**Practical No: - 09**

**AIM: -** To Study matching patterns in selenium.

**DESCRIPTION: -**

**MATCHING TEXT PATTERNS**

* In some test cases, QA Engineers need to verify text or number patterns by using **verifyTextPresent**, **verifyTitle**, **verifyAlert**, **assertConfirmation**, **verifyText**, and **verifyPrompt**.
* Selenium locators can utilize pattern to describe special characters, and dynamic texts to verify what text is expected rather than having to specify that text exactly.
* Patterns allow you to describe, via the use of special characters, what text is expected rather than having to specify that text exactly.
* There are three types of patterns:
  + globbing,
  + regular expressions, and
  + exact

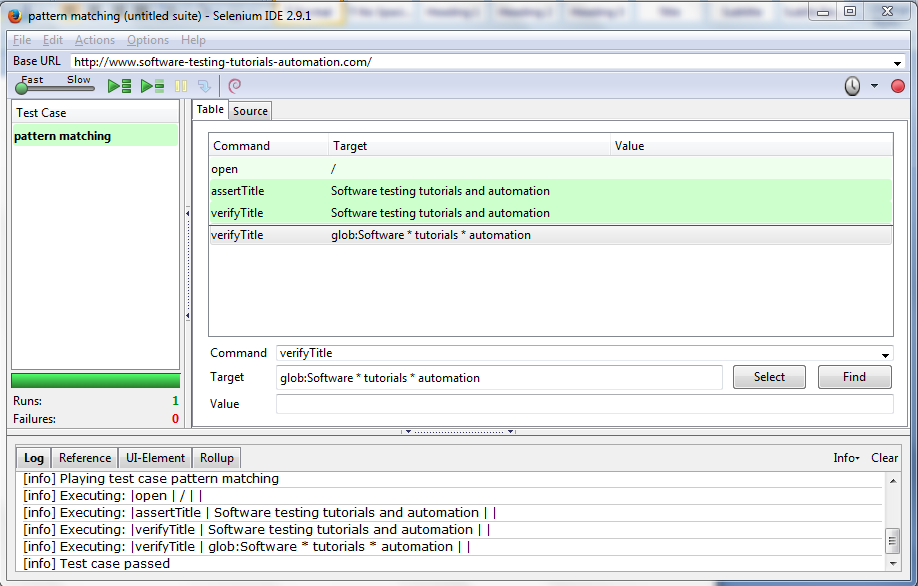
**MATCHING TEXT PATTERNS - GLOBBING PATTERNS**

* Most people are familiar with globbing as it is utilized in filename expansion at a DOS or Unix/Linux command line such as ls \*.c.
* In this case, globbing is used to display all the files ending with a .c extension that exist in the current directory.
* Globbing is fairly limited. Only two special characters are supported in the Selenium implementation:

**\*** - which translates to “match anything,” i.e., nothing, a single character, or many characters.

**[ ]** - (character class) which translates to “match any single character found inside the square brackets.”

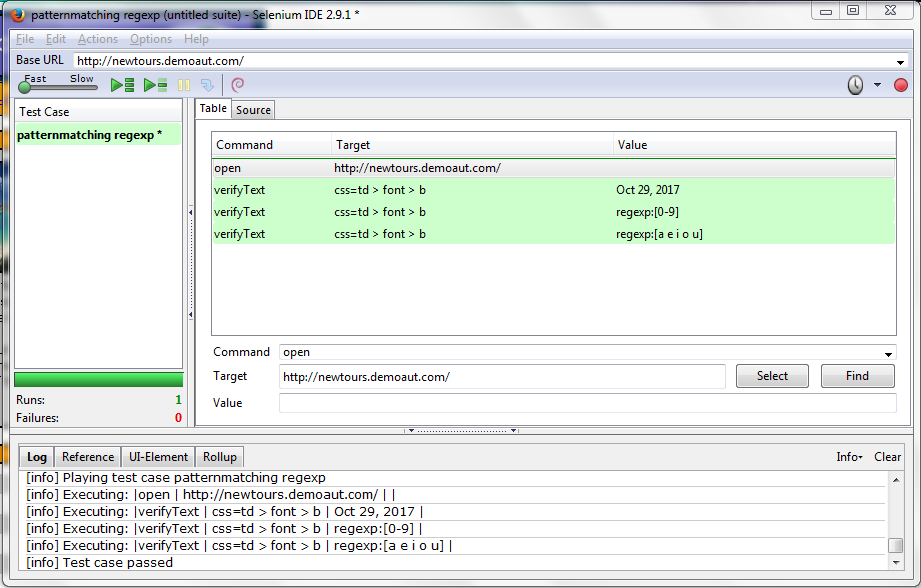
* To specify a globbing pattern parameter for a Selenese command, you can prefix the pattern with a **glob:** label.
* However, because globbing patterns are the default, you can also omit the label and specify just the pattern itself.
  + **\*** which translates to “match anything,” i.e., nothing, a single character, or many characters.
  + **[ ]** which translates to “match any single character found inside the square brackets.
  + **[aeiou]** matches any lowercase vowel
  + **[0-9]** matches any digit
  + **[a-zA-Z0-9]** matches any alphanumeric character



**MATCHING TEXT PATTERNS – REGULAR EXPRESSIONS**

* Regular expression patterns are the most powerful of the three types of patterns that Selenium supports.
* Regular expressions are also supported by most high-level programming languages, many text editors, and a host of tools,
* In Selenium, regular expression patterns allow a user to perform many tasks that would be very difficult otherwise.
* For example, suppose your test needed to ensure that a particular table cell contained nothing but a number. **regexp: [0-9]+** is a simple pattern that will match a decimal number of any length**.**
* Regular expression patterns in Selenium need to be prefixed with either **regexp:** or **regexpi:**
* The former is case-sensitive; the latter is case-insensitive.

|  |  |
| --- | --- |
| **PATTERN** | **MATCH** |
| **.** | any single character |
| **[ ]** | character class: any single character that appears inside the brackets |
| **\*** | quantifier: 0 or more of the preceding character (or group) |
| **+** | quantifier: 1 or more of the preceding character (or group) |
| **?** | quantifier: 0 or 1 of the preceding character (or group) |
| **{1,5}** | quantifier: 1 through 5 of the preceding character (or group) |
| **|** | alternation: the character/group on the left or the character/group on the right |
| **( )** | grouping: often used with alternation and/or quantifier |



**MATCHING TEXT PATTERNS – EXACT PATTERN**

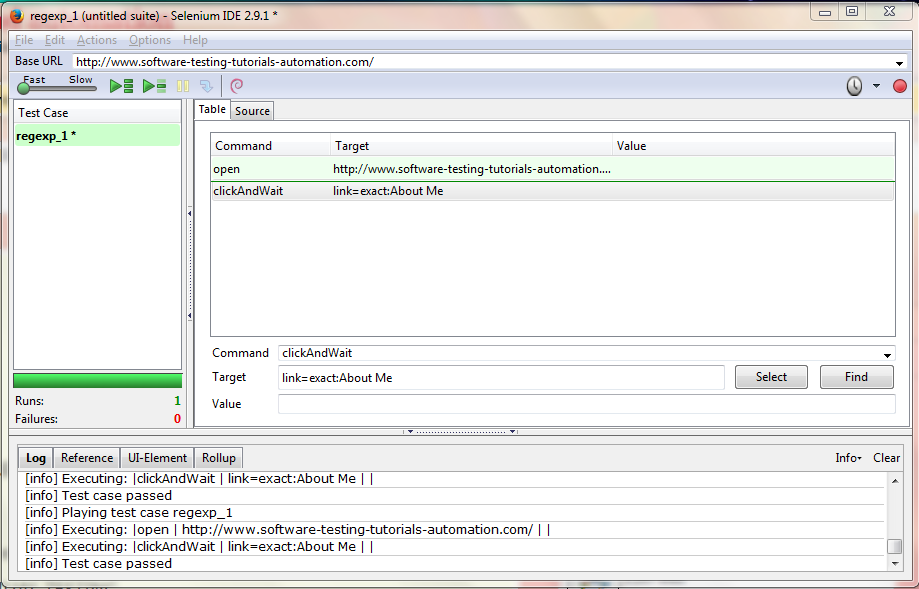
* Exact pattern means that users need to verify the exact texts or numbers. It uses no special characters at all.
* Patterns with the prefix ‘exact:’ will match the given text as it is.

**Command Target Value**

clickAndWait link=search

verifyValue exact: \*.java

* In this example the ‘\*’ (asterisk) will work as a normal character rather then a pattern-matching wildcard character.



**Below are a few common regular expression patterns:**

* **regexp:(0[1-9]|1[012])[- /.](0[1-9]|[12][0-9]|3[01])[- /.](19|20)\d\d**  
  : match a date in ‘mm/dd/yyyy’ format with any of the ‘-‘, ‘/’, ‘.’ as separators.
* **regexpi:^[A-Z0-9+\_.-]+@[A-Z0-9.-]+$**  
  : match a generic email address.
* **regexp:^[0-9]{5}(?:-[0-9]{4})?$**  
  :match a ZIP code (U.S. postal code), allowing both the five-digit and nine-digit (ZIP + 4) formats.
* **regexp:^(?!000|666)(?:[0-6][0-9]{2}|7(?:[0-6][0-9]|7[0-2]))-?(?!00)[0-9]{2}-(?!0000)[0-9]{4}$**  
  :match U.S Social Security numbers in the in the AAA-GG-SSSS format.
* **regexp:^(https?|ftp|file)://.+$**  
  :match almost any url.

**CONCLUSION: -** We have studied the matching patterns in selenium.