Lab 1: Python Basics

Practices in Visual Computing 1, Fall 2023

Subjects

1. Intro and Logistics
2. Basic Python
3. Intermediate Python
4. Basic Numpy

TAs

Mehdi Safaee (MSc in CS)

Responsible for Labs

mehdi_safaee@sfu.ca

Aryan Mikaeili (MSc in CS)

Responsible for Assignments

aryan_mikaeili@sfu.ca

Assignments

1. Classical image processing
2. Deep learning
3. Geometric processing
4. Games/simulations

Tips for the course

- Preferably, use VSCode (or other **IDEs**) for projects and **Notebooks** for snippets
- Use `conda` to manage environments and packages
- Avoid using ChatGPT for assignments. But definitely use it as a learning tool!
  - You may be asked to write code snippets during project demos
- Avoid loops whenever possible
- Stick to Python best practices,
  - Proper namings, PEP8, Docstrings, etc
  - Try to seek optimal yet clean and easy-to-understand solutions

Subjects

1. Intro and Logistics
2. Basic Python
3. Intermediate Python
4. Basic Numpy

https://github.com/Mehdi0xC/visual-computing-1

Question #1

Basic Arithmetic and Functions

- Write a Python function that calculates the volume and surface area of a sphere given its radius and prints it in a clean format.
  - Approximate π with 3.14
  - Round the values to two decimal points

Question #2

Loops

- Write a Python code that approximates π using Euler's method
  - Don't use `Numpy` or `Math` library
  - Approximate with 1 million iterations

$$1/1^2 + 1/2^2 + 1/3^2 + 1/4^2 + \ldots = \pi^2/6$$

Question #3

Lists

- Write a Python function that gets two lists, one called weights and second called values, as input, and then calculate weighted sum. Use random module for weights and values.
  - Calculate for lists of 1 million elements
  - Use `random.uniform(0, 1)`
  - Justify the answers
- Advanced tips
  - Redo with list comprehension
  - Experiment with seed

Subjects

1. Intro and Logistics
2. Basic Python
3. Intermediate Python
4. Basic Numpy

Question #4

Dictionaries

You are given a dictionary, containing student IDs as keys and a second dictionary as values. the second dictionary contains grades for each assignment in a key-value format: `"a1": 95`

- Write a function that accepts this dictionary and adds "**avg**" to each students grades.

Question #5

Files and Strings

- For the previous question, construct the input dictionary from a .csv file with this data:

```
std_id, a1, a2, a3, a4
std1, 75, 80, 95, 90
std2, ...
...
```

- After calculating the averages, write them into a new .csv file:

```
std_id, a1, a2, a3, a4, avg
std1, 75, 80, 95, 90, 85
std2, ...
...
```

- Also try directly saving in .json format

Question #6

Object Oriented Programming

- Create a class student that gets std_id, name, age, gender, and major as constructor.
- It should also create a list of dictionaries, called courses.
- A method add_course will receive a course_name: grade key-value pair and will append it to courses list
- A method get_gpa will return students GPA by calculating average for all courses.
- A method export_info will create a txt file containing human readable information about the student.
    - Advanced tip: use __str__ for custom string representation of the class

Subjects

1. Intro and Logistics
2. Basic Python
3. Intermediate Python
4. Basic Numpy

Question #7

Create the following two ndarrays

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{pmatrix}$$

And calculate the following:

$$AB \qquad BA$$

Question #8

First create a random 10x10 matrix, then using array slicing, change it to the following format:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & x & x & x & x & x & x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Question #9

Redo the question three, this time using numpy arrays. Use time module to compare the speedup between list and ndarrays.