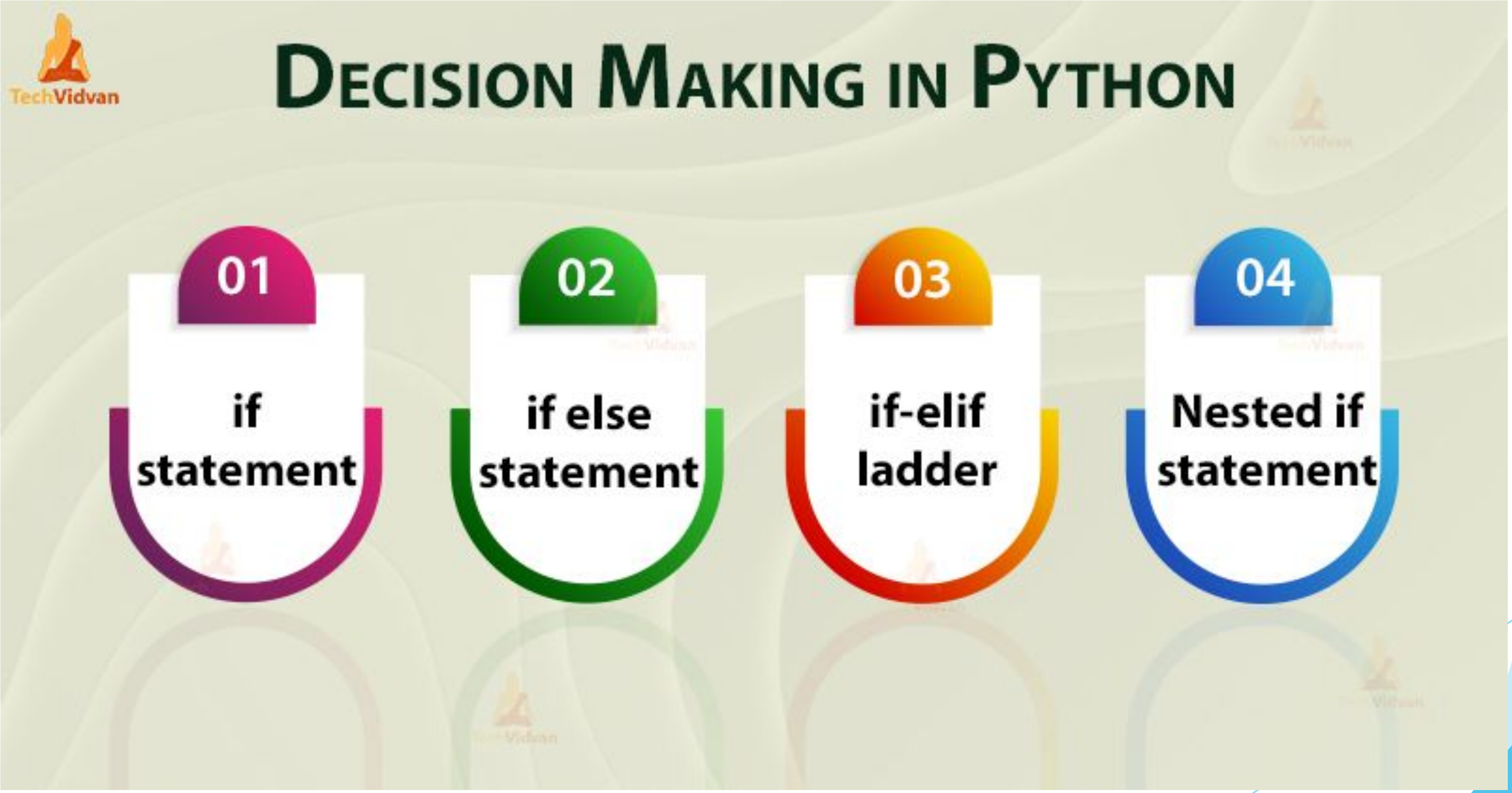
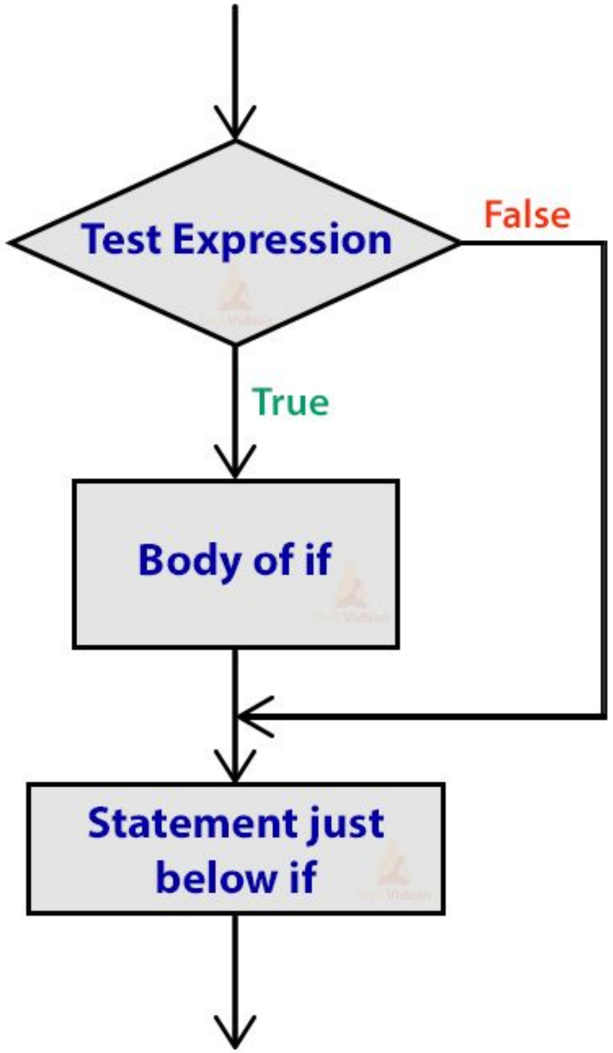
Python Session

DataDNA

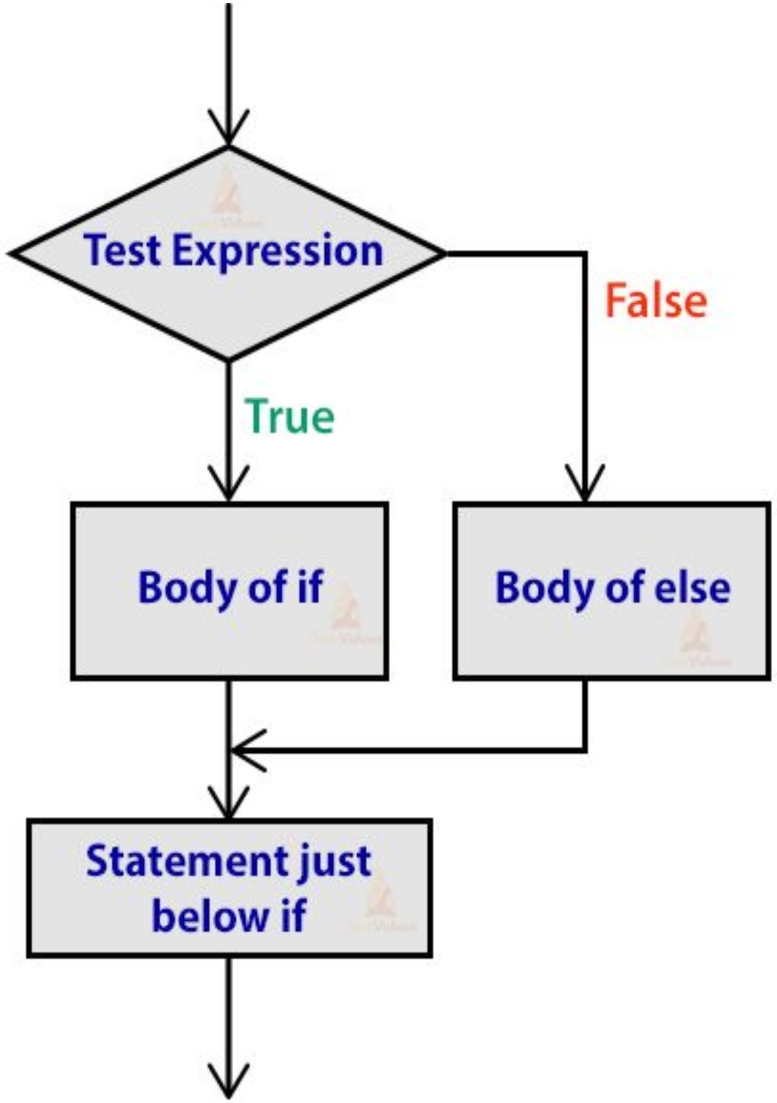
Decision Making



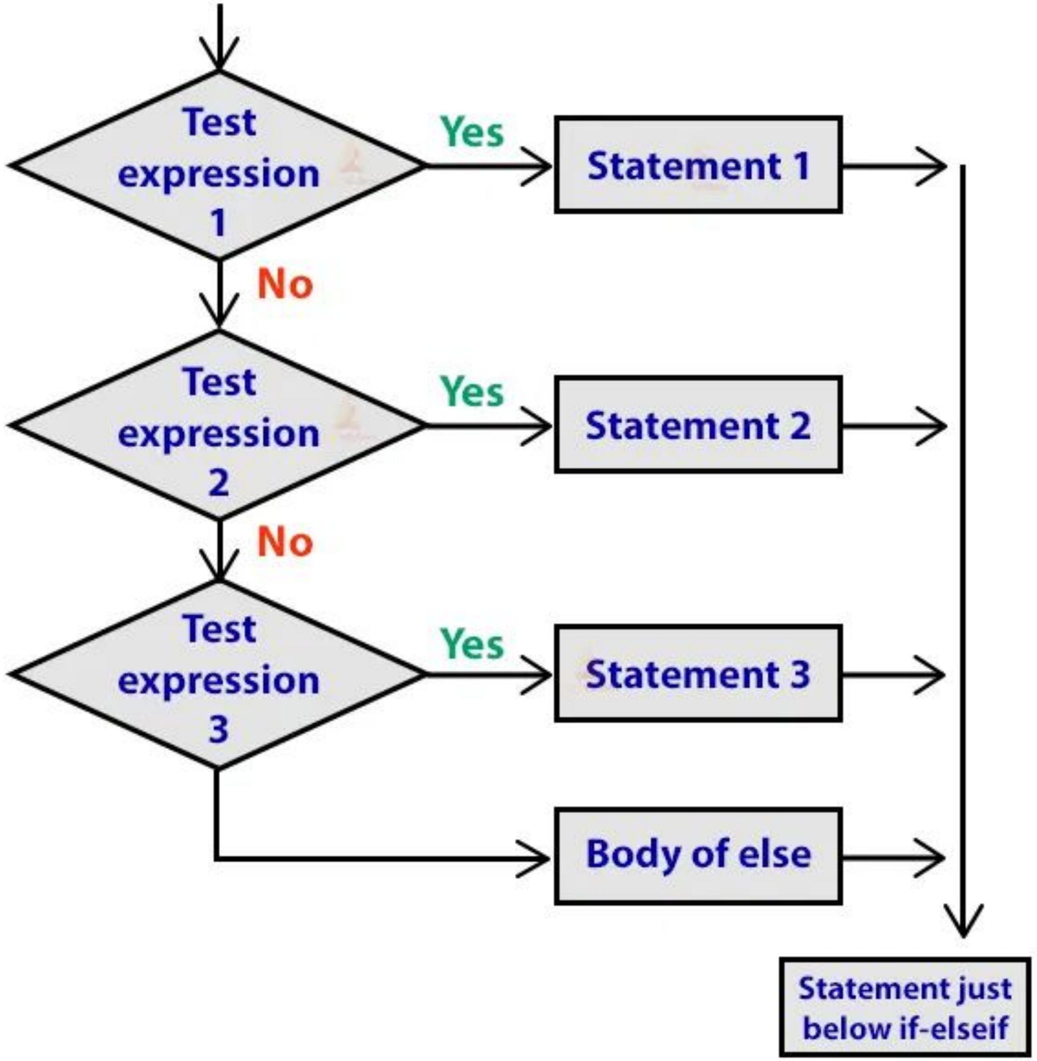
If Statement



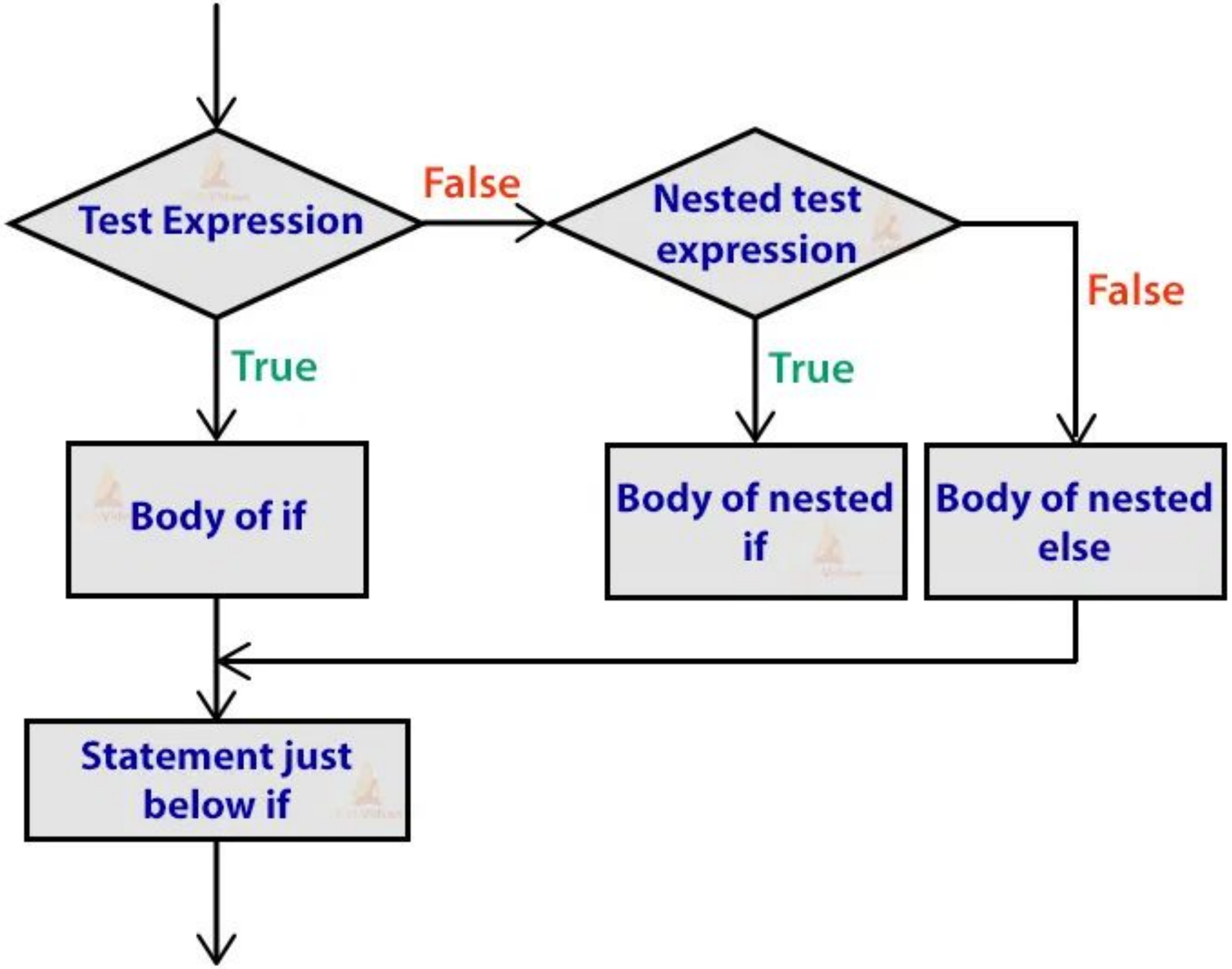
If Else Statement



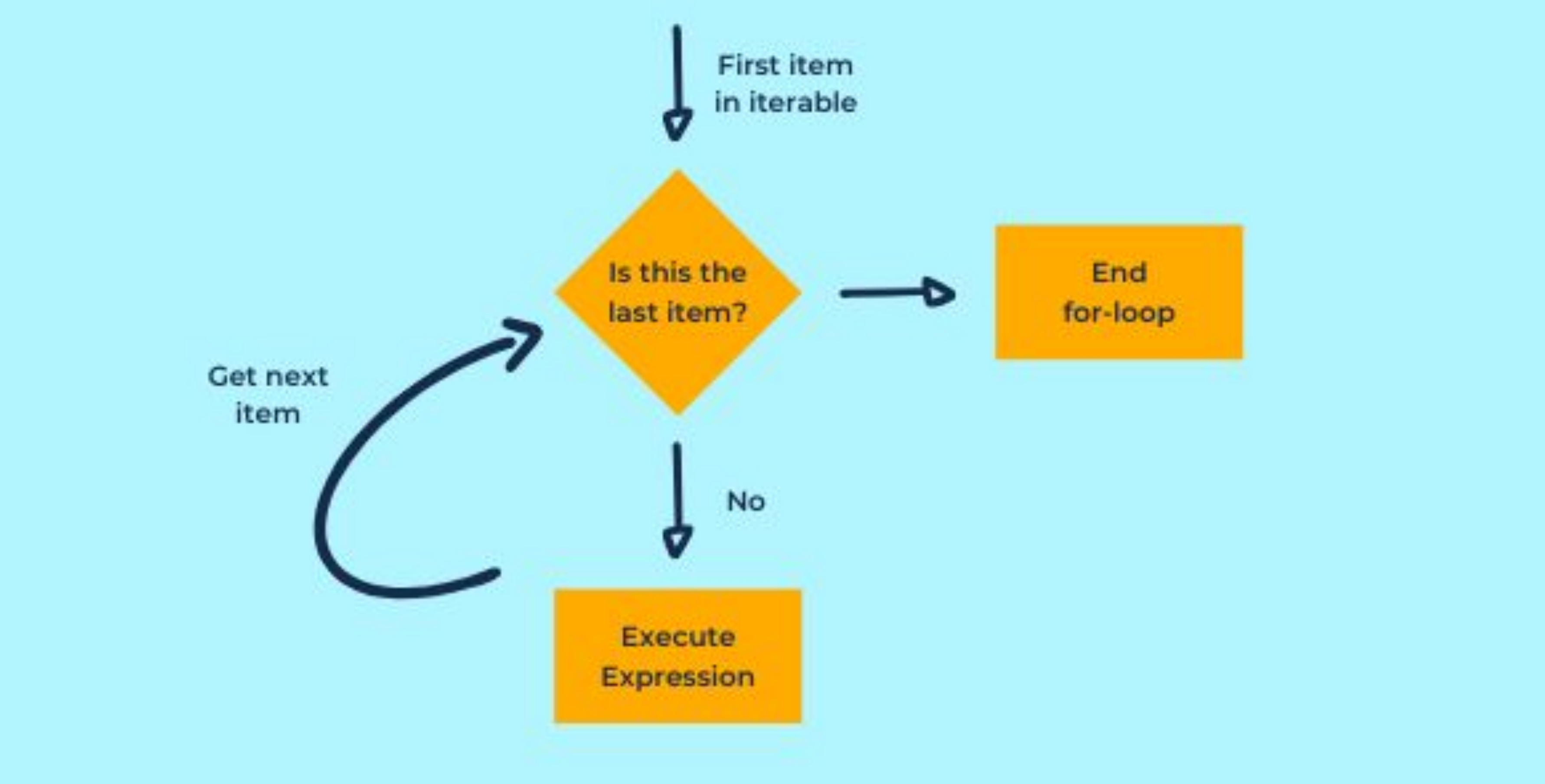
If Elif Ladder



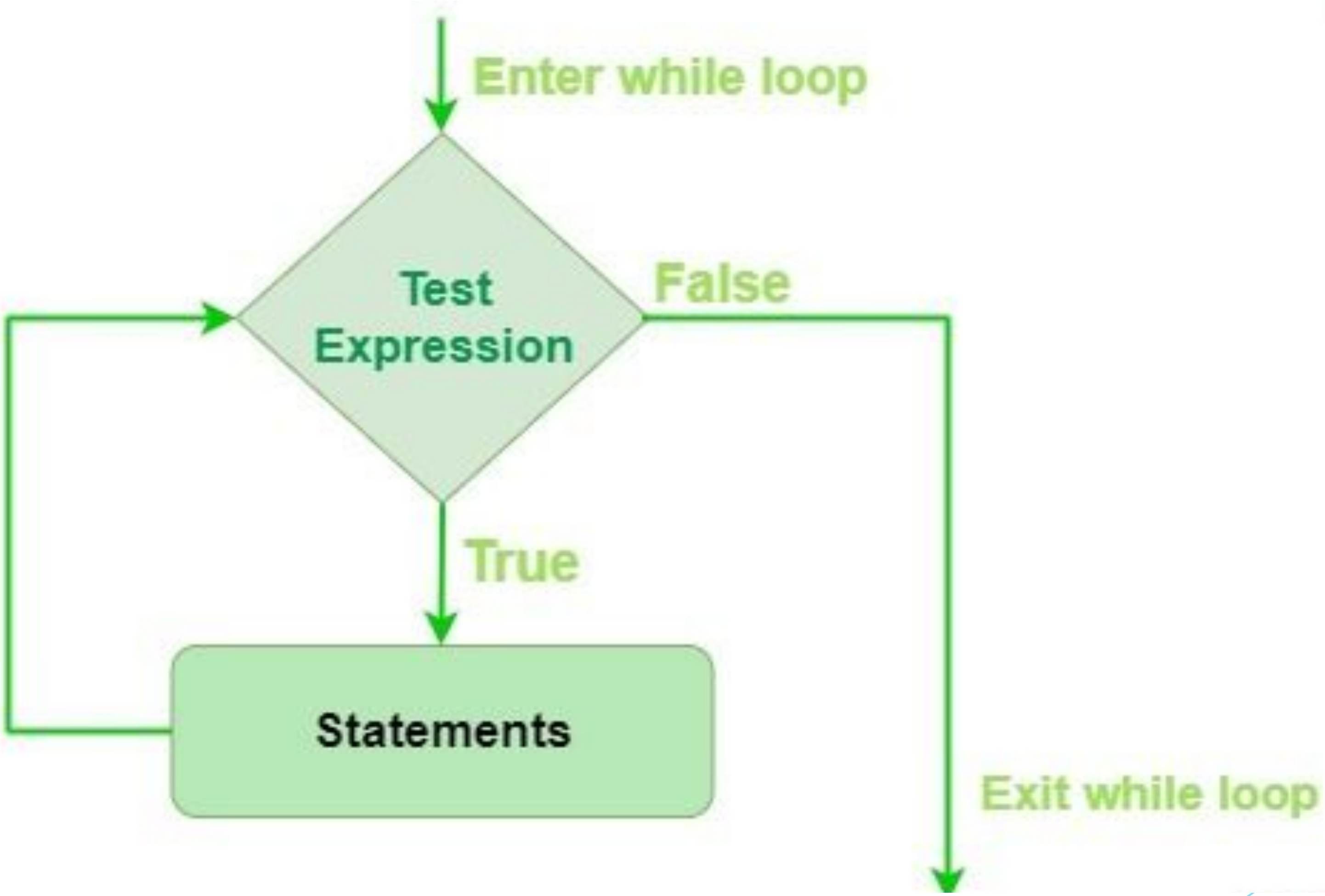
Nested if Statement



For Loop - Python



While Loop - Python



Key difference between For and While Loop: In case of For Loop, we know the number of iterations. But in the case of While Loop, the number of iterations will go on until the test expression holds true.

Functions

A block of code that is dedicated to do a particular task and reusable in nature. useful in structuring the code.

- **Built In Functions** (Eg:- print())
- **User Defined Functions** (Functions defined by User to get some tasks repeated in the project)
- **Anonymous functions** (Lambda Function: Function not declared in standard way like def. Lambda is used to create simpler expressive function. It can take multiple values but just return one.)

Functions

Things to note while creating a function:

- Starts with keyword : def
- arguments: Inputs (otherwise called as Parameters)
- code to perform a task
- docstring : Information about what specific function does
- return : Output

Example Syntax

```
def function_name( arguments):
        "docstring"
         code
         return output
```

Functions

Argument Types:

1. **Required arguments** (should be given in the order in which it is defined in the function)

2. **Keyword arguments** (When keyword is used while calling a function, order does not matter in this case)

3. **Default arguments** (Giving the default value for argument while defining the function. So the user does not require to a value while calling the function)

4. **Variable-length arguments** (Automatically handle the any given number of parameters)

Modules

- A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.

- Simply, a module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code.

The import Statement

You can use any Python source file as a module by executing an import statement in some other Python source file.

Syntax: Import module_name **[! Module_name.py is the module file]**

Modules

Locating Modules

When you import a module, the Python interpreter searches for the module in the following sequences –

- The current directory.
- If the module isn't found, Python then searches each directory in the shell variable PYTHONPATH.
- If all else fails, Python checks the default path.

Directory Handling

Tasks:

- Knowing current working directory
- Create a new directory
- Rename an existing directory
- List the files in a new directory
- Directory Traversal
- Delete a directory

Library Used: OS

File Handling

Python can facilitate doing following things in terms of handling files.

- **Creating -** f.open(<<filename>>,<<mode>>)

  x=> Create, Will create a file if file does not exist, if not it will thrown error

  w => will create a file, if mentioned file does not exist

  a = > Append, will append the given content to the mentioned file

  !Note: ‘w’ will overwrite the previous data whereas the ‘a’ will append the data to the previous one

- **Updating - f.write(<<text>>)**

  Works hand-in-hand with f.open()

  Works in collaboration with given mode.

File Handling

Python can facilitate doing following things in terms of handling files.

- **Reading**

  f.read() - Reading whole file at a time

  f.readline() - Reading one line of the file at a time

  f.readlines() - Creates the list of all lines in the file.

- **Deleting**

  os.remove(<<file_path>>)

File Handling

- Working with word document in the same way we do for text file gives encoding error
- Hence for efficiently working with word document requires library like **python-docx**

  install it using <<pip install python-docx>>

- Likewise, for working with pdf, we can go for libraries like **pypdf, pdfminer**
- **General Note: So whenever you require to work with some kind of file, look for a relevant library in web and leverage it for your project.**

File Conversion

- Convert python notebook to python script

```
!jupyter nbconvert pynb_conversion.ipynb --to script
```