Refactor Example: Split loop

Starter code: https://github.com/carlosgonzalez96/split-loop-example

Code Context: There is some gathered data in JSON format about scorpions. In the JSON file there two pieces of data: “length” which contains an array of ten measured scorpion in centimeters (cm) and “venom” which is the amount of venom amount that has been extracted from ten scorpions in milligrams (mg).

What is a Split Loop?

Split loop refactoring is used when we see a loop that is doing two or more things at once. This refactoring may be simple, but it has some controversy because some would argue that using one loop to do multiple things is time efficient. This is something I would agree with but if you are doing two or more thing at once in the same loop then when ever you would need to change or modify the loop you would need to understand the loop to make those changes. Splitting the loop would ensure that you know what the behavior is doing.

Example:

Here is some starter code that calculates the average length of ten scorpions from the data and the total amount of venom extracted from scorpions (only adding venom that is more than 0.40mg for a scorpion).

In this example there is loop that does the calculation, but each calculation is doing two different things (calculation-wise).

```
import json

scorpions =  '{ "length":[0.9,21,1,10,21,5,18,4,15,2], "venom":[0.39,0.62,0.20,0.45,0.33,0.62,0.55,0.40,0.22,0.38]}'
scorpion_data = json.loads(scorpions)

total_venom_amount = 0
average_length = 0

for data in range(10):
    average_length += scorpion_data["length"][data]
    if scorpion_data["venom"][data] > 0.40:
        total_venom_amount += scorpion_data["venom"][data]

average_length = average_length / len(scorpion_data["length"])

print "Average length: " + str(average_length) + "cm"
print "Total amount of venom extracted: " + str(total_venom_amount) + "mg"

#Output
#Average length: 9.79cm
#Total amount of venom extracted: 2.24mg
```

The first step is to copy the loop. Yes, it does feel weird that we are copying a loop to do another task when it could do it in one.

```
import json

scorpions =  '{ "length":[0.9,21,1,10,21,5,18,4,15,2], "venom":[0.39,0.62,0.20,0.45,0.33,0.62,0.55,0.40,0.22,0.38]}'
scorpion_data = json.loads(scorpions)

total_venom_amount = 0
average_length = 0

for data in range(10):
    average_length += scorpion_data["length"][data]
    if scorpion_data["venom"][data] > 0.40:
        total_venom_amount += scorpion_data["venom"][data]

for data in range(10):
    average_length += scorpion_data["length"][data]
    if scorpion_data["venom"][data] > 0.40:
        total_venom_amount += scorpion_data["venom"][data]

average_length = average_length / len(scorpion_data["length"])

print "Average length: " + str(average_length) + "cm"
print "Total amount of venom extracted: " + str(total_venom_amount) + "mg"
```

Second step is to remove duplication from each loop.

```
import json

scorpions =  '{ "length":[0.9,21,1,10,21,5,18,4,15,2], "venom":[0.39,0.62,0.20,0.45,0.33,0.62,0.55,0.40,0.22,0.38]}'
scorpion_data = json.loads(scorpions)

total_venom_amount = 0
average_length = 0

for data in range(10):
    average_length += scorpion_data["length"][data]
    if scorpion_data["venom"][data] > 0.40: #this gets deleted
        total_venom_amount += scorpion_data["venom"][data] #this gets deleted

for data in range(10):
    average_length += scorpion_data["length"][data] #this gets deleted
    if scorpion_data["venom"][data] > 0.40:
        total_venom_amount += scorpion_data["venom"][data]

average_length = average_length / len(scorpion_data["length"])

print "Average length: " + str(average_length) + "cm"
print "Total amount of venom extracted: " + str(total_venom_amount) + "mg"
```

result

```
import json

scorpions =  '{ "length":[0.9,21,1,10,21,5,18,4,15,2], "venom":[0.39,0.62,0.20,0.45,0.33,0.62,0.55,0.40,0.22,0.38]}'
scorpion_data = json.loads(scorpions)

total_venom_amount = 0
average_length = 0

for data in range(10):
    average_length += scorpion_data["length"][data]

for data in range(10):
    if scorpion_data["venom"][data] > 0.40:
        total_venom_amount += scorpion_data["venom"][data]

average_length = average_length / len(scorpion_data["length"])

print "Average length: " + str(average_length) + "cm"
print "Total amount of venom extracted: " + str(total_venom_amount) + "mg"
```

We are also going to put each loop respective global variable right above and their print statements down below. This is a type of refactoring called *Slide Statements*.

```
import json

scorpions =  '{ "length":[0.9,21,1,10,21,5,18,4,15,2], "venom":[0.39,0.62,0.20,0.45,0.33,0.62,0.55,0.40,0.22,0.38]}'
scorpion_data = json.loads(scorpions)

average_length = 0
for data in range(10):
    average_length += scorpion_data["length"][data]

average_length = average_length / len(scorpion_data["length"])
print "Average length: " + str(average_length) + "cm"

total_venom_amount = 0
for data in range(10):
    if scorpion_data["venom"][data] > 0.40:
        total_venom_amount += scorpion_data["venom"][data]

print "Total amount of venom extracted: " + str(total_venom_amount) + "mg"
```

And there we have it, two loops. Again, it is a simple refactor that may turn heads but now the code is more open to other modifications, it is more quickly readable, and it gives us a chance to refactor the code even more.

Bonus:

There is still room form much refactoring. For example, we could extract the loops to their own functions.

```
import json

scorpions =  '{ "length":[0.9,21,1,10,21,5,18,4,15,2], "venom":[0.39,0.62,0.20,0.45,0.33,0.62,0.55,0.40,0.22,0.38]}'
scorpion_data = json.loads(scorpions)

def average_scorpion_length():
    average_length = 0
    for data in range(10):
        average_length += scorpion_data["length"][data]

    average_length = average_length / len(scorpion_data["length"])
    print "Average length: " + str(average_length) + "cm"

def total_venom_extract():
    total_venom_amount = 0
    for data in range(10):
        if scorpion_data["venom"][data] > 0.40:
            total_venom_amount += scorpion_data["venom"][data]

    print "Total amount of venom extracted: " + str(total_venom_amount) + "mg"

average_scorpion_length()
total_venom_extract()
```

Additionally, we could also *replace loop with pipeline* for our average scorpion length function.

```
import json

scorpions =  '{ "length":[0.9,21,1,10,21,5,18,4,15,2], "venom":[0.39,0.62,0.20,0.45,0.33,0.62,0.55,0.40,0.22,0.38]}'
scorpion_data = json.loads(scorpions)

def average_scorpion_length():

    average_length = reduce(lambda total,data: data+total, scorpion_data["length"])
    print "Average length: " + str(average_length) + "cm"

def total_venom_extract():
    total_venom_amount = 0
    for data in range(10):
        if scorpion_data["venom"][data] > 0.40:
            total_venom_amount += scorpion_data["venom"][data]

    print "Total amount of venom extracted: " + str(total_venom_amount) + "mg"

average_scorpion_length()
total_venom_extract()
```

Another refactor is *substitute algorithm* for our for loop and if statement in our total venom extract function.

```
import json

scorpions =  '{ "length":[0.9,21,1,10,21,5,18,4,15,2], "venom":[0.39,0.62,0.20,0.45,0.33,0.62,0.55,0.40,0.22,0.38]}'
scorpion_data = json.loads(scorpions)

def average_scorpion_length():

    average_length = reduce(lambda total,data: data+total, scorpion_data["length"])
    print "Average length: " + str(average_length) + "cm"

def total_venom_extract():

    total_venom_amount = sum([i for i in scorpion_data["venom"] if i ≥ 0.40])
    print "Total amount of venom extracted: " + str(total_venom_amount) + "mg"

average_scorpion_length()
total_venom_extract()
```

And lastly, we can refactor the print statements by removing them from each function and print each function outside. Also, for each function, return the variable and delete the function calls.

```
import json

scorpions =  '{ "length":[0.9,21,1,10,21,5,18,4,15,2], "venom":[0.39,0.62,0.20,0.45,0.33,0.62,0.55,0.40,0.22,0.38]}'
scorpion_data = json.loads(scorpions)

def average_scorpion_length():
    average_length = reduce(lambda total,data: data+total, scorpion_data["length"])
    return average_length

def total_venom_extract():
    total_venom_amount = sum([i for i in scorpion_data["venom"] if i > 0.40])
    return total_venom_amount

print "Average length: " + str(average_scorpion_length()) + "cm" + "\n" + "Total amount of venom extracted: " + \
    str(total_venom_extract()) + "mg"
```