Basic Programming - Seminar 2

Alexandra Ciobotaru

5 November 2021

The scope of this seminar is to get familiar with strings and lists in Python and get a sense on how they are used when programming. In this project we will exercise string concatenation, string formatting and working with lists: indexing, appending and extending them.

1 Create a new PyCharm project

Create a new PyCharm project and name it intuitively, something like "Seminar2". In the venv folder on the left, right click and create a new script. Name it intuitively as well, maybe "get_user_info_from_keyboard.py", because we are going to create a script where the user inserts some information from keyboard and Python displays it nicely on the screen, in a formatted way.

1.1 Create the skeleton of the project

Write comments in your script to know what you are going to do:

```
# Create empty list to store data

# Ask user for first name

# Ask user for last name

# Append these values to the created empty list

# Ask user for age and append it to list

# Ask user for city and append it to list

# Ask user what they enjoy and append it to list

# Create output text

# Print output to screen
```

1.2 Start filling in the skeleton with variable assignments

Under the first comment, create an empty list that you will be using to store the information you get from user:

```
user_info = []
```

Under the second comment, get the name from user keyboard and put it into variables to use later:

```
first_name = input("What is your first name?: ")
print(first_name)
last_name = input("What is your last name?: ")
print(last_name)
```

Then put those into the `user_info` list:

```
user_info.extend([first_name, second_name])
print(user_info)
```

Similarly, under the third comment, get the age from user keyboard and put it into a variable:

```
age = input("What is your age?: ")
print(age)
```

Append the variable to the `user_info` list (fill in the blanks):

```
user_info. ...(...)
```

Similarly, create a variable called 'city' to store the city the user lives in, a variable called 'love' to store what the user enjoys doing, and append those to the `user_info` list as well. How many elements do you have in your information storing list? Use the `len` method to find out.

1.3 Use string formatting to print the output to screen

We will pretend that the output is a response of the computer to the user.

```
string = "Your name is {} and you are {} years old.
          Now I know that you live in {} and you love {}"
output = string.format(user_info[0] + " " + user_info[1],
         user_info[2], user_info[3], user_info[4])
print(output)
```

Sure, you could also have used variable names instead of indexing the list:

```
output = string.format(first_name + " " + last_name, age,
         city, love)
print(output)
```

2 Create an email slicer

Create a new script and name it `email_slicer.py`. In this project we will ask the user for their e-mail address using the input function. After that, we're going to slice the e-mail address for the user name and the email domain name, being that the username is what it is before the @ symbol and the domain what if after the @ symbol.

2.1 Create an outline of the project

Write some comments as the skeleton of your project:

```
# get user email address from keyboard

# slice out user name

# slice domain name

# display output message
```

2.2 Start filling in the outline

Get the user email from keyboard using the input function and assign it to the variable email. We will use the strip() method to remove any accidental spaces inserted from the keyboard because, as you saw last seminar, python gives an error when that happens.

```
email = input("What is your email address?: ").strip()
```

The user will input something like: johndoe@yahoo.com. Up until the '@' symbol we have our **username**, and everything from the '@' symbol until the dot before the end of the string would be the **domain**.

Using the index function create a slice that contains the username and save it in a variable called 'user' (fill in the blanks).

```
user = email[ : email.index(...)]
```

Similarly, use the index function to create a slice that contains the domain and save it in a variable called 'domain' (fill in the blanks). Pay attention to slice up until the dot before the end of the string.

```
domain = email[....]
```

Use the capitalize method to write the domain name with capital letter.

```
domain = domain.capitalize()
```

Create the output message using the format method - put in the first place the username and in the second place the domain name (fill in the blanks). Notice how we escaped the quotes using backslash.

```
output = "Your username is \'{}\' and your domain name is {}". .....
```

Print the output message.

```
print(output)
```

Congratulations, you have finished your second seminar!