1. Passing a string value representing your regular expression to re.compile() returns a Regex pattern object.
2. Raw strings are particularly useful when working with regular expressions, as they allow you to specify patterns that may contain backslashes without having to escape them.
3. The search() method returns the index (position) of the first match. The search() method returns -1 if no match is found.
4. Match objects have a group() method that will return the actual matched text from the searched string.
5. Group 0 is the entire match, group 1 covers the first set of parentheses, and group 2 covers the second set of parentheses.
6. Periods and parentheses can be escaped with a backslash: \., \(, and \).
7. If the regex has no groups, a list of strings is returned. If the regex has groups, a list of tuples of strings is returned.
8. The | character signifies matching “either, or” between two groups.
9. which character ?not mentioned
10. The + matches one or more. The \* matches zero or more.
11. The {4} matches exactly three instances of the preceding group. The {4,5} matches between four and five instances.
12. The \d, \w, and \s shorthand character classes match a single digit, word, or space character, respectively.
13. The \D, \W, and \S shorthand character classes match a single character that is not a digit, word, or space character, respectively.
14. The .\* performs a greedy match, and the .\*? performs a nongreedy match.
15. Either [0-9a-z] or [a-z0-9].
16. Passing re.I or re.IGNORECASE as the second argument to re.compile() will make the matching case insensitive.
17. The . character normally matches any character except the newline character. If re.DOTALL is passed as the second argument to re.compile(), then the dot will also match newline characters.
18. 'X drummers, X pipers, five rings, X hens'
19. The re.VERBOSE argument allows you to add whitespace and comments to the string passed to re.compile()
20. e.compile(r'^\d{1,3}(,{3})\*$') will create this regex, but other regex strings can produce a similar regular expression.
21. re.compile(r'[A-Z][a-z]\*\sWatanabe')
22. re.compile(r'(Alice|Bob|Carol)\s(eats|pets|throws)\ s(apples|cats|baseballs)\.', re.IGNORECASE)