Level 1

Chapter 1

Introduction

Sudip Ghimire

https://sudipghimire.com.np

Image, Trademark, Docs: https://python.org

Template design: **Jimena Catalina** (https://www.slidescarnival.com/salerio-free-presentation-template/1875#preview)

HELLO!

I’m Sudip Ghimire
I’ve been working as a full-stack software engineer and a system architect since 2017

What is Python?

- An interpreted, high level, general purpose programming language
- Created by Guido Van Rossum, First released in 1991
- Can be used in
  - Software and Web Development
  - Mathematics, Data Science and Artificial Intelligence
  - Games and Simulation
  - Scripting

An Interpreter

- A computer program that directly executes instructions without requiring to be compiled to a machine level code.
- The code executes line-by-line when required
- Unlike compiler, it runs instructions until the line where error is spotted

An Interpreter (Continued...)

- It does not generate intermediary code, hence, is highly efficient in terms of memory
  - Drawback: Needs to be interpreted every time
- Programming languages such as Perl, Python, MATLAB, QBasic

Hello World Python Vs Other

Java

```java
class Main{
  public static void main(String args[]){
    System.out.println("Hello World");
  }
}
```

Python

```python
print('Hello World')
```

Why Python?

- Compatible with all major operating systems
- Larger community support
- Easier Syntax, cleaner code, shorter instructions
- Can be used in both procedural and Object-Oriented approach
- Easier to develop, maintain and support

Installing Python

- Windows
  - Browse https://python.org and download the latest version or the version of your choice
  - Install Python for one or all users (do not forget to check “add python to path” while installing)
- Linux and Mac OS comes with python pre-installed so we generally do not need to install python unless specific version of python is required.

Python Syntax

- Unlike other programming languages, python do not need to end the statement with semicolons e.g. `x = 5`
- When we insert a new line, the python automatically identifies whether the statement is complete or not

```
1. x = 5            # the statement is terminated here
2. person = {       # the statement is not terminated here
3.     "name": "John"
4. }    # the statement will now be terminated with new line
```

Python Syntax (Continued...)

- **Indentation** is the most important part of python syntax
  - We use indentation to create a code block. E.g. function, classes, loops, etc.
  - The standard python indentation size is 4 spaces but we can use any number of spaces or tabs
  - If a block of code consists different number of indentation, then the program would not run

Python Syntax (Indentation example)

```
1. x = 5
2. for value in range(x):
3.     print('inside the loop')  # indentation
4.     print(value)                                  #
   end of the block
5. print('loop has been ended')  # de-indentation
```

Python Syntax (Continued...)

- Python variables are dynamically typed
  - We do not need to specify type of the variable. E.g. `x = 5`
  - But from Python 3.6 Python has added feature of Type Hinting. E.g. `x: int = 5`
- Single line comments are added by adding `# in front of the line`
- Multiline comments can be added by enclosing statements within triple quotes `'''` or `"""`.

Python Shell

- We can access python shell by either of ways
- Open an idle python editor that installs along with python
- On CMD/Terminal/PowerShell, you can type **`python`** and enter
- In Windows, we can access the shell with even shorter command, **`py`**
- Once the shell is active, we can see the shell starting with symbol **`>>>`**

Python Editor and IDE

- The default python editor, Idle offers basic syntax highlighting and debugging capability
- When we start coding more than one file, then we use different Integrated Development Environments also known as IDE
- Using IDE helps us working with multiple files, workspace, and debugging files
- IDE speeds up development workflow by auto importing, auto completing, code refactoring, etc.

Python Editor and IDE (Continued...)

- Commonly used IDE are
  - Visual Studio Code (VS Code)
  - PyCharm Community / Professional Edition
  - Spyder (Generally preferred for Data Science)
  - Jupyter Notebook (Generally preferred for AI/ML/Data Science)

Python Hello World Program

- Refer to Lab 1