

11.S: REGULAR EXPRESSIONS (SUMMARY)



Contributed by [Chuck Severance](#)
Clinical Associate Professor (School of Information) at [University of Michigan](#)

While this only scratched the surface of regular expressions, we have learned a bit about the language of regular expressions. They are search strings with special characters in them that communicate your wishes to the regular expression system as to what defines "matching" and what is extracted from the matched strings. Here are some of those special characters and character sequences:

- `^` Matches the beginning of the line.
- `$` Matches the end of the line.
- `.` Matches any character (a wildcard).
- `\s` Matches a whitespace character.
- `\S` Matches a non-whitespace character (opposite of `\s`).
- `*` Applies to the immediately preceding character and indicates to match zero or more of the preceding character(s).
- `*?` Applies to the immediately preceding character and indicates to match zero or more of the preceding character(s) in "non-greedy mode".
- `+` Applies to the immediately preceding character and indicates to match one or more of the preceding character(s).
- `+?` Applies to the immediately preceding character and indicates to match one or more of the preceding character(s) in "non-greedy mode".
- `[aeiou]` Matches a single character as long as that character is in the specified set. In this example, it would match "a", "e", "i", "o", or "u", but no other characters.
- `[a-z0-9]` You can specify ranges of characters using the minus sign. This example is a single character that must be a lowercase letter or a digit.
- `[^A-Za-z]` When the first character in the set notation is a caret, it inverts the logic. This example matches a single character that is anything *other than* an uppercase or lowercase letter.
- `()` When parentheses are added to a regular expression, they are ignored for the purpose of matching, but allow you to extract a particular subset of the matched string rather than the whole string when using `findall()`.
- `\b` Matches the empty string, but only at the start or end of a word.
- `\B` Matches the empty string, but not at the start or end of a word.
- `\d` Matches any decimal digit; equivalent to the set `[0-9]`.
- `\D` Matches any non-digit character; equivalent to the set `[^0-9]`.