BASIC ORIENTATION TO

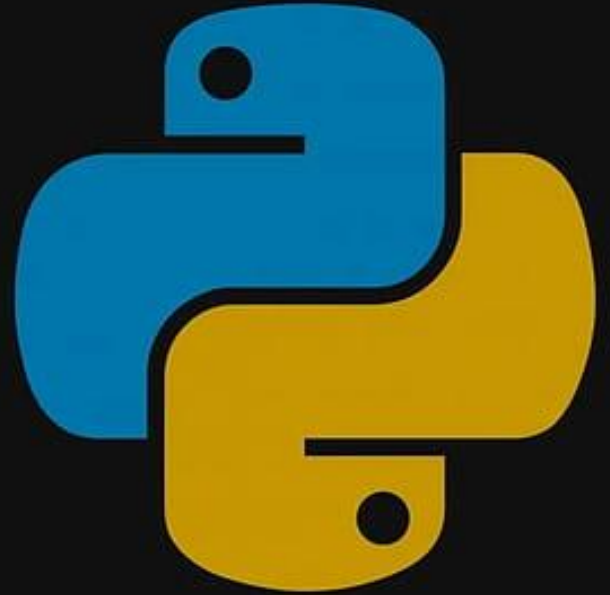**Medical Education Unit
UCMS and GTB Hospital, Delhi**

Disclaimer

- Academic and non-commercial use only
- Prepared by Apoorv Chauhan, Medical Student, UCMS and GTB Hospital, Delhi
- For Python basics workshop for UCMS and GTBH Faculty students held on 23rd Sep 2023

Programming is an Art

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

Why Should Doctors Code?

Navigating technologies in healthcare

Technology has become the mainstay of interaction

Electronic health records

Making healthcare system more efficient

Deal with bugs

Good coding skills can help a medical professional properly evaluate those technologies and help guide the hospital or their patients on which app to use

Coding helps to improve clinical competence

Programming involves logical steps which when combined can produce a set of actions.

Constantly involving oneself in such can structure the brain in problem-solving.

It help a clinician to follow rules during their practice knowing that they are present for a purpose and bypassing them may lead to undesirable results

Coding helps doctors participate in health technology creation and development

Engineers and scientists may often find it difficult to comprehend the nature and impact of their technology in healthcare provision

As most technologies are becoming reliant on computers, knowledge of programming can help a doctor invent or properly contribute to the creation or development (including development) of a health tech from apps to full moveable machines

Our health system is desperate for innovation

We need doctors with new-age skills to invent new-age solutions to age-old problems.

- An email list that intermittently provides patients education about their disease (ie obesity, diabetes) with the end goal of improving patient compliance with interventions
- A mobile app for patients to maintain their medications on, including the reason they are on each one

Programming aids in better evidence-based medicine

Evidence-based medicine is pivotal in current medical practice and it is heavily dependent on data science. Doctors with good skills in coding will be better at acquiring useful data and analyzing it.

To Become Irreplaceable

AI Revolution of modern times

Doctors who understand the intricacies of the systems that are replacing practitioners

MedTech is an exploding field.

Learning to code will set apart a doctor from the rest and help such one make the most impact in the healthcare system.

What is Python?

Python is a high-level programming language that uses instructions to teach the computer how to perform a task. It's easy to learn and powerful.
High-level means that it is closer to a human language like English.
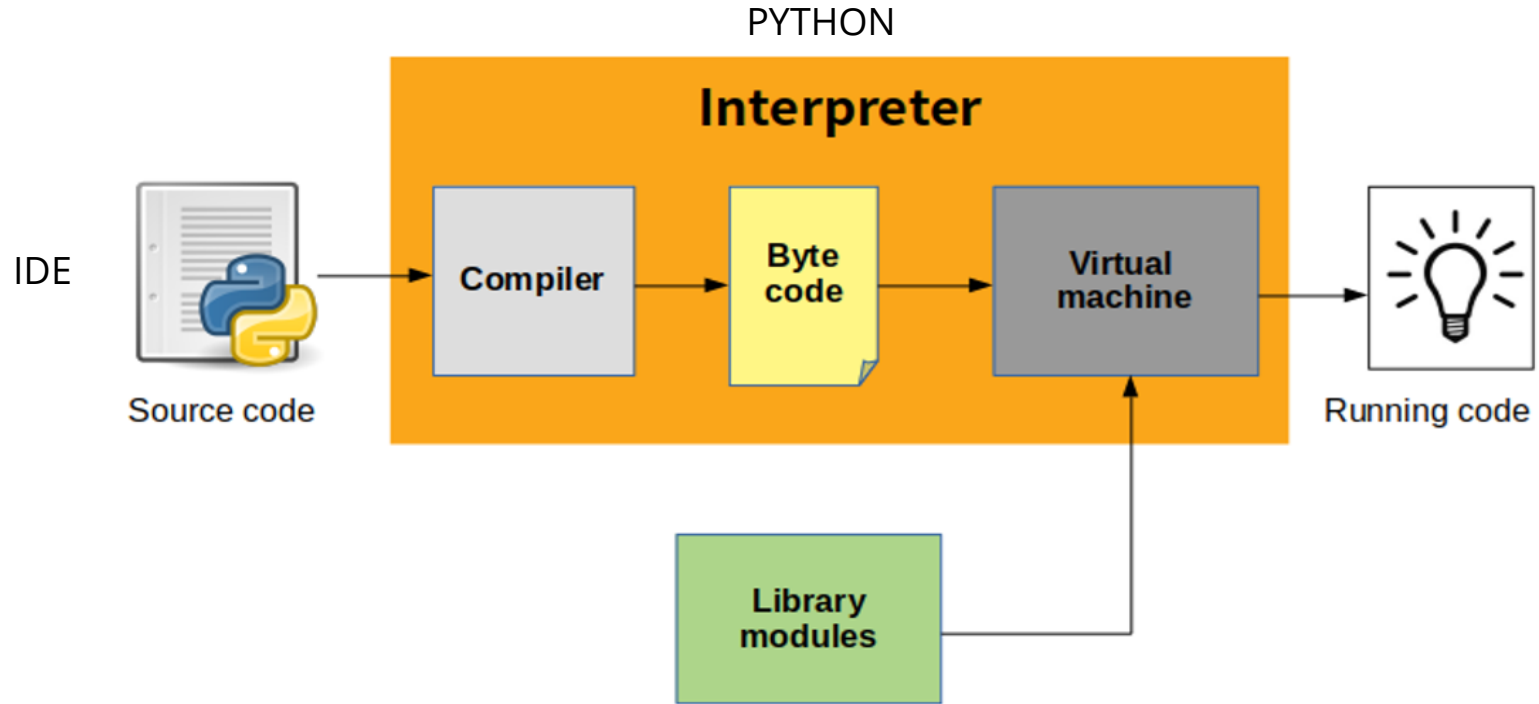
It's free and open source.

There are multiple versions of Python. Currently, the latest is Python 3.11.

Where is Python Used?

Python is used in:
- App and web development
- Artificial Intelligence, Machine Learning
- Data Science
- Desktop Applications
- Gaming

How Python Works?

Python as a Programming Language

A programming language is a language used to create programs.

- *More Versatile*
- *Easy to Learn - relatively*
- *Close to Daily Use Language*

Other programming languages: C++, JAVA

Let's Get Started!

Programs in Python

A program is a set of instructions that tells the computer how to perform a specific task.

In this workshop, we will be using Python to give instructions to the computer.

Does this mean we can make our computers do anything?

We want to display "Hello World" on the output screen. But, how?

Syntax (is like grammar)

Syntax refers to the rules that define the structure of a language. Syntax in computer programming means the rules that control the structure of the symbols, punctuation, and words of a programming language.

or

Syntax refers to a concept in writing code dealing with a very specific set of words and a very specific order to those words when we give the computer instructions.

print()

It is a built-in function in Python to display things on the screen. A built-in function is a function that is predefined and can be used directly.

Whatever we want to display, needs to be written inside brackets and enclosed within double quotes "..".

Comments

Comments are additional information mentioned along with 'codes' so that we know the context in which the code has been written.

This helps the programmer to understand their own codes later also in the correct context and to communicate with others also.

Comments are ignored by the Python Interpreter.

Python supports single-line comments

Anything written following '#' is considered a single-line comment in Python.

Codes are case and indentation sensitive

Coding is case sensitive. Beware of the capital letters and small letters.

Indentation in Python refers to the spaces and tabs that are used at the beginning of a statement.

Data Storage in Python & Variables

In programming variables act as containers to store values.

These values can be anything, from numbers to text or entire functions.

The values stored in the variables can be used further in our programs.

We can retrieve them by referring to the variable names which are used to identify these variables.

The variable is the name and assignment operator(=) is used to assign a variable (name) to the values.
E.g.
age = [20, 25, 45, 35, 22]

Data Types in Python

Strings -
Alphanumeric
Anything written between "..."(double quotes) or '...'(single quotes) is a string in Python.

Numeric -
When the value of a variable is a number, it's called a numeric type variable. E.g. Age
Could be integers(without decimal or fraction) or float numbers

Boolean -
True / False
AND/ OR / NOT

User Input in Python

To ask for any information or input from the user in Python, we just need to use the input() function. This will ask the user for some input. While asking for user input(), we can add extra information to tell the user what information they should provide.

```python
int(input("Enter your name: "))
```

Lists

A list is a data structure in Python which is used to hold multiple values, just like what we do with a shopping list.

These values may or may not be of the same type.

Data structure is a format for storing data.

A list contains items separated by commas. These items are enclosed within square brackets.

A variable can be used to store a list.

Tuples

A tuple is a data structure just like a list which is used to hold multiple values. These values may or may not be of the same type. But if it was exactly the same as a list, we wouldn't need it. Right?

A tuple is a sequence of items separated by commas. These items are enclosed within parentheses(round brackets). A variable can be used to store a tuple.

Unlike lists, tuples are immutable. Immutable refers to the fact that items in a tuple cannot be updated or deleted. So, there is no update or deletion of items in tuples. But we can delete the entire tuple.

Dictionaries

A dictionary in Python is a data structure which contains data in the form of pairs of keys and their values.

A key and value pair forms an item in the dictionary. A key is usually a string.

Items in a dictionary are separated by a comma. Note, the last item doesn't have a comma following it. Keys and values are separated by a colon(:).

Items in a dictionary are enclosed within curly brackets

Calculations & Numerical Operations

Arithmetic Operators in Python are the basic math operations listed below:

     +   : Addition Operator(plus) - Adds two number

     -    : Subtraction Operator(minus) - Subtracts two numbers

    /   : Division Operator(slash) - Divides two numbers

    *   : Multiplication Operator - Multiplies two numbers

   % : Modulo Operator - returns the Remainder of the division operation

   // : Truncate Operator - returns the Quotient of the division operation

Comparison Operators

Now, let's talk about Comparison Operators.
These operators compare the values on either side of the operand and determine the
relation between them. They are also referred to as relational operators.
They return values in true or false.

Various comparison operators are (==, !=, <, >, <=etc)

Logical Operators

Logical operators in Python are used for conditional statements that are true or false.
Logical operators in Python are AND, OR and NOT.

For logical operators, following conditions are applied.
- AND operator - It returns True if both the operands(right side and left side) are true
- OR operator - It returns True if either of the operand(right side or left side) is true
- NOT operator - returns True if operand is false

Decision Making in Python

Just like we make decisions in our daily lives, our program also needs to make decisions. To make decisions in Python, we have decision-making statements. The decision making statements provided by Python are:

1. If
2. If-else
3. elif

An 'if' statement can be used to check if a condition is true or false. If the condition evaluates to true, the code block following the if statement is evaluated.

In Python, to indicate the block of code indentation is used. Each line must be indented by the same amount in a block of code. A typical if statement looks like:

if condition:
    statements if true

A colon(:) is used in Python to make the code more readable and to separate the different parts of the expressions.

Repeating Tasks in Python

Python provides us with an efficient way to do the repeat tasks - Loops.
Loops allow us to execute a statement multiple times. There are two types of loops in Python:

1. for loop
2. while loop

Let's see how we can use these in our programs.

A **for loop** repeats over a given sequence.

for variable in sequence:
    statement

Where *variable* refers to a counter variable. This variable will update every time the loop is executed. And *sequence* contains the data over which the looping needs to be done.

range()

range is a function in Python. range() takes a starting value and an ending value.
It generates a list of numbers between them inclusive of starting value and excluding
the ending value. In our case, from 1 inclusive to 101 exclusive.

If only one value is passed to range(), it assumes that the starting value is 0 and
generates a list of numbers.
Hence, range(100) and range(0,100) will give the same result.

A third parameter can also be passed to the range function, which represents the step.
A step is a number by which the index will change every time the loop executes.

By default, a step is 1.

Exercise!

Task: Write a program that prints the integers from 1 to 100. But for multiples of three print "Fizz" instead of the number, and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

How to Make Python Perform Tasks: Functions

Functions are a set of instructions which perform a specific task and can be reused multiple times. The specific task can be adding two numbers or finding the greatest number in the list.

Functions in Python are written as described below:

```
def functionName(parameters):
    statements
    return something
```

def is used to declare a function.
functionName is the name of the function.

Exercise!

LESSER OF TWO EVENS: Write a function that returns the lesser of two given numbers *if* both numbers are even, but returns the greater if one or both numbers are odd

Hint: min & max functions

min(a,b) returns the minimum argument

max(a,b) returns the maximum argument

Exercise!

MAKES TWENTY: Given two integers, return True if the sum of the integers is 20 *or* if one of the integers is 20. If not, return False

A function may take in parameters, or may not. Parameters consist of variables used by the function, like the two numbers to be summed.
statements are the operations which we need to perform.

And, a return statement marks the end of the function and can be used to return the output. We can avoid putting it in if we don't have to return any value.

Modules

Any file containing Python code having a .py extension, for example, example.py is called a module.

The name of the file becomes the module name.

Inside the file, we can have definitions and implementations of classes, variables, or functions.

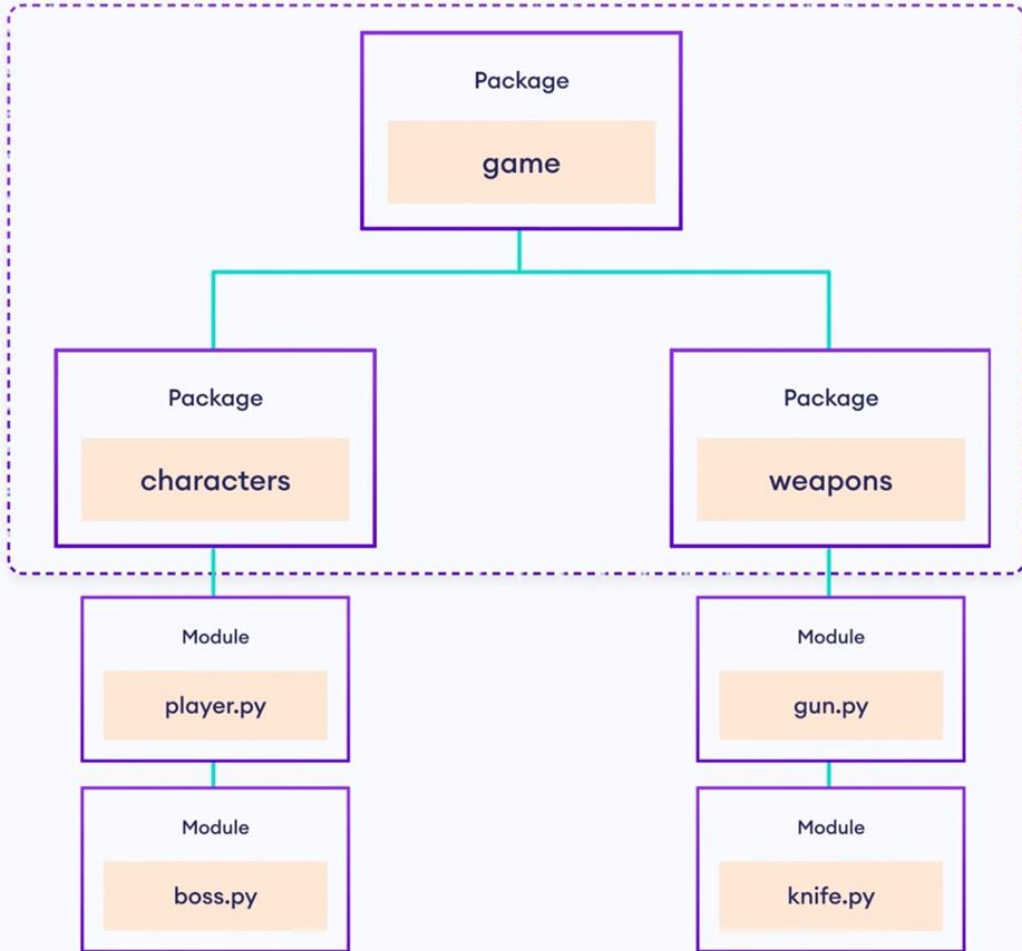These can then be used in other Python programs.

Benefits of Modules in Python

Structured Code:

- Makes development easier and less error-prone.
- Easier to understand and use.

Reusability:

- Can be used by other parts of application
- Eliminates duplication need

Package

game

Package

characters

Package

weapons

Module

player.py

Module

gun.py

Module

boss.py

Module

knife.py

The Import Statement:
To use the functionality present in any module, we need to import it into our current program.

import module_name

We can use the functions inside a module by using a dot(.) operator which allows us to call the function by first specifying the module name, and then the name of the function.

module_name.functionName()

Modules in Python could be either built-in or custom modules.
One such built-in module is the statistics module.