)
final_obj = list(mapped_obj)

print(final_obj)
```

```
[('python', 'okay'), ('java', 'okay'), ('c', 'okay')]
```

We are returning a tuple in the function

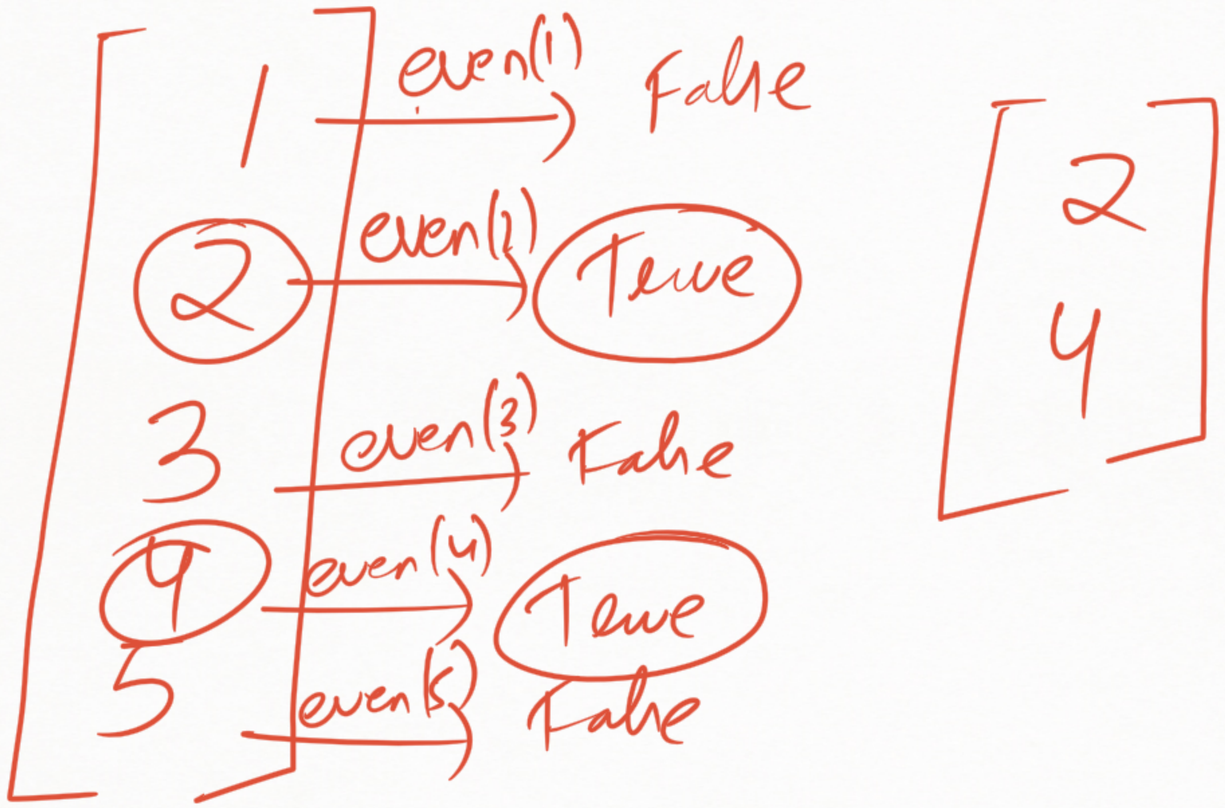
This is a tuple

"python" → fun1 → ('python', 'okay')

"java" → fun1 → ('java', 'okay')

"c" → fun1 → ('c', 'okay')

filter(fun_name, iterable_name)



[2, 4]

```
def even(n):
    if n%2 == 0:
        return True
    return False
```

Lambda functions

→ One liner functions

```
def fun1(a)
    return a+5
```

} lambda a: a+5

→ They can also be named, but they are often used without a name.