Lab 1 – Introduction to Python

Script Languages

Aims

1. Starting Python Interpreter
2. Running Simple Applications
3. Installing Additional Python Modules
4. Managing Dependencies

Introduction

Python is a scripting language, and its programs are executed by an interpreter. In simplified terms, the basic way to run applications written in this language is to pass the path to the script as an argument to the interpreter, which is responsible for executing commands "line by line". At any time, you can edit the script and run the program again without the need for rebuilding (recompiling, linking, etc.). Because of this, scripting languages are good tools for tasks such as prototyping, task automation, data processing, etc.

Assignments

1. Running the interpreter in the REPL mode.

REPL (Read-Eval-Print Loop) mode allows running the interpreter in shell mode. Once started, the user enters expressions for evaluation (after the prompt symbol `>>>`), which are then evaluated by the interpreter and the result of the evaluation is displayed.

a. To check the version of the Python interpreter available in the current path, run the console/command prompt and then type: `python --version`.
b. Launch the Python Interpreter in REPL mode and construct a program that prints the string "Hello world!".

```
python
```

```
>>> print(“Hello world!”)
```

c. Use a variable to modify the message.

```
>>> name = “Python”
```

```
>>> print(“Hello, {0}!”.format(name))
```

```
>>> print(f“Hello, {name}!”)
```

d. Perform some arithmetic operations by constructing expressions.

   i. addition (`+`), subtraction (`-`), multiplication (`*`),
   ii. division (`/`), integer division (`//`), remainder (`%`),
   iii. exponentiation (`**`).

e. Next, construct several arithmetic expressions using the `_` symbol as one of the operands, e.g.

```
>>> _ * 2.0
```

```
>>> _ + 1
```

Zastanów się, jaką wartość reprezentuje ten symbol.

f. End the interactive session by calling the expression `exit()`.

2. Running Jupyter notebooks.

Jupyter Notebook - a graphical tool that runs in a web browser and allows mixing of text with Python scripts, enabling the creation of documents with computable elements (e.g. visualizations).

a. Install Jupyter Notebook by typing in the terminal/command prompt.

```
pip install notebook
```

b. Launch Jupyter Notebook.

```
jupyter notebook
```

c. Kliknij przycisk listy rozwijanej 

n the right-hand side and select the "Python 3 (ipykernel)" option.

d. Enter the following code into the text field next to the `In [ ]:`.

```
name = "World"

print(f"Hello, {name}")
```

Press the "Run" button to execute the code.

e. In the next text field, type

```
# My Python application

This is my first Jupyter notebook.
```

Using the drop-down list located under the menu, change the block type from Code to Markdown. Press the "Run" button.

f. Press the save button  . Analyze the contents of the *.ipynb file in a text editor.

3. Running scripts in the console.

a. Open your preferred IDE (e.g. VSCode).
b. Create a file named app.py with the following contents:

```
name = input("Enter your name: ")

print(f'Hello, {name}!')
```

c. Run the file from the terminal/command line using the command:
`python app.py`

4. Simple Python program

Write a program in Python that calculates the area of a triangle based on the height and base length entered by the user. To convert between a string and a number, you can use the `int()` function.

5. Using dependency manager

In addition to its extensive standard library, Python has a rich repository of modules and tools: The Python Package Index (PyPI).

Modules can be installed from the terminal/command line using the pip command, as was done in previous tasks. After installing the module, it should be imported into the script using the `import` statement. For example, for the wikipedia module, which allows for content to be downloaded from Wikipedia pages:

```
pip install wikipedia

…

>>> import wikipedia

>>> wikipedia.page("Script language").summary
```

While working on larger projects, it is useful to use a dedicated tool to manage modules and their versions. There are several options to choose from:

- Venv + pip https://docs.python.org/3/library/venv.html
- Poetry https://python-poetry.org
- Conda https://conda.io/en/latest/
- Pipenv https://pipenv.pypa.io/en/latest/

Please familiarize yourself with the documentation of the chosen tool. Then, using it, create a program that depends on the wikipedia module. Let the program accept the name of an article on Wikipedia from the user and output its summary and URL to the page on the English Wikipedia.