

Beyond the Textbook (Severance - Chapter 11)

Regular Expressions

Always Use Python Raw Strings For R.E. Patterns

- Using **raw strings** to hold R.E. patterns is a **best practice**.
- Use a raw string even if the pattern does not yet contain a `\`.
- This promotes consistency and easy maintenance of R.E. patterns.

Truthiness of Return Values From R.E. Functions

- `re.search()` returns a `match` object when search succeeds.
- `re.search()` returns `None` when search fails.
- `re.findall()` returns a populated list when find succeeds.
- `re.findall()` returns an empty list when find fails.
- This allows us to test the results of these function calls for `True` and `False`.
- While these functions return **truthy** values, they do not return boolean values.

Shorthand Character Classes

Expression	Class Includes
<code>\d</code>	Any digit [0-9]
<code>\D</code>	Not any digit
<code>\w</code>	Any <i>word character</i> [A-Za-z0-9_]
<code>\W</code>	Not any <i>word character</i>
<code>\s</code>	Any <i>whitespace character</i> (space, tab, return, newline, formfeed)
<code>\S</code>	Not any <i>whitespace character</i>

Forming Compound R.E. Patterns

- A simple logical `and` is implicit in patterns.
- A simple logical `or` can be expressed with the `|` (pipe symbol)
- More complex compound expressions may require the combination of R.E. statements with Python `and` and `or`.
- See:
 - `_10_forming_compound_re_patterns.py`

Use `re.IGNORECASE` Rather Than `str.lower()`

- The `re` functions take a `re.IGNORECASE` *flag* parameter that instructs the R.E. engine to ignore the difference between lowercase and upper case letters.
- This is better documentation than shifting the string with `lower()`.
- The text can be extracted using `re.findall()` with its case structure intact.
- See:
 - `_20_ignoring_case_using_str_lower.py`
 - `_30_ignoring_case_using_re_ignorecase.py`

Last Revised 2022-10-09