Answer 1

The re.compile() function returns Regex objects.

Answer 2

Raw strings are used so that backslashes do not have to be escaped.

Answer 3

The search() method returns Match objects.

Answer 4

The group() method returns strings of the matched text.

Answer 5

Group 0 is the entire match, group 1 covers the first set of parentheses, and group 2 covers the second set of parentheses.

Answer 6

Periods and parentheses can be escaped with a backslash: \., \(, and \).

Answer 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.

Answer 8

The | character signifies matching "either, or" between two groups.

Answer 9

The ? character can either mean "match zero or one of the preceding group" or be used to signify nongreedy matching.

Answer 10

The + matches one or more. The \* matches zero or more.

Answer 11

The {3} matches exactly three instances of the preceding group. The {3,5} matches between three and five instances.

Answer 12

The \d, \w, and \s shorthand character classes match a single digit, word, or space character, respectively.

Answer 13

The \D, \W, and \S shorthand character classes match a single character that is not a digit, word, or space character, respectively.

Answer 14

Passing re.I or re.IGNORECASE as the second argument to re.compile() will make the matching case insensitive.

Answer 15

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.

Answer 16

The . performs a greedy match, and the .? performs a nongreedy match.

Answer 17

Either [0-9a-z] or [a-z0-9]

Answer 18

'X drummers, X pipers, five rings, X hens'

Answer 19

The re.VERBOSE argument allows you to add whitespace and comments to the string passed to re.compile().

Answer 20

re.compile(r'^\d{1,3}(,\d{3})\*$') will create this regex, but other regex strings can produce a similar regular expression.

Answer 21

re.compile(r'[A-Z][a-z]\*\sNakamoto')

Answer 22

re.compile(r'(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.', re.IGNORECASE)