Þátttaka í kennslustund, viku 7. kennslustund B

Ylfa - ylfas23@ru.is
Donna - donn24@ru.is
Bjarki - bjarkik24@ru.is

Umræða í krossaspurningum | Quiz 12: More on functions

1. Svöruðum C = def buy_martini(age=37, shaken=True, stirred=False): - rétt

2.Svöruðum A = greeting (name="James", 37) |
   - This is incorrect because you cannot mix keyword and positional arguments in this order. Keyword arguments must always follow positional arguments.

3. Svöruðum C = def convert_float_to_int(num: float) → int:
   - The argument num is expected to be of type float, and this is indicated by num: float.
   - The return type is specified with → int, meaning the function is expected to return an integer.

4. Svöruðum A = Function declarations and associated program statements. Sem var ekki rétt
   - Rétta svarið er C = A name and a Python object associated with that name. (tilraun #2)

5. Svöruðum A = [10, 20, 30] - rétt

6. Svöruðum D = [1, 20, 30] - rétt

7. Svöruðum D. = 21 | print(my_num) er að biðja okkur um að prenta út my_num = 21, - rétt
   ef þetta væri print(increment(my_num)) þá myndum við fá none því að def increment(num) en ekki def increment(my_num)

8. Svöruðum B = Það myndi koma upp error vegna þess að new_int er ekki defined - hún er bara í fallinu sem er kallið í - rétt

9. Svöruðum C = Nothing, an error will occur. - sem var ekki rétt
   - Rétta svarið er B = None of the other answers. (tilraun #2)

10. Svöruðum C =
   - id(my_list1) == id(my_list2): skilar true, my_list2 is er það sama og my_list1
   - id(my_list1) == id(my_list3): Skilar False, my_list3 er nýr listi þótt hann sé eins þá er er þetta á öðrum stað.

Umræða í Forritunarverkefni

Problem B | Get numbers from file

Hver byrjaði með þetta verkefni: Kennarinn
Hver kláraði verkefnið í tölvu: Allir

Örstutt útskýring á verkefninu:

Write a program that extracts all numbers from a given text file. The results should be saved in a list and finally printed out in sorted order.

First, the program should ask the user for a file name, repeatedly until the user enters a name of a file that actually exists, as evidenced by the fact that it can be opened. The file will be located in the /src/ directory.

Input:

The input consists of one or more line. At least one line will contain the name of a file that is accessible to the program.

Output:

The output should consist of m lines.

The output should read "not found! Please try again." if file not found in directory.

Finally, the last line should contain an ordered list of all numbers found in the file, separated by commas and surrounded by square brackets.

As soon as the program receives a name of a file that it can open, it should stop asking for input and process the file.

Stutt útskýring á lausnar hugmyndinni:

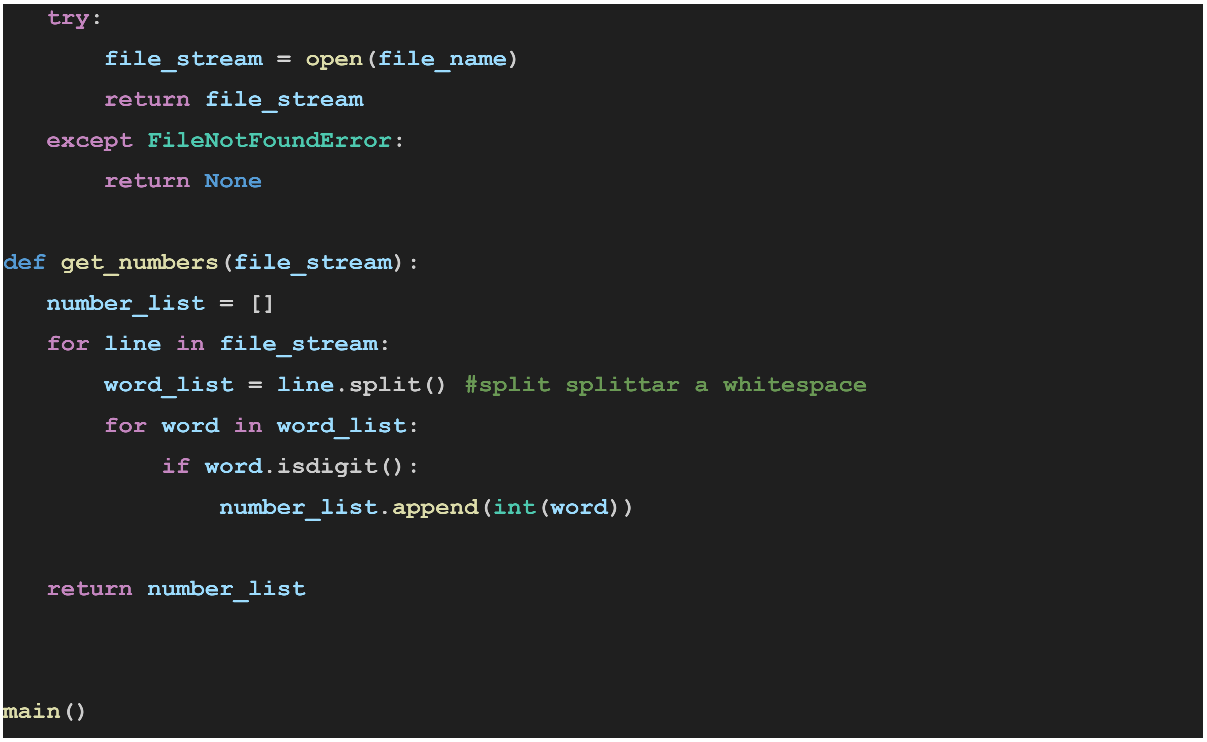
```
#open file according to input
#repeat until file found
# Get numbers from file into a list
#print the list sorted

def main():
   file_stream = get_file_stream()
   number_list = get_numbers(file_stream)
   print(sorted(number_list))

def get_file_stream():
   file_name = input()
   file_stream = open_file(file_name)
   while file_stream is None:
       print(f'{file_name} not found! Please try again.')
       file_name = input()
       file_stream = open_file(file_name)

   return file_stream

def open_file(file_name):
```



```
    try:
        file_stream = open(file_name)
        return file_stream
    except FileNotFoundError:
        return None

def get_numbers(file_stream):
    number_list = []
    for line in file_stream:
        word_list = line.split() #split splittar a whitespace
        for word in word_list:
            if word.isdigit():
                number_list.append(int(word))

    return number_list


main()
```

Ef einhver vandamál komu upp

- **Hvert var vandamálið?**

Var með villu í while lykkjunni.

- **Hvernig var það leyst?**

while file_stream is None:

Problem E | Analyze list

Hver byrjaði með þetta verkefni: Bjarki
Hver kláraði verkefnið í tölvu:

Örstutt útskýring á verkefninu:
We are writing a program that given a list of positive integers prints out the following:

1. The list
2. The sorted version of the list
3. The sorted list of the composite numbers (not prime numbers)
4. Minimum,maximum and average values in the list.

You need to print out an error message if the user enters invalid values into the list.

Input:

One or more lines, each containing a list. If invalid numbers are chosen, the program should ask for a valid list until a valid list is chosen.
Output:

At least 4 lines. First if the user input an invalid list the program should display: “Incorrect input! Please try again.” for every invalid input until a valid one is encountered. Then the program should print the following four lines:

1. “Input list: {l}.
2. “Sorted list: {s} where s is the list sorted
3. “Composite list: {c}, where c is a list of composite numbers in l.
4. “Min: {lmin}, Max:{lmax}. Average: {u}

Stutt útskýring á lausnar hugmyndinni:

```
import math

def main():
   input_list = get_correct_data()

   print(f"Input list: {input_list}")

   sorted_list = sorted(input_list)
   print(f"Sorted list: {sorted_list}")

   composite_list = sorted(set([n for n in input_list if not is_prime(n) and n >
1]))
   print(f"Composite list: {composite_list}")

   lmin = min(input_list)
   lmax = max(input_list)
   avg = sum(input_list) / len(input_list)
   print(f"Min: {lmin}, Max: {lmax}, Average: {avg:.2f}")


def get_correct_data() -> list:
   """Asks user repeatedly for input until valid."""
   while True:
       input_list = get_data()
       if input_list is None:
           print("Incorrect input! Please try again.")
       else:
           return input_list
```

```
def get_data() -> list:
   """Returns a list of positive integers input by the user.

   Returns None if the input contains non-integers or integers < 0.
   """
   try:
       input_list = list(map(int, input().split(',')))
       if all(n > 0 for n in input_list):
           return input_list
       else:
           return None
   except ValueError:
       return None



def is_prime(n: int) -> bool:
   """Returns True if the given positive number is prime and False otherwise."""
   if n < 2:
       return False
   for i in range(2, int(math.sqrt(n)) + 1):  # Feel free to improve this function
if you like.
       if n % i == 0:
           return False
   return True


if __name__ == "__main__":
   main()
```

Ef einhver vandamál komu upp

- **Hvert var vandamálið?**
  Not accepted in Kattis but works. First because my composite list was not sorted. Then because duplicates were printed in composite definition.
- **Hvernig var það leyst?**
  sorted() and set added to the composite definition.

Problem C | RemoveOdds

Hver byrjaði með þetta verkefni: Donna
Hver kláraði verkefnið í tölvu:

Örstutt útskýring á verkefninu:
The goal of the project is to write two functions, extract_evens and remove_odds, to process integer lists:

- extract_evens returns a new list containing only even numbers from the given list, without modifying the original list.
- remove_odds modifies the original list by removing all odd numbers, leaving only even numbers.

Input:
A list of integers l with length between 1 and 1000, where each integer is between 1 and 100.
Output:

- For extract_evens: A new list containing only the even numbers from the input list.
- For remove_odds: The original list is modified in place, so it only contains even numbers, and no return value is needed.

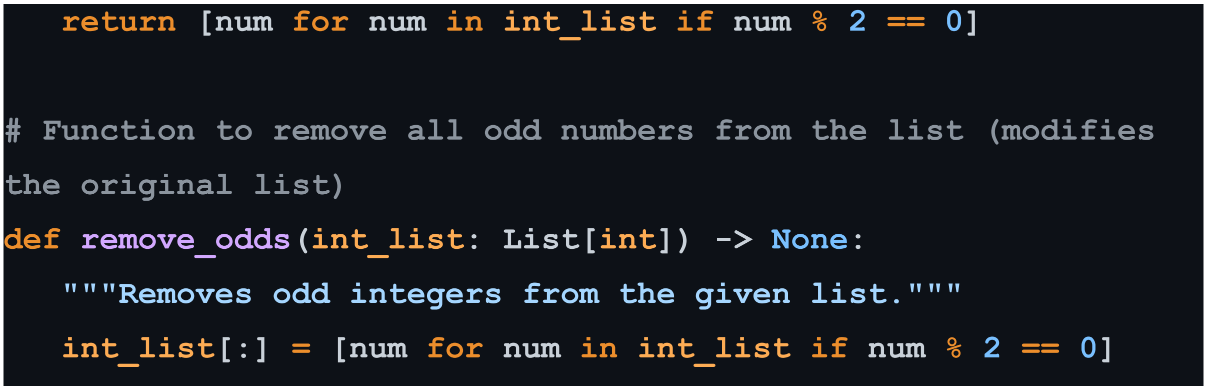
Stutt útskýring á lausnar hugmyndinni:
The solution involves using list comprehensions for both functions:

- extract_evens generates a new list containing only even numbers from the input list. This ensures the original list is not changed.
- remove_odds modifies the input list by filtering out odd numbers. This is done in-place using slice assignment, which updates the list directly.

removeOdds Solution

```
from typing import List

# Function to return a new list with only even numbers (does
not modify the original list)
def extract_evens(int_list: List[int]) -> List[int]:
   """Returns a new list with only the even integers from the
given list."""
```



```python
    return [num for num in int_list if num % 2 == 0]

# Function to remove all odd numbers from the list (modifies the original list)
def remove_odds(int_list: List[int]) -> None:
    """Removes odd integers from the given list."""
    int_list[:] = [num for num in int_list if num % 2 == 0]
```

Ef einhver vandamál komu upp

- **Hvert var vandamálið?**
  While implementing the remove_odds function, I initially had trouble understanding how to modify the original list in place. I wasn't sure if I should return a new list or update the same list without creating a new one.
- **Hvernig var það leyst?**
  I learned that lists in Python are mutable, and I could use slice assignment (int_list[:] = ...) to modify the original list directly. This allowed me to filter out the odd numbers and ensure that the list is updated without creating a new list.

Problem A | Music

Hver byrjaði með þetta verkefni: Ylfa
Hver kláraði verkefnið í tölvu:

Örstutt útskýring á verkefninu:

Input:

The function should accept three strings, g, b and s, in this order, as parameters, but in case some of the arguments are not provided the function should also default to the following values for those arguments that are not provided: “Classic Rock” for g, “The Beatles” for b, and “Freddie Mercury” for s.

Output:

The function should not return anything (just None), but should print out the following three lines:

- “The best type of music is {g}.”
- “The best music group is {b}.”
- “The best lead vocalist is {s}.”

Stutt útskýring á lausnar hugmyndinni:
The function state_music_opinion takes three arguments: genre, music_group, and vocalist. If no values are provided for these, it defaults to "Classic Rock" for the genre, "The Beatles" for the music group, and "Freddie Mercury" for the vocalist. The function then prints three sentences stating the best type of music, the best music group, and the best lead vocalist based on the given or default values.

```python
def state_music_opinion(genre="Classic Rock", music_group="The Beatles", vocalist="Freddie Mercury"):
   print(f"The best type of music is {genre}.")
   print(f"The best music group is {music_group}.")
   print(f"The best lead vocalist is {vocalist}.")
```

Ef einhver vandamál komu upp

- **Hvert var vandamálið?**
  Engin vandamál komu upp.

- **Hvernig var það leyst?**
  Engin vandamál komu upp.