Python Beginner Labs

By: Ömer Fırat Bekiroğlu

Lab 6 – Functions

Goals

- - Understand how to define and call functions
- - Use parameters and return values
- - Learn about scope (local vs. global variables)

Concept Brief

Functions let you organize and reuse code. They're like named blocks that take input and return output.
Example:

```
def greet(name):
  return "Hello " + name
```

Tasks

1. - Write a function greet_user(name) that returns 'Hello, <name>!' and test it with 2 names.
2. - Write a function add(a, b) that returns their sum. Try with different values.
3. - Write a function is_even(n) that returns True if n is even, else False.
4. - Write a function convert_to_seconds(hours, minutes) that returns the total number of seconds.

Stretch Task

Write a function calculate_tip(amount, percentage) that returns the total amount after tip. Then ask the user for bill and tip and print final total using this function.

Reflection

- - What happens if you forget to return?
- - How are functions helpful when writing large programs?

Lab 7 – Dictionaries

Goals

- - Create and access dictionaries
- - Loop through keys and values
- - Understand .get(), .update(), .pop()

Concept Brief

Dictionaries store key-value pairs. Think of them like address books:
Example:
contacts = {'Alice': '123', 'Bob': '456'}
print(contacts['Alice'])

Tasks

5. - Create a dictionary with 3 friends and their phone numbers.
6. - Add one more friend. Then update an existing number.
7. - Loop through the dictionary and print each friend's name and number.
8. - Ask the user for a name. If it exists in your dictionary, print the number. If not, say 'Not found'.

Stretch Task

Create a student grades dictionary. Ask user for 3 student names and 3 grades, store them, then print average grade and names of students who passed (>= 60).

Reflection

- - What happens if you access a key that doesn't exist?
- - Why is .get() safer than []?

Lab 8 – Nested Structures & Logic

Goals

- - Understand how to nest lists/dictionaries
- - Combine control structures with data structures

Concept Brief

You can store dictionaries inside lists, or vice versa.
Example:
students = [{'name': 'Ali', 'grade': 85}, {'name': 'Ayşe', 'grade': 72}]

Tasks

9. - Create a list of 3 dictionaries. Each dictionary has a 'name' and 'score' key.
10. - Loop through the list and print: 'Name: Ayşe, Score: 72' format.
11. - Find and print the student with the highest score.

Stretch Task

Ask user to input data for 3 students: Store their name and scores in a list of dictionaries. Then print class average and all students above average.

Reflection

- - Did you get stuck with [ ] vs { }?

- - How do you access a value inside a dictionary inside a list?

Lab 9 – File I/O

Goals

- - Open, read, and write text files
- - Handle simple line-by-line input/output

Concept Brief

Python lets you read/write files easily:
Example:

```
with open('notes.txt', 'w') as f:
  f.write('Hello!\n')
```

Tasks

12. - Create a file called my_notes.txt and write 3 lines of text into it.
13. - Read the file back and print each line.
14. - Ask the user to input a sentence and append it to the file.
15. - Count how many lines the file has.

Stretch Task

Create a journal app: Each time it's run, it asks for a journal entry. It saves the date and entry to a file. When it starts, it prints all past entries. (Hint: use datetime module)

Reflection

- - Did you forget to close a file? (Why is with open(...) better?)
- - What kind of programs use file storage like this?

Lab 10 – Mini Project: Quiz App

Goals

- - Combine all previous topics into a small app
- - Practice functions, loops, input, and dictionaries

Concept Brief

The app should ask 5 multiple-choice questions. Store questions and correct answers in a dictionary or list. Keep score and show it at the end.

Suggested structure:

```
questions = [
 {'question': 'What is 2 + 2?', 'options': ['2', '4', '6', '8'], 'answer': '4'}
]
```

Tasks

16. - Create a list of 5 question dictionaries with question, options, and answer.
17. - Loop through each question: Show question and options, get user input, tell if correct or not.
18. - Keep a score variable and print final score.

Stretch Task

Add difficulty levels: Easy/Medium/Hard (user selects). Record the score into a file with date.

Reflection

- - Which part did you reuse from earlier labs?
- - What did you struggle to organize?