| **Term** | **Definition** |
| --- | --- |
| after\_all() | Any code you place in this test fixture will execute once after all the features. |
| after\_feature() | Any function in this test fixture will run after each feature. |
| after\_scenario() | Any code you place in this test fixture will execute once after each scenario. |
| after\_step() | Any code you place in this test fixture will execute once after each step. |
| after\_tag() | Any code you place in this test fixture will execute once after each tag. |
| And | To make a series more readable and fluid, you could use And keywords to replace each repeated keyword after the first step in the series. |
| API | Application program interface. |
| Background | A test fixture that you can use to specify the context once and then establish it before every scenario in the feature. |
| BASE\_URL | An environment variable that tells you the location of the system under test. |
| before\_all() | Any code you place in this test fixture will execute once before all the features. |
| before\_feature() | Any function in this test fixture will run before each feature. |
| before\_scenario() | Any code you place in this test fixture will execute once before each scenario. |
| before\_step() | Any code you place in this test fixture will execute once before each step. |
| before\_tag() | Any code you place in this test fixture will execute once before each tag. |
| BDD | Behavior driven development focuses on the system's behavior as observed from the outside in. |
| Concordion | An open-source BDD tool for the Java framework and uses normal language using paragraphs for specifications. |
| context.driver.quit() | This function tells the web browser to close down after all tests are complete, ensuring you don't have multiple web browsers running in memory afterward. |
| context.table | Variable to load table data from an array of dictionaries. |
| Cucumber | The oldest BDD tool and uses the Gherkin syntax for specifications. |
| Feature | In a Gherkin document, the first keyword is usually Feature, followed by a colon and the title. This is a syntax that most agile teams use when they define their user stories. |
| Features folder | Contains all the files that control Behave with an extension of .feature and steps files in a subfolder. |
| getenv() | Imports any configuration parameters from the Environment. |
| Gherkin | The most used syntax in BDD. |
| Given keyword | A set of preconditions. These are the conditions required to put the system into the state it needs to be to perform the tests. |
| Integration testing | At this level, you're combining individual units and testing them as a group to expose faults in the interactions between the units. |
| Scenario | A situation that describes a single behavior of a feature, and you use the Given, When, Then syntax to write that description. |
| Selenium | A collection of tools for automating web browser activity. |
| Specification | In Gherkin, it is a description of how the system should behave in a situation. |
| Steps folder | A collection of Python files with steps matching the Gherkin statements in the feature files. |
| step\_impl() | A function to implement the step. |
| System testing | At this level, you're testing the complete system end to end. The purpose is to evaluate the system's compliance with the specified high-level requirements and ensure the whole system works together. |
| TDD | Test driven development focuses on the internal workings of the system. |
| Then keyword | Some testable outcomes are observed. This is the expected outcome of the action that the user performs. |
| Unit testing | At this level of the software testing process, you test individual units or components of a software system. |
| WAIT\_SECONDS | An environment variable that controls how long Selenium waits for a response from the UI. |
| When keyword | When an event occurs. These events are the actions that the user takes to interact with the system under test. |