Lambda equation explanation.

For example we have array with few zeroes:

```
o1 = [0, 1, 2, 3, 0, 5, 6, 0, 8, 0]
o1
```

```
[0, 1, 2, 3, 0, 5, 6, 0, 8, 0]
```

Now we can create filter object, that will include indexes of the zeroes in initial array ("range(len(o1))" is just another way to write "for x = 0; x < len(o1); x++") :

```
f = filter(lambda x: o1[x] == 0, range(len(o1)))
```

Let's output this indexes and list elements:

```
for x in f:
    print("At index {} we have value {}".format(x,o1[x]))
```

```
At index 0 we have value 0
At index 4 we have value 0
At index 7 we have value 0
At index 9 we have value 0
```

The second part: what if we want to include elements from second list? Let's create second list with zeroes in another places:

```
o2 = [9, 0, 8, 7, 6, 0, 0, 3, 0, 1]
o2
```

```
[9, 0, 8, 7, 6, 0, 0, 3, 0, 1]
```

Here is an example of working with our lambda equation on the array (dataframe is just a improved array)

```
# We take copy of the initial array (that one we want to fill missed values)
o3 = o1.copy()
# For every missed item we take item with same index from another array
for i in filter(lambda x: o1[x] == 0, range(len(o1))):
    # i - missed index in the o1, so we just take element from o2 to cover it
    o3[i] = o2[i]
o3
```

```
[9, 1, 2, 3, 6, 5, 6, 3, 8, 1]
```

Now we want to do same operation in opposite way: fill empty o2 elements with o1:

```
# We take copy of the initial array (that one we want to fill missed values)
o4 = o2.copy()
# For every missed item we take item with same index from another array
for i in filter(lambda x: o2[x] == 0, range(len(o2))):
    # i - missed index in the o2, so we just take element from o1 to cover it
    o4[i] = o1[i]
o4
```

```
[9, 1, 8, 7, 6, 5, 6, 3, 8, 1]
```