Best coding Practice:

There are a few best practices when it comes to learning how to code, and they center around these 7 concepts:

1. Variable naming conventions
2. Class and function naming conventions
3. Clear and concise comments
4. Indentations
5. Portability
6. Reusability and scalability
7. Testing

Variable naming convention:

Variable name should be easy to understand and clearly represent the data we store the way we name our variable is key to making our code reliable. The idea of variable naming while coding is simple to create variable names that are self-explanatory theory and follow a consistent theme throughout the code. Some types of variable naming conventions are as follows:

Multiword Delimited – Used to separate multiple words in a variable name without white space.

Hungarian Notation – This convention describes the purpose or type of the variable in the first part of the variable name, and then uses a descriptor to indicate the variable's function. In Hungarian notation, the camelCase notation is used to delimit the words.

Class and function naming conventions:

Similar to variable naming conventions, functions and classes should also consist of descriptive titles that are delimited by using conventions, as mentioned above. The purpose of using appropriate naming conventions is to make sure that the variables, functions, and classes within our code can be easily distinguished from one another.

Clear and concise comments:

Coding comments are segments of code that are ignored by the compiler. This means they are never passed to the computer and are not processed. Their sole purpose is to help the programmer understand the code, especially when returning to work on unfamiliar scripts in the future. Comments can be broadly classified into:

a). Implementation comments:

Implementation comments are meant for commenting out code or for comments about the particular implementation. Doc comments are meant to describe the specification of the code, from an implementation-free perspective. to be read by developers who might not necessarily have the source code at hand.

b). Documents comments:

A doc comment is an ordinary multiline comment that begins with /** (instead of the usual /*) and ends with */. A doc comment appears immediately before a class, interface, method, or field definition and contains documentation for that class, interface, method, or field.

Indentations:

Formatting and indentation are necessary to organize our code. Ideal coding formatting and indentation include correct spacing, line length, wraps and breaks. By employing indentations, white-spacing, and tabs within the code, programmers ensure their code is readable and organize. Problems and confusion only occur when indentation and formatting styles change halfway through a script.

Portability:

Portability is a key aspect that ensures functionality of our program. If our code contains literal (hard-coded) values of environmental parameters, such as usernames, host names, IP addresses or URLs, it will not run on a host having a different configuration than ours. In order to tackle this, we would have to 'parametrize' variables and configure them before running our software in different environments. This can be controlled with property files, databases, or application servers. In addition, resources like XML files must also have variables instead of literal values. Otherwise we'll have to change the references while coding every time we want to port our application to another environment.

Reusability and scalability:

In coding, reusability is an essential design goal. Because if modules and components have been tested already, a lot of time can be saved by reusing them. Software projects often begin with an existing framework or structure that contains a previous version of the project. Therefore, by reusing existing software components and modules, we can cut down on development cost and resources.

Testing:

Testing our work while coding is a vital part of software development and should be well-planned. It requires the test cases to be prepared before the actual coding of the software begins. Also, while basic unit testing is a good practice to adopt, it is also useful to perform Automated Functional Testing (AFT) with the help of tools such as Geb Spock and Selenium.