# Lab Exercise 1: Data Science Ecosystem Installation, Integrated Development Environments, and Code Standards in Python (PEP-8)

Humberto Díaz

Artificial Intelligence Post-Degree Diploma

The University of Winnipeg - PACE

Foundations in Data Science

Muhammad Shahin PhD.

March 14, 2025

Learning Objectives

- Set up your Python programming environment

* Install Anaconda Python distribution
* Install Visual Studio Code
* Python extension for Visual Studio Code

- Create your first python script

- Learn how to apply key components of the PEP-8 style guide to your code

## Environments Already Installed

1. Visual Studio Code

A screenshot of a computer

AI-generated content may be incorrect.

The picture above confirms the correct installation of Visual Studio Code

1. Anaconda Navigator and Jupyter Notebook

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

The pictures above confirm the installations of Anaconda Navigator and within it, Jupyter Notebook

1. Python IDLE

A screenshot of a computer

AI-generated content may be incorrect.

The picture above confirms the correct installation of the Python IDLE

## First python script using IDLE, Visual Studio Code, and Jupyter notebook

Variables for the year, month, and day of my birth were created. Moreover, the mean and sum of those variables were calculated without using any external packages, resulting in the following script properly written according to PEP-8.

# Lab Exercise 1

# 1. Define birth date variables.

year = 1998

month = 10

day = 14

# 2. Calculate sum and mean of the birth date.

sum\_of\_dates = year + month + day

mean\_of\_dates = sum\_of\_dates / 3

# 3. Define name and student number as variables.

name = "Humberto Diaz"

student\_number = 3177986

# 4. Print the outcome.

print(f"Name: {name}")

print(f"Student Number: {student\_number}")

print(f"Summatory of dates: {sum\_of\_dates}")

print(f"Mean of dates: {mean\_of\_dates}")

This script prints the calculation results: my name, and my student number in the following environments:

1. Script executed in Visual Studio Code

A screenshot of a computer program

AI-generated content may be incorrect.

1. Script executed in Jupyter Notebook

A screenshot of a computer program

AI-generated content may be incorrect.

1. Script executed in IDLE

A screenshot of a computer

AI-generated content may be incorrect.

## Key Differences Between Python 2 and Python 3

The following points highlight some changes in Python.

1. **Print Statement vs. Print Function**

In Python 2: print "Hello, world" (parentheses are not required).

In Python 3: print("Hello, world") (parentheses are mandatory).

1. **Integer Division**

In Python 2: 5 / 2 returns 2 (integer division by default).

In Python 3: 5 / 2 returns 2.5 (true division by default).

1. **Unicode Support**

In Python 2, strings are ASCII by default (str).

In Python 3, strings are Unicode by default (str).

1. **input() Function**

In Python 2, input() evaluates the input as Python code, and raw\_input() is used for plain text input.

In Python 3, input() always returns a string.

1. **Iterators in range()**

In Python 2, range() returns a list, and xrange() returns a generator.

In Python 3, range() is a generator by default (equivalent to xrange()).

1. **Compatibility and Support**

Python 2 is no longer officially supported since January 2020.

Python 3 is the recommended and actively developed version.