There are different styles of programming languages. They are different in the way a problem is approached and coded. And hence there are different syntax for the same.

Broad classification is defined as:

Imperative programming HOW TO DO

It works by changing the program state through assignment statements. It performs step by step task by changing state. Examples are C, Fortran, Basic

It is further classified as:

Procedural programming: It is also known as inline programming. It is about writing a set of instructions encapsulated/bundled together as sub-routines to perform specific tasks. These subroutines are called procedures.

The key idea is that you are developing your own procedure

The program has data and procedures to work on the data. And these two are separate entities. The programs can be written with only three things:

Sequence (execution moves forward one statement at a time)

Selection (if statements)

Iteration (loops)

So essentially, we would break a complex system (problem) into smaller pieces (called procedures) and the main method would call these procedures whenever needed.

This is otherwise called as TOP-DOWN approach (start from a big, and end with smaller pieces - decomposition)

Examples: Fortran, COBOL and C

OOP: Here the program is built from objects. An object is an instance of a class, which is an encapsulation of data and the procedures (called methods here) that manipulate them.

A class is a blueprint of an object. You can think of a class as a concept and the object as the embodiment of that concept.

Examples: C#, Java, Perl and Python

Declarative Programming WHAT TO DO

The fundamental difference is that imperative programming describes HOW TO DO SOMETHING while declarative programming is about WHAT TO DO. It expresses the logic of computation without talking about its control flow.

Examples: SQL, HTML

Python is a HLL; it’s a mix

https://tylermcginnis.com/imperative-vs-declarative-programming/

Here the execution is based on expressions instead of statements (as in imperative programming).

Functional programming: Here the subroutines are called functions. These are pre defined functions that perform certain actions; but you do not know how they are performing the calculations in the back.

Example is Scala, Haskell

Python is a mix

Database/Data driven programming approach: It is based on data and its movement to provide features such as file creation, data entry, update, query and reporting functions