Week 7- Regular Expressions (regex)

regexes - A pattern.

re - A library in Python for regular expressions
↳ Find, check for, and or replace Patterns

Common (Versatile) re function
↳ re.search(pattern, string, flags=0)

Explained ↓

re.search - name of function

Pattern - (argument) pattern that you want to search for

String - The string you pass in for the Pattern to search in

flags - Allows you to modify the behavior of the regular expression

↳ ie flags = re.IGNORECASE # ~~ignor~~ Case insensitive
re.I # same ~~as~~ as but abbreviated
re.M or MULTILINE #
re.DOTALL #

Pattern Special Symbols

period → . - Any character except a newline

* - 0 or more repetitions

+ - 1 or more repetitions

? - 0 or 1 pepetition

{m} - m repetitions

{m,n} - m through n repetitions

^ - Matches the start of the string

$ - matches the end of the string or just before the newline at the end of the string

[ ] - set of characters to be allowed

[^] - Complementing the set, set of characters NOT allowed

\w - Any word character

* more on next page

\b - boundary between a \w and a \W character.

\d - decimal digit
\D - Not a decimal digit
\s - white space characters
\S - Not a whitespace character
\w - word characters, numbers, and underscores
\W - Not a word character
A | B - either A or B
(...) - a group
(?:...) - non-capturing Version # Group without assigning to a group
(?P<name>...) - Give group a name # Call with variable.group("name")

Substitute / Replace
re.sub(pattern, repl, string, Count = 0, flags = 0)

Split
re.split(pattern, string, maxsplit = 0, flags = 0)

Find
re.findall(pattern, string, flags = 0)