1. Base

–Init : will be Files named \_\_init\_\_.py are used to mark directories on disk as a Python package directories. If you have the files

mydir/spam/\_\_init\_\_.py

mydir/spam/module.py

and **mydir** is on your path, you can import the code in module.py as:

import spam.module

or

from spam import module

If you remove the \_\_init\_\_.py file, Python will no longer look for submodules inside that directory, so attempts to import the module will fail.

The \_\_init\_\_.py file is usually empty, but can be used to export selected portions of the package under more convenient names, hold convenience functions, etc. Given the example above, the contents of the \_\_init\_\_ module can be accessed as

import spam

**Context.py**: To Switch Between the Context to Access the DOM

@Property: Points to the Class Property defined in Builtin.py

**LibraryComponent.Py**

Inheriting from Context Aware Class

Defined some methods logging and assertions

1. **Keywords** 
   1. **Init**
   2. **Alert.py: To Handle the Alerts in the page** 
      1. **from** selenium.common.exceptions **import** WebDriverException  
         **from** selenium.webdriver.support **import** expected\_conditions **as** EC  
         **from** selenium.webdriver.support.ui **import** WebDriverWait  
           
         **from** SeleniumLibrary.base **import** keyword, LibraryComponent  
         **from** SeleniumLibrary.utils **import** is\_truthy, secs\_to\_timestr
      2. Class AlertKeywords
      3. Methods
         1. input\_text\_into\_prompt
         2. input\_text\_into\_alert
         3. alert\_should\_be\_present
         4. alert\_should\_not\_be\_present
         5. handle\_alert
   3. **BrowserManagement.py:** 
      1. **Open Browser**
      2. **Close Browser**
      3. **Switch Browser**
   4. Element.py
   5. formElement.py
   6. Frames.py
   7. Waiting.py
   8. Screenshot.py

**Excel Library:**

We Will look into the Excel Library Definition