# PYTHON-BASIC TO INTERMEDIATE

Learn with N@ima

# Day 1

## Introduction to Python

## Programming Language

A **programming language** is a system of notation for writing computer programs

A **computer program** is a sequence or set of instructions in a programming language for a computer to execute.

Python is an **interpreted** and **object-oriented high-level** programming language. It has dynamic semantics.

**History of Python:** The Python programming language was introduced by Guido Van Rossum in 1989, and the first version of Python was released in 1991.

In 1994, Python 1.0 was released with new features like map, filter, lambda. There were further updates in the language, and Python2.X and Python3.X were also released.

## Compiler vs Interpreter

Compiler and Interpreter are two different ways to translate a program from programming or scripting language to machine language.

- A **compiler** takes entire program and converts it into object code which is typically stored in a file. The object code is also referred as binary code and can be directly executed by the machine after linking. Examples of compiled programming languages are C and C++
- An **Interpreter** directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code. Examples of interpreted languages are **Perl, Python and MATLAB**.

## What is an interpreted language in programming?

An interpreted language is a programming language that is generally interpreted, without compiling a program into machine instructions.

It is one where the instructions are not directly executed by the target machine, but instead, read and executed by some other program.

# Object Oriented Programming

Object-oriented programming is based on the concept of objects.

In object-oriented programming data structures, or objects are defined, each with its own properties or attributes.

Each object can also contain its own procedures or methods.

Example: A car has a model name, a colour, a year in which it was manufactured, an engine size and so on. We would therefore create a Car object with the name, colour, engine size and year as attributes. For every new car we use, we would use the car object. For instance, we can have a 2019 Blue BMW or a 2017 Red Audi. In each instance we would reuse the code contained in the original object.

## High and Low level language

A high-level language is one that is user-oriented in that it has been designed to make it straightforward for a programmer to convert an algorithm into program code.

A low-level language is machine-oriented. Low-level programs are expressed in terms of the machine operations that must be performed to carry out a task.

## Why Python?

Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).

Python has a simple syntax similar to the English language.

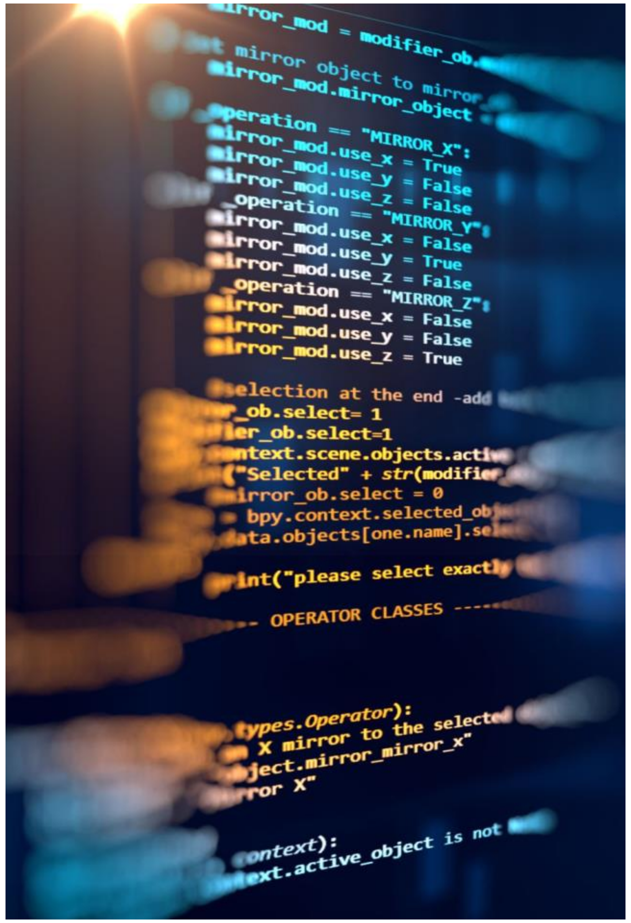
Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Python can be treated in a procedural way, an object-oriented way or a functional way.

## Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.



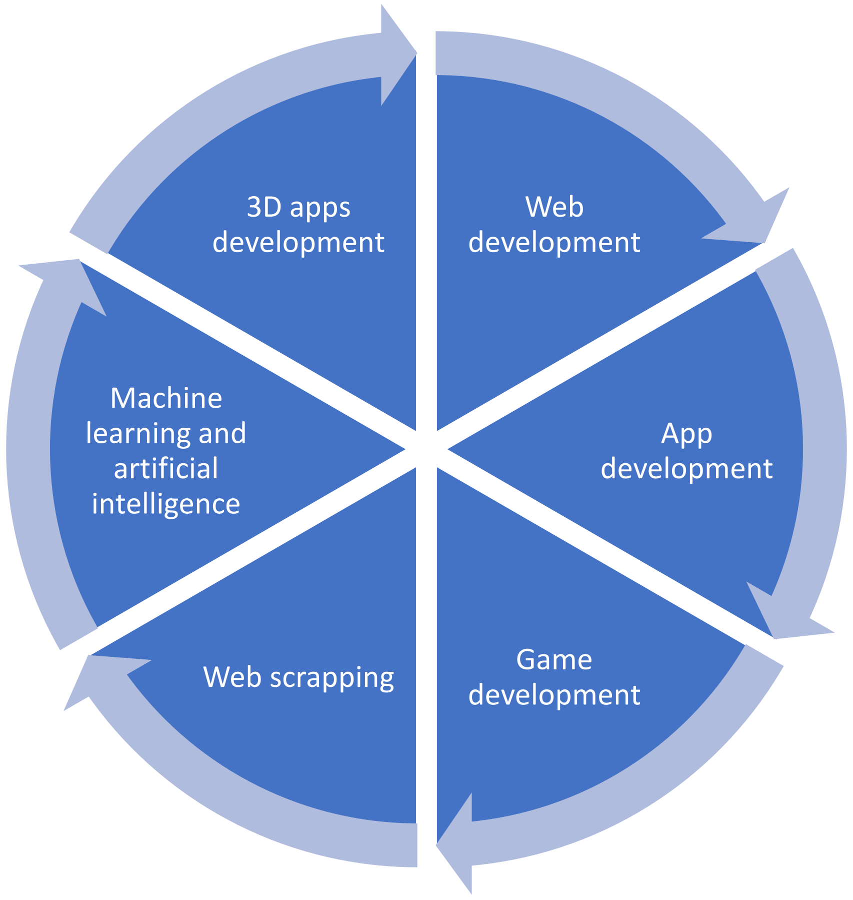
```
irror_mod = modifier_ob.
 mirror object to mirror
irror_mod.mirror_object =
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

election at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.activ
("Selected" + str(modifier
irror_ob.select = 0
 bpy.context.selected_obj
ata.objects[one.name].sel

int("please select exactl

- OPERATOR CLASSES ----

types.Operator):
 X mirror to the selected
ject.mirror_mirror_x"
ror X"

ontext):
ext.active_object is not
```

## Features of Python

Here are some of the reasons why use Python

- Easy to learn and use
- Open-source language
- Cross-platform
- Broad standard library
- High-level language
- GUI support
- Multi-Paradigm language
- Extendable programming language

# Application of Python



## Python Install

- To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your name>python --version
```

- To check if you have python installed on a Linux or Mac, then on Linux open the command line or on Mac open the Terminal and type:

```
python --version
```

- If you find that you do not have Python installed on your computer, then you can download it for free from the following website: https://www.python.org/

# Python Quickstart

- Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.
- The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python hello.py
```

- Where "hello.py" is the name of your python file.
- Let's write our first Python file, called hello.py, which can be done in any text editor.

```
print("Hello, World!")
```

- Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python hello.py
```

- The output should read: `Hello, World!`

You have written and executed your first Python program.

# Python Command Line

- To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.
- Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

- Or, if the "python" command did not work, you can try "py":

```
C:\Users\Your Name>py
```

- From there you can write any python, including our hello world example from earlier in the tutorial:

```
>>> print("Hello, World!")
```

- Which will write "Hello, World!" in the command line:
- The output should read: `Hello, World!`
- Whenever you are done in the python command line, you can simply type the following to quit the python command line interface: `exit()`

## Syntax

print("Hello!")

name = "Naima"

x = 10

Syntax refers to the rules that define the structure of a language.

In computer science, the syntax of a computer language is the rules that defines the combinations of symbols that considered to be correctly structured statements or expressions in that language.

## Python Syntax

## Execute Python Syntax

As we learned, Python syntax can be executed by writing directly in the Command Line:

```
>>> print("Hello, World!")
    Hello, World!
```

# Python Indentation

- Indentation refers to the spaces at the beginning of a code line.
- Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.
- Python uses indentation to indicate a block of code.
-

```
if 7 > 2:
  print(“Seven is greater than two!")
```

# Python Indentation

Python will give you an error if you skip the indentation:

**Syntax Error:**

-
  ```
  if 5 > 2:
  print("Five is greater than two!")
  ```

- The number of spaces is up to you as a programmer, the most common use is four, but it has to be at least one.

**Example**

-
  ```
  if 5 > 2:
   print("Five is greater than two!")
  if 5 > 2:
          print("Five is greater than two!")
  ```

- You have to use the same number of spaces in the same block of code, otherwise Python will give you an error.

## Python Comments

1

Comments can be used to explain Python code.

2

Comments can be used to make the code more readable.

3

Comments can be used to prevent execution when testing code.

## Python Comments

### Single Line Comment:

Comments starts with a #

- ✓
```
#This is a comment
print("Hello, World!")
```
- ✓
```
print("Hello, World!") #This is a comment
```

### Multi Line Comments:

To add a multiline comment you could insert a # for each line or triple quotes.

- ✓
```
#This is a comment
#written in
#more than just one line
```
- ✓
```
"""
This is a comment
written in
more than just one line
"""
print("Hello, World!")
```

# Modules And PIP

**Module :** A module is a file containing a set of functions you want to include in your application.

**PIP:** PIP is the package manager for python, you can use pip to install a module on your system.

- A package contains all the files you need for a module.
- Modules are Python code libraries you can include in your project.
- pip install flask >>installs flask module.

**Types of modules:**

- There are two types of modules in Python.

1. **Built in modules** → Pre installed in Python→ OS, time, etc.
2. **External modules**→ Need to install using pip--> tensorflow, django, flask etc

# Practice Work

1. Use a built-in module of your interest.
2. Install an external module and use it to perform an operation of your interest.

# THANK YOU