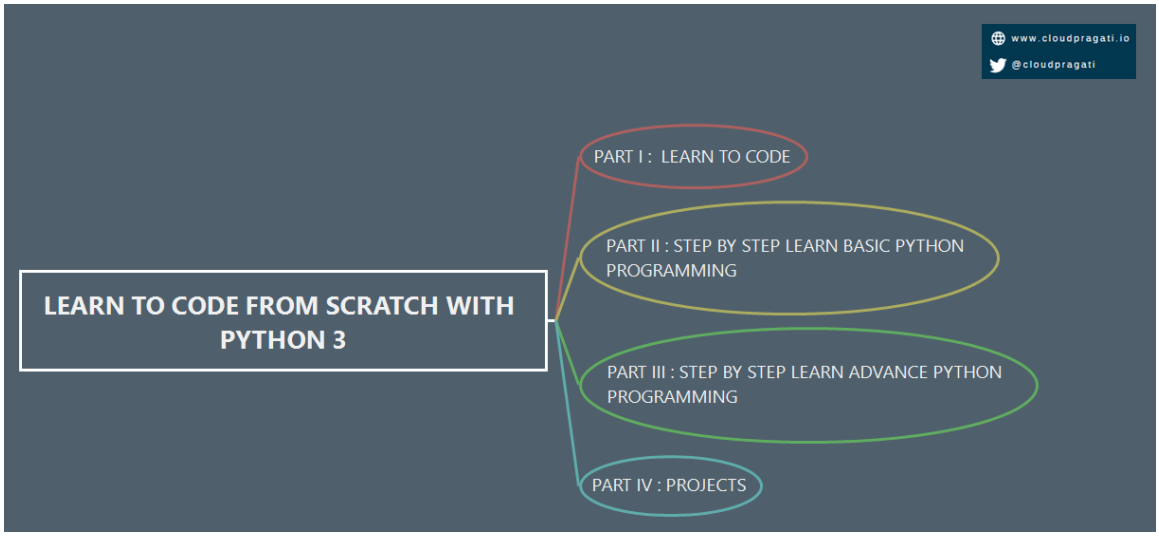
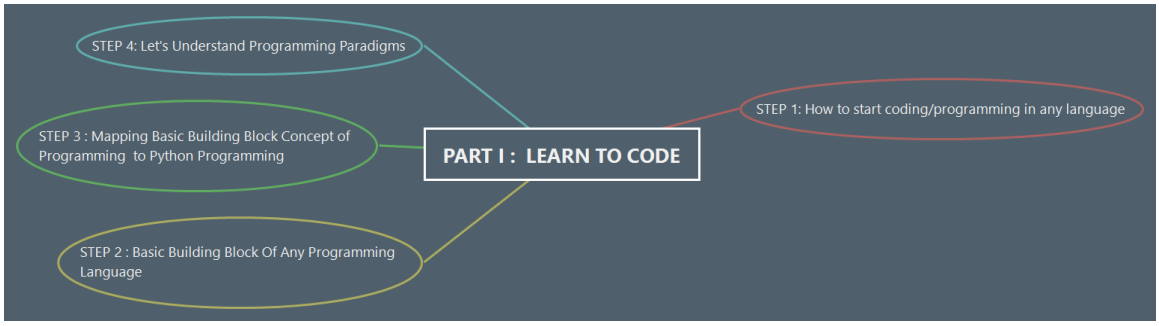
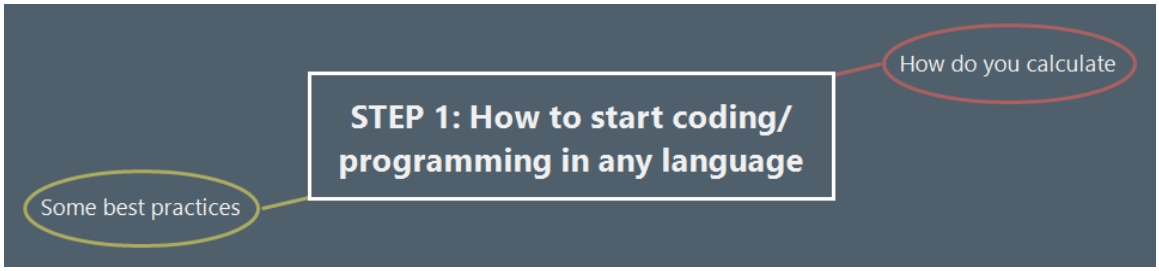
LEARN TO CODE FROM SCRATCH WITH PYTHON 3



1. PART I : LEARN TO CODE



1.1. STEP 1: How to start coding/programming in any language



1.1.1. How do you calculate

1.1.2. Some best practices

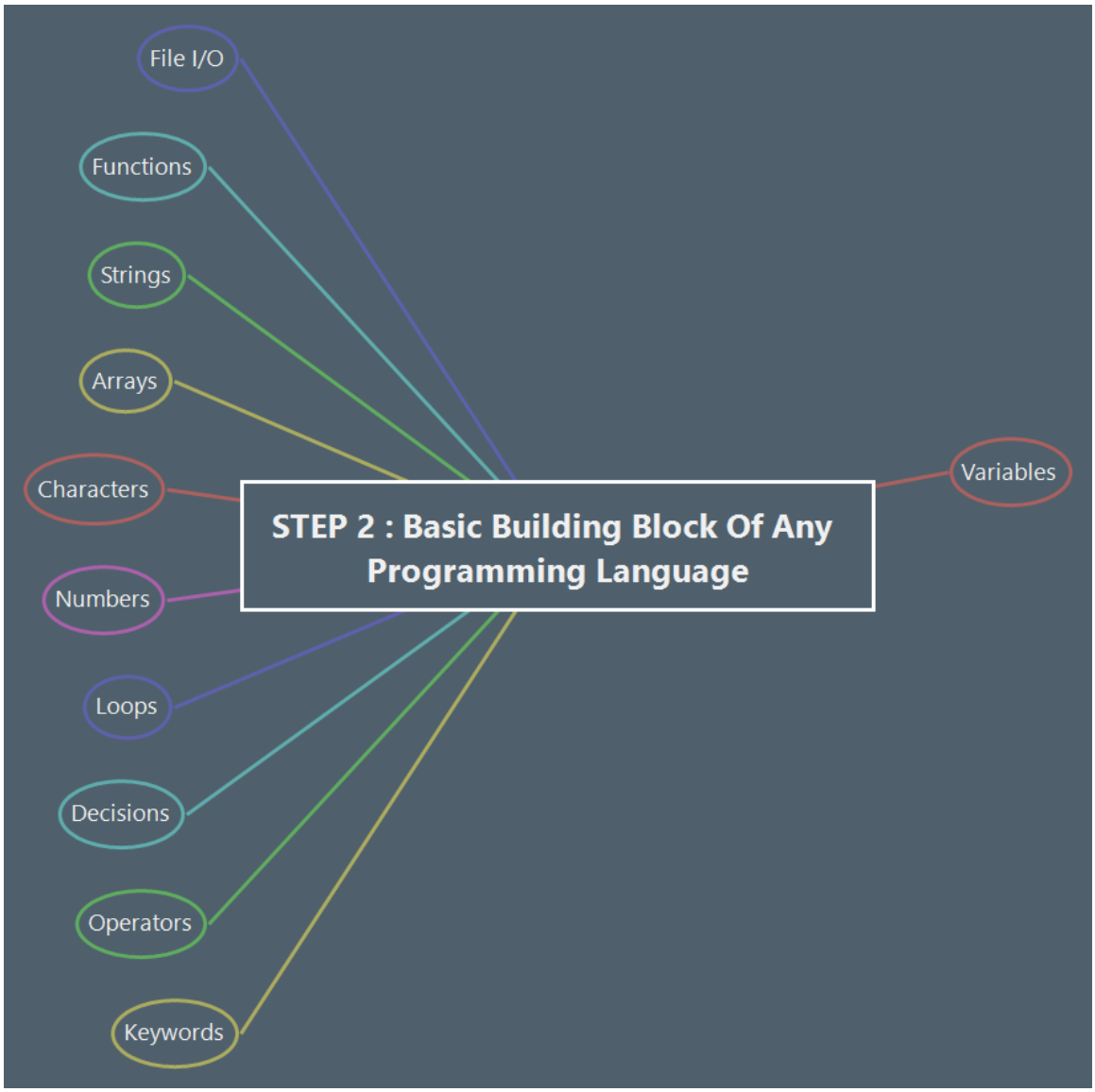
- Start writing pseudo code

- Convert pseudo code into a diagram

- Brush up some basic maths

- Analytical Reasoning

1.2. STEP 2 : Basic Building Block Of Any Programming Language



1.2.1. Variables

1.2.2. Keywords

1.2.3. Operators

1.2.4. Decisions

1.2.5. Loops

1.2.6. Numbers

1.2.7. Characters

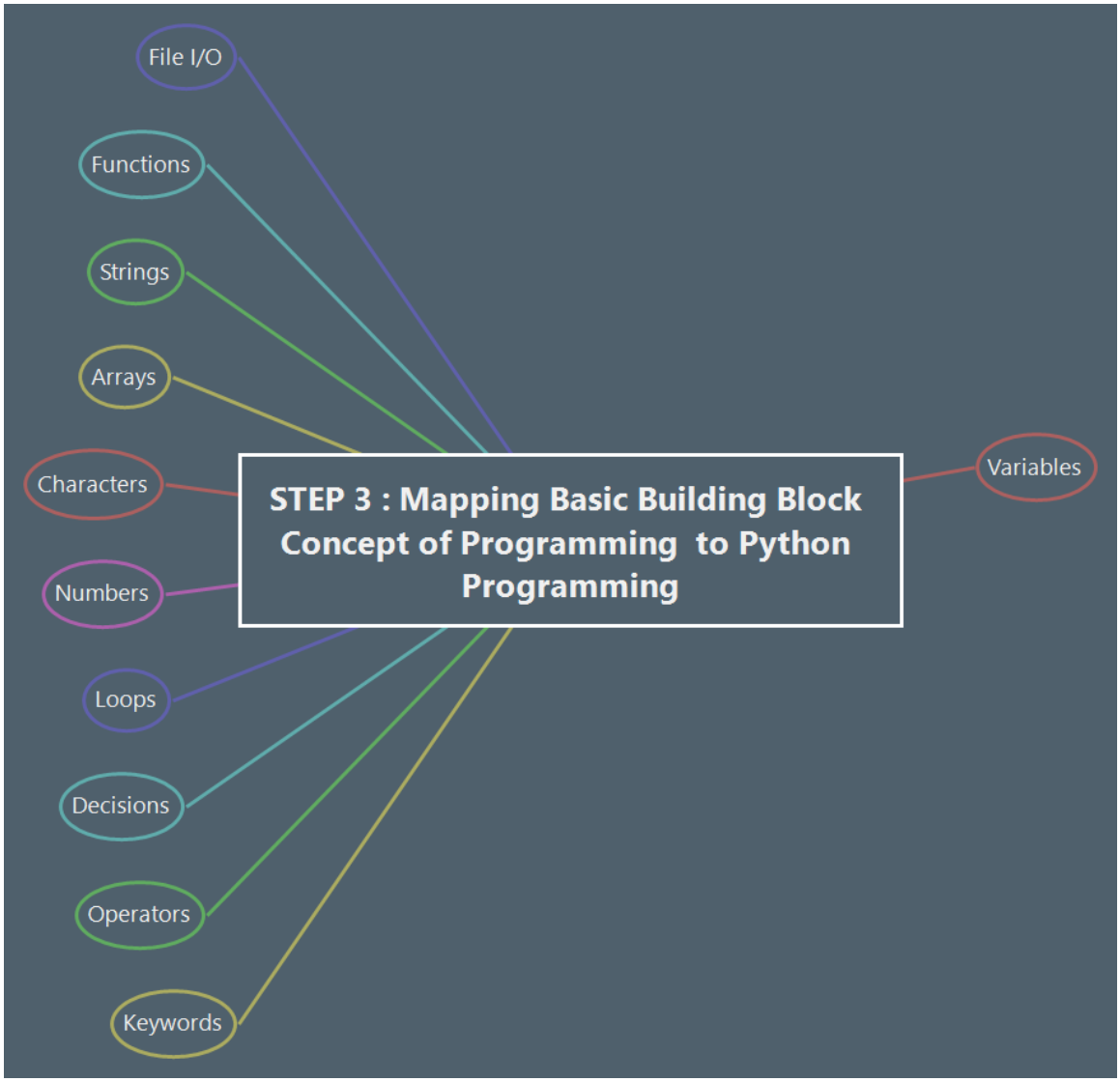
1.2.8. Arrays

1.2.9. Strings

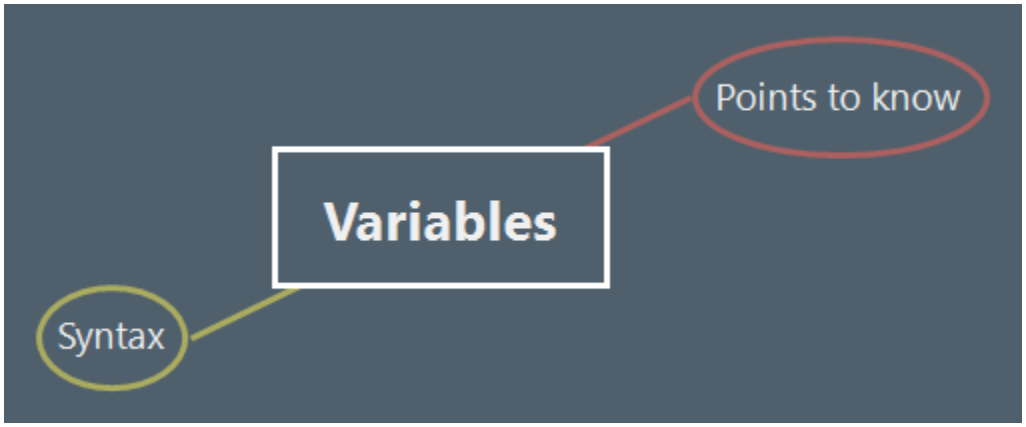
1.2.10. Functions

1.2.11. File I/O

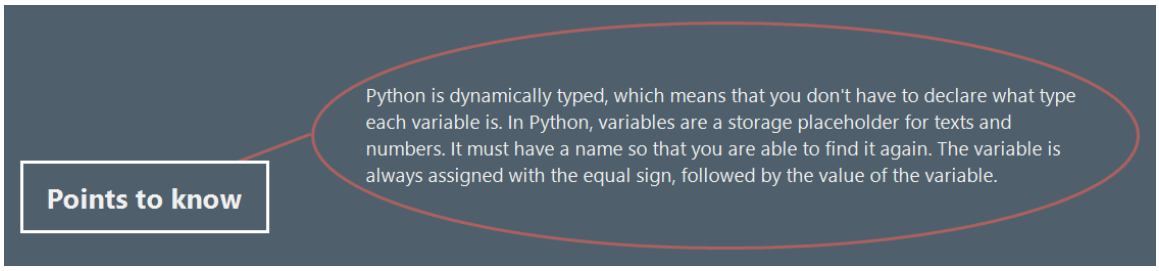
1.3. STEP 3 : Mapping Basic Building Block Concept of Programming to Python Programming



1.3.1. Variables

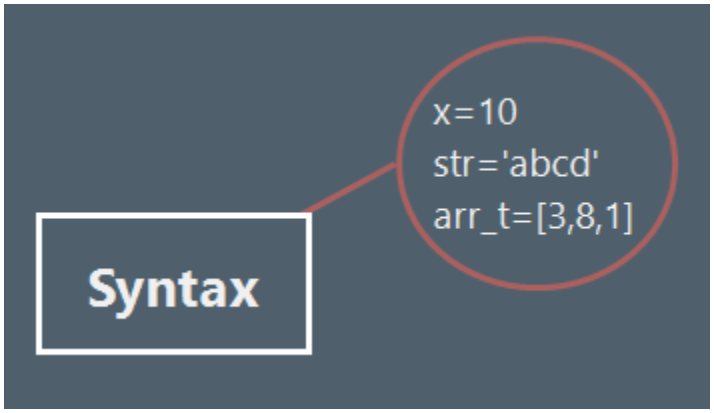


Points to know



Python is dynamically typed, which means that you don't have to declare what type each variable is. In Python, variables are a storage placeholder for texts and numbers. It must have a name so that you are able to find it again. The variable is always assigned with the equal sign, followed by the value of the variable.

Syntax

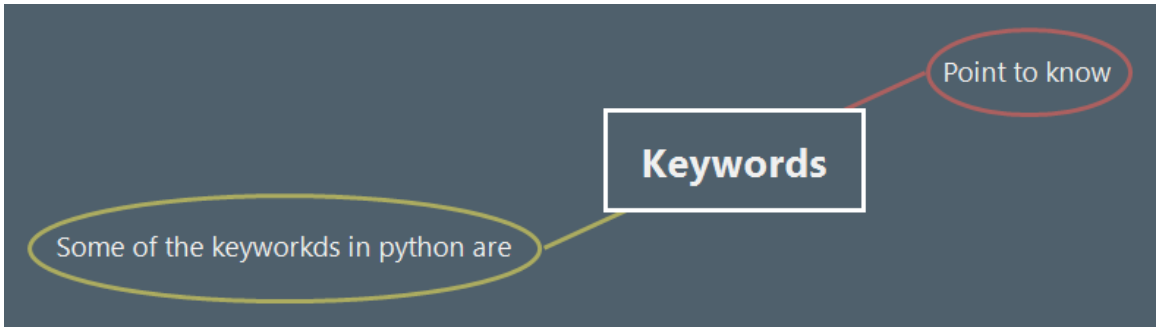


x=10

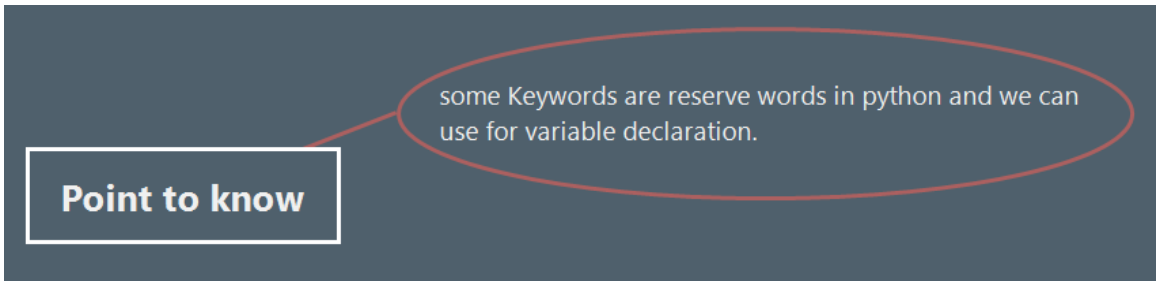
str='abcd'

arr_t=[3,8,1]

1.3.2. Keywords



Point to know



some Keywords are reserve words in python and we can use for variable declaration.

Some of the keyworkds in python are

1.3.3. Operators

1.3.4. Decisions

1.3.5. Loops

1.3.6. Numbers

1.3.7. Characters

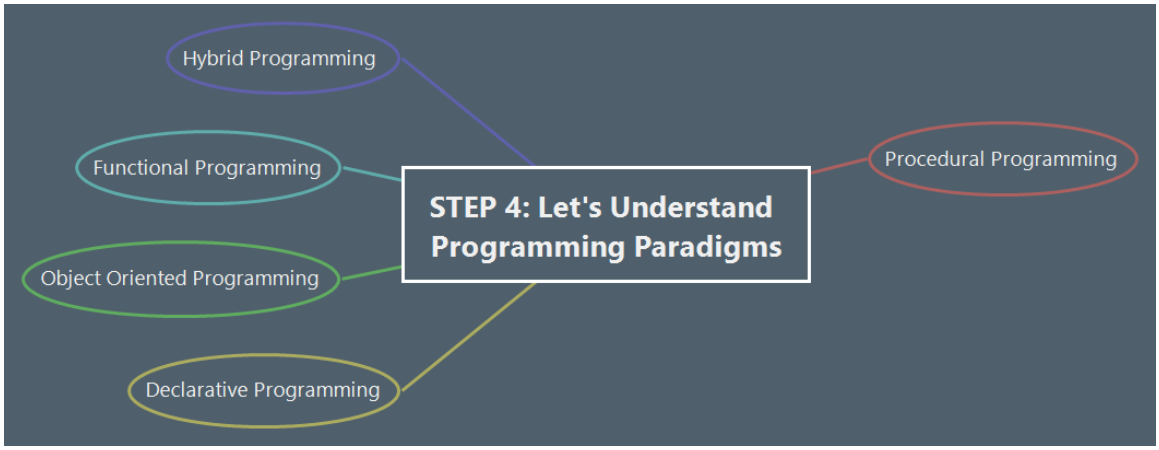
1.3.8. Arrays

1.3.9. Strings

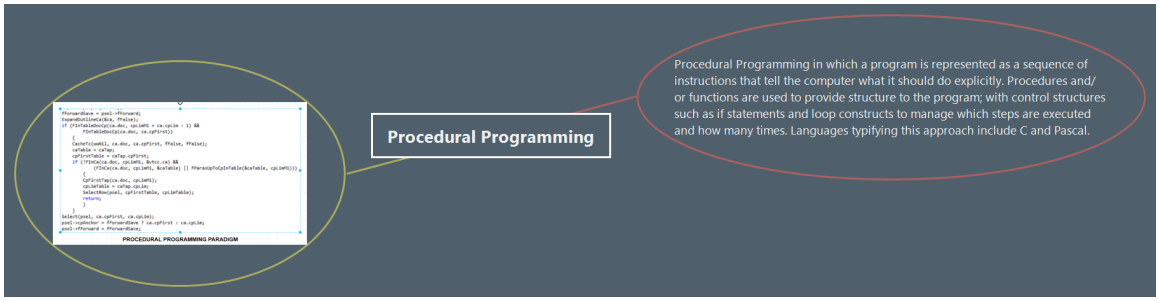
1.3.10. Functions

1.3.11. File I/O

1.4. STEP 4: Let's Understand Programming Paradigms



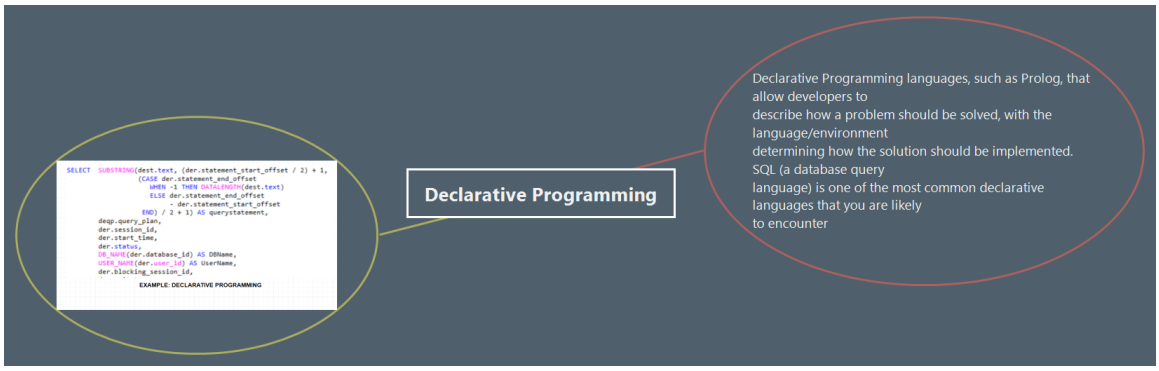
1.4.1. Procedural Programming



Procedural Programming in which a program is represented as a sequence of instructions that tell the computer what it should do explicitly. Procedures and/ or functions are used to provide structure to the program; with control structures

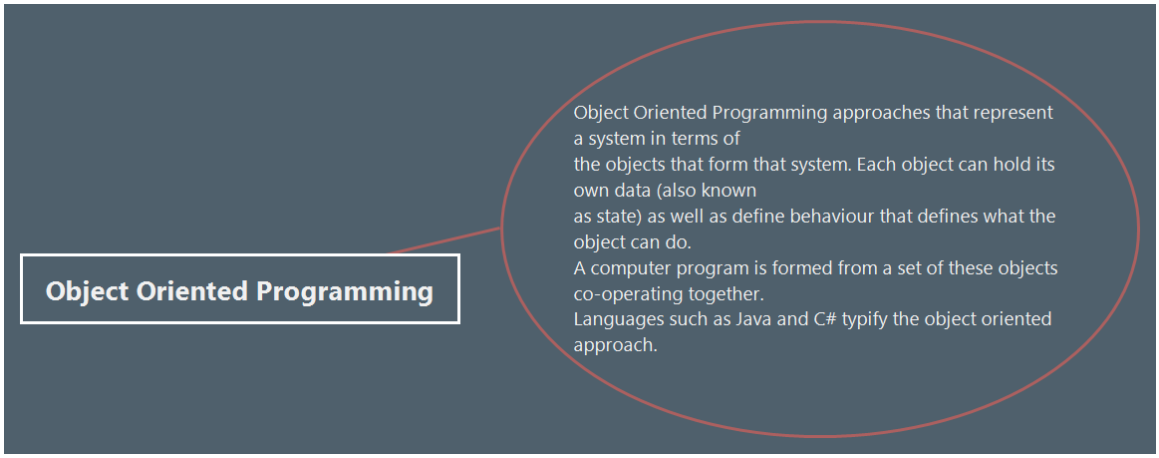
such as if statements and loop constructs to manage which steps are executed and how many times. Languages typifying this approach include C and Pascal.

1.4.2. Declarative Programming



Declarative Programming languages, such as Prolog, that allow developers to describe how a problem should be solved, with the language/environment determining how the solution should be implemented. SQL (a database query language) is one of the most common declarative languages that you are likely to encounter

1.4.3. Object Oriented Programming



Object Oriented Programming approaches that represent a system in terms of the objects that form that system. Each object can hold its own data (also known

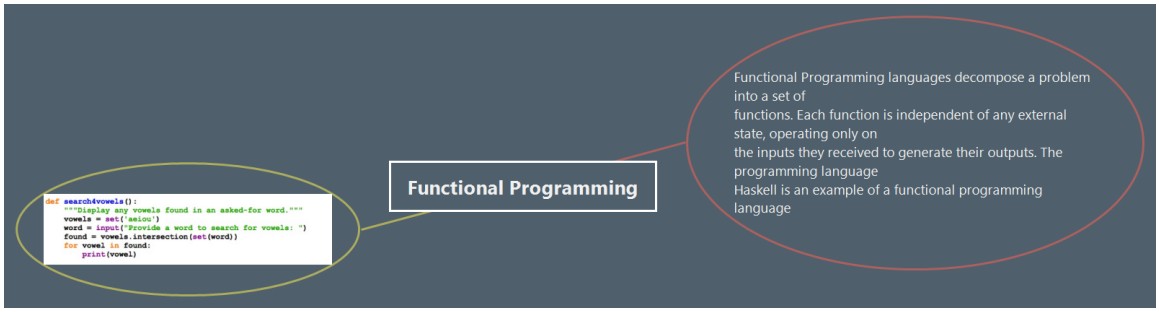
as state) as well as define behaviour that defines what the object can do.

A computer program is formed from a set of these objects co-operating

together.

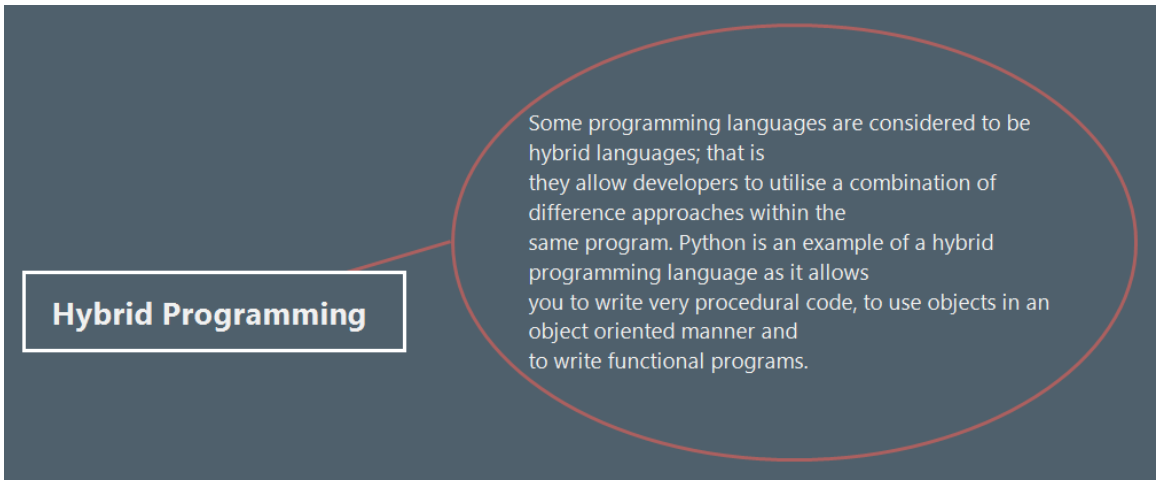
Languages such as Java and C# typify the object oriented approach.

1.4.4. Functional Programming



Functional Programming languages decompose a problem into a set of functions. Each function is independent of any external state, operating only on the inputs they received to generate their outputs. The programming language Haskell is an example of a functional programming language

1.4.5. Hybrid Programming



Some programming languages are considered to be hybrid languages; that is they allow developers to utilise a combination of difference approaches within the

same program. Python is an example of a hybrid programming language as it

allows

you to write very procedural code, to use objects in an object oriented manner and

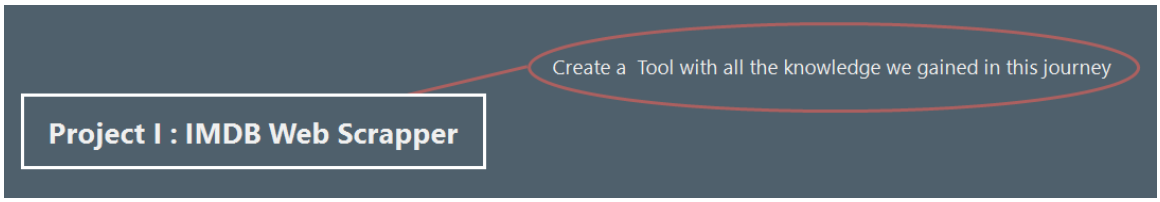
to write functional programs.

2. PART II : STEP BY STEP LEARN BASIC PYTHON PROGRAMMING

3. PART III : STEP BY STEP LEARN ADVANCE PYTHON PROGRAMMING

4. PART IV : PROJECTS

4.1. Project I : IMDB Web Scrapper



4.1.1. Create a Tool with all the knowledge we gained in this journey

4.2. Project II: Create a Game called Snake