How do functions break up?

They stop calling each other!

Thank you Manav!

Source: https://twitter.com/catalinmpit/status/1442571813364568071?lang=en

CMPT 120

Lecture 13 – Practice Exam 3 - **SOLUTION**

Feedback on Practice Exam #2

- Very wel1 done! ☺
  - Very few answers with repeated code
- Don't forget header comment block and comments though!

In-Class Activity

Course grading scheme on our course website: ***Best 7 in-class exercises out of 10: 1% each, for a total of 7%***

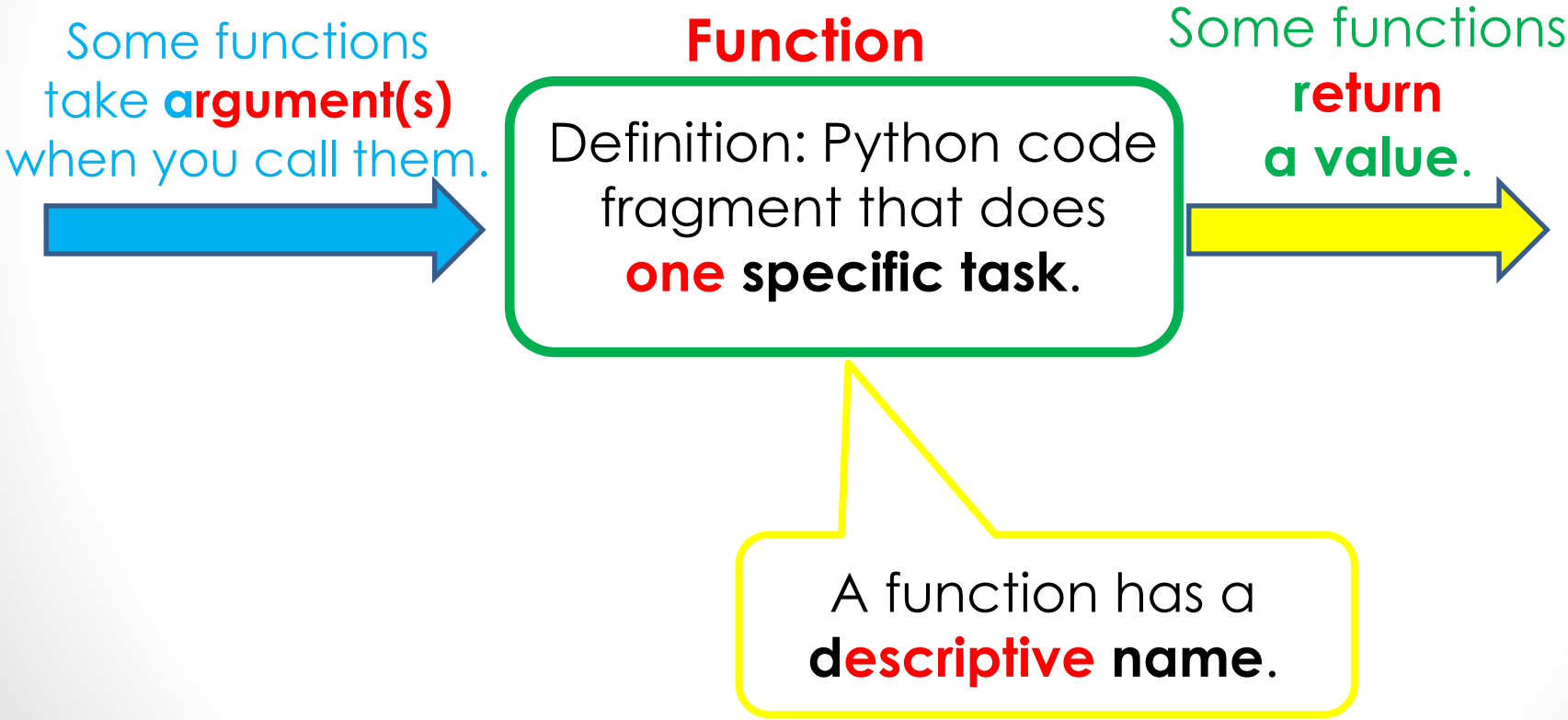
- Our in-class activity #3 -> 1%
  - Write your answer to **Q. _5_** on the provided sheet of paper
  - Write your **lastname**, **firstname** and **student number** on the provided sheet of paper
  - At the end of today's class, hand in your sheet of paper in the appropriate pile:
    - **Pile 1** -> if your lastname start with a letter that is between '**A**' and '**L**'
      - **Pile 1** is on your **left-hand side** of the classroom
    - **Pile 2** -> if your lastname start with a letter that is between '**M**' to letter '**Z**'
      - **Pile 2** is on your **right-hand side** of the classroom

Try to answer the questions **1st without using your computer**, then confirm your answer using **Python IDLE shell**!

Part 1 - Theory and Understanding

Question 0

- Fill in the blanks:



Strings and Lists

1. What would the variable `aSlice` contain once the following Python code fragment has executed?

```
city = "Cranbrook"
aSlice = city[:5]
```

A. `"Cranbr"`
B. `"Cranb"`
C. `""` (empty string)
D. An error message
E. None of the above.

2. What would the variable `found` contain once the following Python code fragment has executed?

```
city = "Cranbrook"
found = city.find("k")
```

A. `8`
B. `-1`
C. `9`
D. An error message.
E. None of the above.

3. What does the following Python code fragment produce?

```
numbers = list(range(10))
aSlice = numbers[:8:2]
print(aSlice)
```

A. `[]` (i.e., an empty list)
B. `[0, 2, 4, 6, 8]`
C. `[0, 2, 4, 6]`
D. An error message
E. None of the above

4. What does the following Python code fragment produce?

```
grades = ['B', 'A', 'D']
print(grade[3])
```

A. `['B', 'A', 'D']`
B. `['B', 'A']`
C. `[]` (i.e., an empty list)
D. An error message.
E. None of the above

Question 5 - Are there *functions* in the real world?

1. Mom says to Louise "**Please, can you go clean your bedroom?**"
   - Specific task: Clean bedroom
   - Argument(s): None required
   - Returned value: None returned

2. Mom says to Louise "**Here's $5, please, can you go buy a bag of apples?**"
   - Specific task: Buy a bag of apples
   - Argument(s): Mom gives $5 to Louise, so $5 is what Louise requires to perform the specific task of buying a bag of apples
   - Returned value: Louise returns a bag of apples (and perhaps some change ☺) to Mom

Question 5 - Are there *functions* in the real world?

3. Mom says to Louise "**Please, can you find my cell phone?**"
   - Specific task: Find Mon's cell phone.
   - Argument(s): None required
   - Returned value: The found cell phone

Following the examples on the pervious slide, can you complete this slide?

4. Mom says to Louise "**Please, can you put these bags on the kitchen counter?**"
   - Specific task: Put bags on kitchen counter
   - Argument(s): "These" bags
   - Returned value: None returned

Ditto here!

Question 6

Let's convert this **while** loop to a **for** loop ?

```
fruit = ["banana", "apple", "plum"]
index = 0
while index < len(fruit):
    print(fruit[index])
    index = index + 1
```

SOLUTION:

```
for index in range(len(fruit)) :
    print(fruit[index])
```

Question 7

What do the following statements produce?

SOLUTION:

a. `"123456789"[2:8:3]` -> 36

b. `"123456789"[2:8:-3]` -> nothing is printed

c. `"123456789"[-2:-8:-3]` -> 85

d. `"123456789"[-2:-8:3]` -> nothing is printed

e. `"123456789"[8:2:3]` -> nothing is printed

f. `"123456789"[8:2:-3]` -> 96

g. `"123456789"[-8:-2:-3]` -> nothing is printed

h. `"123456789"[-8:-2:3]` -> 25

Part 2 - Coding

Solve the following problems by writing a Python program **on a piece of paper without using your computer**!

Possible solution posted on our course website

Question 8

Problem Statement:

Write an **Predicting Chatbot** that asks your name, your age and a number of years **x** and predicts how old you will be in **x** years, as shown in the **sample run** below:

```
Welcome to the Predicting Chatbot!
This Bot predicts how old you will be in x years.
Please, enter ...
        your name (letters): BillyBob
        your age (an integer): 27
        the number of years (x) for the prediction: 56

Dear BillyBob:
In 56 years, you will be 83 years old.
Bye!
```

Question 8 (cont'd)

Problem Statement: (cont'd)

You cannot assume the user will always enter a string for the name and an integer for the age and for **x** as illustrated in the following **sample runs**:

```
Welcome to the Predicting Chatbot!
This Bot predicts how old you will be in x years.
Please, enter ...
        your name (letters): Roy99
        your age (an integer): 34
You entered an invalid name: Roy99.
Bye!
```

```
Welcome to the Predicting Chatbot!
This Bot predicts how old you will be in x years.
Please, enter ...
        your name (letters): Roy
        your age (an integer): ten
You entered an invalid age: ten.
Bye!
```

```
Welcome to the Predicting Chatbot!
This Bot predicts how old you will be in x years.
Please, enter ...
        your name (letters): Roy
        your age (an integer): 10
        the number of years (x) for the prediction: lots
You entered an invalid number of years: lots.
Bye!
```

Possible solution posted on our course website

Question 9

Problem Statement:

Write a **Milk Survey Bot** that asks the user whether s/he has tried almond, coconut, cow, goat, hemp, oat, rice, and/or soy milk.

Your **Milk Survey Bot** must then print the number of different kinds of milk the user has tried.

Here is a **sample run**:

```
How many different types of milk have you tried?
For example, have you tried ...
... almond milk? (y/n): y
... coconut milk? (y/n): n
... cow milk? (y/n): y
... goat milk? (y/n): y
... hemp milk? (y/n): n
... oat milk? (y/n): y
... rice milk? (y/n): n
... soy milk? (y/n): y
Wow! You have tried 5 different kinds of milk (out of 8).
```

Question 9 (cont'd)

Doing this is called **hard-coding** a value. Not a good idea! What would happen if we add and/or remove types of milk from our list?

BONUS Part 1:

Write your Python code such that it does not include the actual number 8 in its last print statement like this: `print("Wow! You … (out of 8).")`
Instead, your program must compute this number by calling a function.

Here is a **sample run** with the BONUS Part 1:

```
How many different types of milk have you tried?
For example, have you tried ...
... almond milk? (y/n): y
... coconut milk? (y/n): n
... cow milk? (y/n): y
... goat milk? (y/n): y
... hemp milk? (y/n): n
... oat milk? (y/n): y
... rice milk? (y/n): n
... soy milk? (y/n): y
Wow! You have tried 5 different kinds of milk (out of 8).
```

This number 8 here!

Question 9 (cont'd)

Problem Statement: (cont'd)

After your **Milk Survey Bot** has printed the number of different kinds of milk the user has tried, it then prints the names of the milks the user has tried.

Hint: This printing must be done after (outside) the loop.

Here is a **sample run** with the BONUS Part 2:

```
How many different types of milk have you tried?
For example, have you tried ...
... almond milk? (y/n): y
... coconut milk? (y/n): n
... cow milk? (y/n): y
... goat milk? (y/n): y
... hemp milk? (y/n): n
... oat milk? (y/n): y
... rice milk? (y/n): n
... soy milk? (y/n): y
Wow! You have tried 5 different kinds of milk (out of 8).
BONUS PART - You tried:
almond
cow
goat
oat
soy
```