

Programming

Lab Sessions

Assignment IV - Presentations

- Zoom room for presentations:
Meeting ID: 641 7002 9853
Passcode: 034035
- Group order for presentation available at marcomarinho.com
- **Be on time**
- Martin will be available to solve doubts at the usual meeting room with an exercise list.

Resources

- Slides for this lecture and the solutions for the third exercise list are available at marcomarinho.com.
- Videos for Martins solutions of the exercises are up.
- Assignment 5 is available.

DET ÄR MATEMATIKFRI FREDAG MINA BEKANTA



Tuples and Strings

- Tuples are a sequence of values of **any type that cannot be changed once they are created.**
- Values of a tuple are accessed just like with lists.
- Strings are treated as tuples in python. Therefore, strings are immutable.

Example

- When to use a Tuple? When the set of values to be stored in the program is not supposed to be changed (by the user or another programmer).

Example

#The white keys in a piano can only have one of these notes

```
white_keys = ('C', 'D', 'E', 'F', 'G', 'A', 'B')
```

#Accessing elements with index

```
for i in range(len(white_keys)):
    print(white_keys[i])
```

#Accessing the elements with direct iteration

```
for key in white_keys:
    print(key)
```


Burger Shop



Example

- Let us write a program where a client can input the ingredients for his burgers, separated by a comma, we will then store the ingredients separately in a list.

Example

#Input from the user

```
ingredients = input('Please enter the ingredients, separate each ingredient by a comma: ')
```

#Split the ingredient string into a list using the comma as the indicator

```
ingredient_list = ingredients.split(',')
```

#Clears the strings from each ingredient, removing spaces in front and after

```
for i in range(len(ingredient_list)):
```

```
    ingredient_list[i] = ingredient_list[i].strip()
```

#Print the resulting list

```
print(ingredient_list)
```

Example

- Now let's do the inverse, from a list of ingredients, let's print it in a readable way, separated by commas.

Example

#Input from the user

```
ingredients = ['burger', 'mayo', 'cheese', 'egg', 'bacon']
```

#initialize result string

```
result = ""
```

#Concatenate all the ingredients to a single string

for ingredient **in** ingredients:

```
    result = result + ingredient + ','
```

#Remove the last comma from the string

```
result = result[0:-2]
```

#Print the result

```
print(result)
```

Example

- Now, lets read a burger menu from a file. Our program will let the client find the burgers that contain a certain ingredient that he wants to have.

Example

```
#Read the data from the file
source = open('burgers.txt')
data = source.readlines()

#Clean the data read from the file
data_clean = []
for line in data:
    data_clean.append(line.strip())

#Initialize list with names of burgers and the ingredients
names = []
ingredient_lists = []

#Add ingredients and names to the right list
for line in data_clean:
    #If line is empty, do nothing
    if line == '':
        continue
    #If list has a comma, add it to ingredients
    elif ',' in line:
        ingredient_lists.append(line)
    #Else, add it to names
    else:
        names.append(line)

#Get the ingredient the client wants
client_ingredient = input("What ingredient would you like to have: ")

#Check which lists of ingredients have the ingredient specified
positive_results = []
for i in range(len(ingredient_lists)):
    if client_ingredient in ingredient_lists[i]:
        positive_results.append(i)

#Print the name of the burgers based with the ingredient
print("The burgers that have this ingredient are:")
for result in positive_results:
    print(names[result])
```

Example

- Now, lets write a program that creates the gigaburger, it has all the ingredients that exist on our menu, but also has 10 times more of it



Example

```
#Read the data from the file
source = open('burgers.txt')
data = source.readlines()

#Clean the data read from the file
data_clean = []
for line in data:
    data_clean.append(line.strip())

#Initialize list with names of burgers and the ingredients
names = []
ingredient_lists = []

#Add ingredients and names to the right list
for line in data_clean:
    #If line is empty, do nothing
    if line == '':
        continue
    #If list has a comma, add it to ingredients
    elif ',' in line:
        ingredient_lists.append(line)
    #Else, add it to names
    else:
        names.append(line)

#Initialize the string for the ingredients in the gigaburger
giga_ingredients = ""

#Loop for each list of ingredients
for lista in ingredient_lists:
    #Split the list of ingredients and clean it
    sublist = lista.split(',')
    for index in range(len(sublist)):
        sublist[index] = sublist[index].strip()
    #If the ingredient is not in the gigaburger, add it 10 times
    for element in sublist:
        if element not in giga_ingredients:
            giga_ingredients = giga_ingredients + '10 X ' + element + ','

#Print the result
print(giga_ingredients[:-2])
```

Example

- Now, lets update the menu file prices for the burgers. Once this is done, let's create a simple interface for a user to order our burgers and get the final price for his order.

Example

```
#Read the data from the file
source = open('burgers.txt')
data = source.readlines()

#Clean the data read from the file
data_clean = []
for line in data:
    data_clean.append(line.strip())

#Initialize list with names of burgers and the ingredients
names = []
ingredient_lists = []
prices = []

#Add ingredients and names to the right list
for line in data_clean:
    #If line is empty, do nothing
    if line == "":
        continue
    #If list has a comma, add it to ingredients
    elif ',' in line:
        ingredient_lists.append(line)
    #Else, add it to names
    else:
        names.append(line.split(' - ')[0])
        prices.append(line.split(' - ')[1].replace(' SEK', ''))

#Initialize total price for the order and the amount ordered of each burger
total_price = 0
order = [0]*len(names)
```

Example

while True:

```
#Print the selection of burgers
print('Please, choose a burger, 0 to quit')
for i in range(len(names)):
    print(i+1, names[i])
selection = int(input('==>'))
if selection == 0:
    break
#Add the price of the selected burger to total and update the order
else:
    total_price = total_price + int(prices[selection-1])
    order[selection-1] = order[selection-1]+1
#Print the current order
print('Your order so far')
for i in range(len(names)):
    print(order[i], names[i])
print("")

#Print the order total
print('Your total is', total_price)
```

Exercise

- Try to write a program that can build a simple menu in a text file. The program asks the user for the name of the burger, the price, and the ingredients one by one. In the end the result should look something like this:

Marco Burger - 200

burger, cheese, onions, garlic, lettuce, mayo

Exercise

```
name = input('Please enter the name of the burger: ')
price = input('Please enter the price of the burger: ')

ingredients = []
while True:
    choice = input('Please type in an ingredient')
    if choice == 'end':
        break
    else:
        ingredients.append(choice)

single_list = ''
for ingredient in ingredients:
    single_list = single_list + ingredient + ', '

source = open('custom.txt', 'w')
source.write(name + ' - ' + price + '\n')
source.write(single_list[:-2])
source.close()
```

End