

# Friends of snakes - Python Handin

Mie Grønkjær Bechmann Jørgensen  
[cph-mj761@cphbusiness.dk](mailto:cph-mj761@cphbusiness.dk)

Web development 2021



# Table of contents

<b>Table of contents</b>	<b>2</b>
Classes	3
Snake class	3
Owner class	4
Text decoration	5
Fancytext page	5
Text files	5
Main page	6
Adding new owner	6
Adding new snake	7
Program text	8
Finished looks	10
Start page	10
Choose 1	11
Choose 2	11
Choose 3	12
Choose 4	13
Choose 0	13
Finishing touches	13

## Classes

I opened up Visual studio and created a new Python project called SnakesAndOwners. I then proceeded to create a new empty python file for the classes I needed. One for the snakes and one for the owner.

### Snake class

I first created the snakes class in it's own empty Python file as mentioned earlier. In this I started with the constructor, or the datatype of the snake object.

first telling the program I want the snake id to begin at 0.

As asked for in the assignment the class contained the snake's name, type id and owner.

I told the program that I wanted the snake id to go up one when creating a new snake, hence the + 1. That's why when starting the id value at 0, the program knows that it has to go up 1 from 0, when creating a new snake. Had I written that the start value was 10, the program would have created the first snake with the snake id ad 11, since it goes up 1.

```
class Snake :
    # Constructor
    snakeid = 0
    # Constructor
    def __init__(self, snakename, snaketype, owner = None) :
        Snake.snakeid = Snake.snakeid + 1
        self.snakename = snakename
        self.snaketype = snaketype
        self.owner = owner
```

I then created the Get and Set values of these properties.

```
# GET
def getSnakeId(self) :
    return self.snakeid
def getSnakeName(self) :
    return self.snakename
def getSnakeType(self) :
    return self.snaketype
def getOwner(self) :
    return self.owner
# SET
def setSnakeName(self, snakename) :
    self.snakename = snakename
def setSnakeType(self, snaketype) :
    self.snaketype = snaketype
def setOwner(self, owner) :
```

```
self.owner = owner
```

I then finished the class with a ToString() method, to create a sentence using the properties I just created, that will show up when showing the snake list text.

```
# ToString
def ToString(self) :
    return self.getSnakeName() + " has the id: " +
str(self.getSnakeId()) + " and is a " + self.getSnakeType() + ". It's
owner is: " + str(self.getOwner())
```

## Owner class

I then proceeded to do the same for the Owners class.

This class only consists of the owner's name since that is the only important feature for this class.

In the constructor I again create an owner number (or id) with the same value technique.

```
class Owners :
    # Constructor
    ownnumber = 0
    # Constructor
    def __init__(self, ownername) :
        Owners.ownnumber = Owners.ownnumber +1
        self.ownername = ownername
```

I then also create the GET and SET for the properties of this class.

```
class Owners :
    # Constructor
    ownnumber = 0
    # Constructor
    def __init__(self, ownername) :
        Owners.ownnumber = Owners.ownnumber +1
        self.ownername = ownername
    # GET
    def getOwnNumber(self) :
        return self.ownnumber
    def getOwnerName(self) :
        return self.ownername
    # SET
    def setOwnerName(self, own_name) :
        self.ownername = ownername
```

Then I, again I create a ToString() for the owner class, to be shown when the owners list is shown with all the information of the new owner.

```
# ToString
def ToString(self) :
    return "Owner with id " + str(self.getOwnNumber()) + " is called"
" + self.getOwnerName()
```

## Text decoration

### Fancytext page

I use this Empty Python file to create a special attribute to the presentation of the text. I use import time, so later on in my code I use time.sleep, which creates a delay in showing the text which creates a more typewriter feel to it. I use the method flush to make sure that the program doesn't flush away the info you have created when closing. Since the program automatically flushes the files when closing.

```
import time

def fancytext(text) :
    for i in text:
        print(i, end='', flush=True)
        time.sleep(0.025)
    print('\n')
```

### Text files

I chose to create empty txt files called Snakes.txt and Owners.txt, which are the files where the lists of snakes and owners are saved into. They will be empty while the program hasn't been used. However once a new snake or owner is created by the user, they will be successfully saved into the text files. New snakes will be saved into Snakes.txt and new owners into Owners.txt. Here is a shown example of the Snakes.txt file having snakes inside the file.

UsingClasses.py	FancyText.py	Owners.txt	Snakes.txt	Owner.py	Snakes.py
1					
2			Snake id: 1. Snake name: Hally. Snake type: mamba. Owner of the snake: John.		
3			Snake id: 2. Snake name: Bruno. Snake type: mamba. Owner of the snake: John.		
4			Snake id: 1. Snake name: Loopy. Snake type: mamba. Owner of the snake: Gary.		

## Main page

The final empty Python page I create is called UsingClasses and this is the page where I will tell the code what to print. As well as the more descriptive code telling the program how to use the classes.

I remember to import all the classes as well as time, to make sure it works on this site as well. I tell what I want from the classes and since I want everything I don't choose anything specific and just choose the whole class to import.

```
from Snakes import Snake
from Owner import Owners
from FancyText import fancytext

import time

ownerslist = list()
snakelist = list()
```

## Adding new owner

Here I tell the program what to define when choosing to add a new owner to the program. So if the user was to choose "add new owner" then it would first ask what the name is for the new owner. Creating a new value for ownerName from the owners class, which is now = to an input. As well as telling the program that 'o' is equal to ownerName.

I then use fw as a new variable which tells the program that to create an open textwrapper which I will use to tell the program where to store my new owners created, which will be the owners.txt file. I can then use the fw variable again telling the program what the program has to write and save when creating a new owner. Taking the owner number (id) as well as the owner's name, which the user just told the program. then when done successfully it will close it the text wrapper as well as writing to the user a little message that the owner was added successfully.

I then use while, if and if else statements telling the program that if the statement the user put it was correct, it should ask the user if they want to add another or not. And if they don't want to do that it will go back to the first text. As well as to remember to break the loop after everything has been checked through.

```
# Adding new owner

def addOwner():
    fancytext(text = "What is the name of the owner?")
    ownerName = input()
    o = Owners(ownerName)
    fw = open("Owners.txt", "a")
    fw.write("\nOwner id: " + str(o.getOwnNumber()) + ". Owner name: " +
```

```

o.getOwnerName() + ".")
    fw.close()
    fancytext(text = "Owner added successfully as: " + o.getOwnerName()
+ " with id " + str(o.getOwnNumber()))
    while True:
        again = input("Add more owners? Y/N: ").lower()
        if (again == "y"):
            addOwner()
        elif (again == "n"):
            break
        else:
            fancytext(text = "Enter Y or N")
            continue
    break

```

## Adding new snake

If the user chooses to make a new snake, most of the code will be the same as the new owner. However here the fw variable or port will be equal to the snakes.txt, so it will save the new snake inside of the text file.

Here the first question for the user will be which type of snake is, a mamba or an anaconda, as told in the assignment description. Again using the if, if else and else statements, to tell the program the value of the user's choice. Meaning, I say if the statement (users answer) is anaconda, then the program should write a message to the user telling them that this type of snake is protected and cannot be added. So when I use else if or (elif) for telling the user if they choose mamba, that this statement is the true statement, since we haven't earlier on told the program which answer is the true answer the if else statement will see that answer as the true answer, since no other true value was given. The program begins with asking for the snake's name, and will then continue on. The program, if mamba is chosen, will then print out a string telling the user that this type of snake can be added, however it is toxic so they should be careful. If successful by choosing the mamba, then it will ask who the owner of the snake is. Then I end the loop and statement if and else statement telling the program that if the user chooses to write anything else, they will get a message saying the program does not know that type of snake, as well as telling them to try again.

I then remember to use a break and then use the same type of code as on the adding new owner code, but I change the information with snakes instead of owners. and the loop continues.

```

# Adding new snake

def addSnake():
    fancytext(text = "What is the name of the snake?")
    snakeName = input()
    while True:
        fancytext(text = "What type of snake is it? \"Mamba\" or
\"Anaconda\"?")

```

```

snakeType = input()
if (snakeType.lower() == "anaconda"):
    fancytext(text = "Oh no! Bad news! This type of snake is
protected, and cannot be added!")
    continue
elif (snakeType.lower() == "mamba"):
    fancytext(text = "Good choice! Though remember this snake is
toxic! so be careful!")
    break
else:
    fancytext(text = "Oh no! I dont know that type of snake...
Please try again with another type of snake.")
    continue
break
fancytext(text = "Who is the owner of the snake?")
snakeOwner = input()
s = Snake(snakeName, snakeType, snakeOwner)
fw = open("Snakes.txt", "a")
fw.write("\nSnake id: " + str(s.getSnakeId()) + ". Snake name: " +
s.getSnakeName() + ". Snake type: " + s.getSnakeType() + ". Owner of the
snake: " + s.getOwner() + ".")
fw.close()
fancytext(text = "Snake added successfully")
while True:
    again = input("Add more snakes? Y/N: ").lower()
    if (again == "y"):
        addSnake()
    elif (again == "n"):
        break
    else:
        fancytext(text = "Enter Y or N")
        continue
break

```

## Program text

In this section I tell the program what text to show when starting off the program. What to write as well as sentences (what strings to show).

I then start a while loop to begin the program after a welcome sentence. I have used my fancytext file for the text I have to print since I already inside that python page have told the program that I will use that to create the strings I wanna print and how to showcase it.

In a string I tell the program that the user should press 0-4 to choose the thing they wanna do. In the while loop I tell the program that a new variable snake\_or\_owner = an input as well as remember to tell the program that the user is allowed to use numbers, hence using int to convert it to using numbers as well. If the user chooses 1, then the program will run the add new owner code I just made, and if they choose 2 then the add new snake code will run.



I then continue to give the other numbers values, so if they choose 3, then it will show (open) the list of owners from the owners.txt file and if they choose 4 then it will show the list of added snakes from the snakes.txt file. The last one is 0 which tells the program to exit the program, or if any other number is written it will ask you to try again. In most of the string codes I use \n which tells the program to create a new line for that sentence.

```
# program

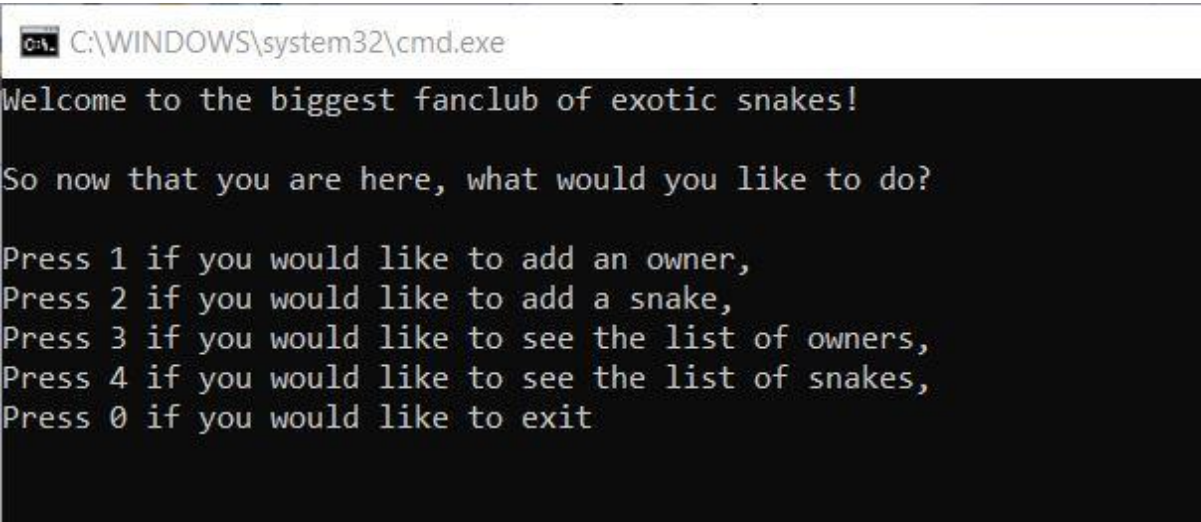
fancytext(text = "Welcome to the biggest fanclub of exotic snakes!")
while True:
    fancytext(text = "So now that you are here, what would you like to do?")
    fancytext(text = "Press 1 if you would like to add an owner, \nPress 2 if you would like to add a snake, \nPress 3 if you would like to see the list of owners, \nPress 4 if you would like to see the list of snakes, \nPress 0 if you would like to exit")
    while True:
        snake_or_owner = input()
        try:
            snake_or_owner = int(snake_or_owner)
        except:
            fancytext(text = "Sorry I didn't get that. You have to enter a number between 0 and 4 to make a choice.")
            continue
        if (snake_or_owner == 1 ):
            fancytext(text = "You chose: Add an owner")
            addOwner()
        elif (snake_or_owner == 2):
            fancytext(text = "You chose: Add a snake")
            addSnake()
        elif (snake_or_owner == 3):
            fancytext(text = "You chose: See list of owners")
            f = open("Owners.txt", "r")
            fancytext(text = f.read())
            fancytext(text = "Allright! That was an interesting list! What's next?")
        elif (snake_or_owner == 4):
            fancytext(text = "You chose: See list of snakes")
            f = open("Snakes.txt", "r")
            fancytext(text = f.read())
            fancytext(text = "Allright! That was an interesting list! What's next?")
        elif (snake_or_owner == 0):
            exit(0)
        else:
```

```
fancytext(text = "You can only enter the numbers 0-4. Try  
again.")  
    continue  
    break  
    continue
```

## Finished looks

### Start page

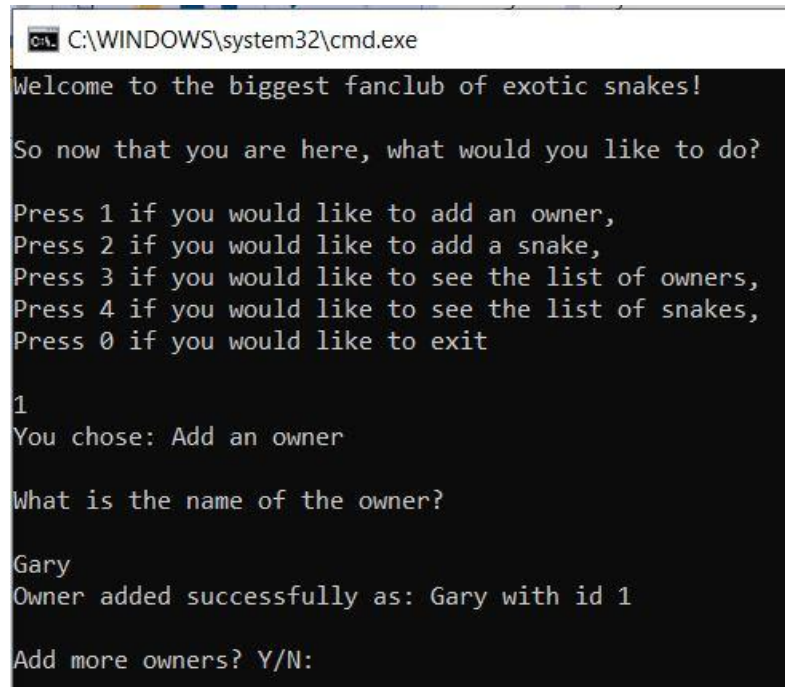
This is how the page looks before an answer is chosen. I will go through the program one by one to show you how it looks when being tested.



```
C:\WINDOWS\system32\cmd.exe  
Welcome to the biggest fanclub of exotic snakes!  
  
So now that you are here, what would you like to do?  
  
Press 1 if you would like to add an owner,  
Press 2 if you would like to add a snake,  
Press 3 if you would like to see the list of owners,  
Press 4 if you would like to see the list of snakes,  
Press 0 if you would like to exit
```

## Choose 1

Choosing to create a new owner will look like this inside the program. Since the program is asking for a name anything you write will be seen as a name so they will add the new owner pretty much no questions asked.



```
C:\WINDOWS\system32\cmd.exe
Welcome to the biggest fanclub of exotic snakes!

So now that you are here, what would you like to do?

Press 1 if you would like to add an owner,
Press 2 if you would like to add a snake,
Press 3 if you would like to see the list of owners,
Press 4 if you would like to see the list of snakes,
Press 0 if you would like to exit

1
You chose: Add an owner

What is the name of the owner?

Gary
Owner added successfully as: Gary with id 1

Add more owners? Y/N:
```

## Choose 2

Choosing the second option lets the user add a new snake, where they have to input the name, type of snake and owner of the snake. I will show you the two examples of what happens when you choose the mamba type, and what happens when you choose the anaconda type.

```

C:\WINDOWS\system32\cmd.exe

Press 1 if you would like to add an owner,
Press 2 if you would like to add a snake,
Press 3 if you would like to see the list of owners,
Press 4 if you would like to see the list of snakes,
Press 0 if you would like to exit

2
You chose: Add a snake

What is the name of the snake?

Loopy
What type of snake is it? "Mamba" or "Anaconda"?

anaconda
Oh no! Bad news! This type of snake is protected, and cannot be added!

What type of snake is it? "Mamba" or "Anaconda"?

mamba
Good choice! Though remember this snake is toxic! so be careful!

Who is the owner of the snake?

Gary
Snake added successfully

Add more snakes? Y/N:

```

### Choose 3

When 3 is chosen it will show the user the list of already existing owners. If the program is used for the first time there will be no owners on the list, and the owners.txt is empty. however since I already created a few owners testing the program it will have a few names.

```

3
You chose: See list of owners

Owner id: 1. Owner name: John.
Owner id: 2. Owner name: Holland.
Owner id: 1. Owner name: Gary.

Alright! That was an interesting list! What's next?

```

## Choose 4

Choosing 4 will then show the list of snakes added to the snakes.txt file, the already existing one, the same form of rules applies to this list as well as the owners list.

```
4
You chose: See list of snakes

Snake id: 1. Snake name: Hally. Snake type: mamba. Owner of the snake: John.
Snake id: 2. Snake name: Bruno. Snake type: mamba. Owner of the snake: John.
Snake id: 1. Snake name: Loopy. Snake type: mamba. Owner of the snake: Gary.

Allright! That was an interesting list! What's next?
```

## Choose 0

Choosing 0 lets the user exit the program.

```
So now that you are here, what would you like to do?

Press 1 if you would like to add an owner,
Press 2 if you would like to add a snake,
Press 3 if you would like to see the list of owners,
Press 4 if you would like to see the list of snakes,
Press 0 if you would like to exit

0
Press any key to continue . . .
```

## Finishing touches

As you can see it still has a few issues mainly the id issue. I have tried to fix it but it continues to not automatically go one up as wished for when creating a new owner or snake. Meaning that they all have the index number of 1, however you can manually change it inside the txt files. For now I will let that mistake be acceptable since I had a hard time with the rest of the code and it is a small error.