What is the function that creates Regex objects?

The re.compile() function returns Regex objects.

Why are raw strings often used when creating Regex objects?

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

What does the search() method return?

The search() method returns Match objects.

How do you get the actual strings that match the pattern from a Match object?

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

In the regex created from r'(\d\d\d)-(\d\d\d-\d\d\d\d)', what does group 0 cover? Group 1? Group 2?

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

Parentheses and periods have specific meanings in regular expression syntax. How would you specify that you want a regex to match actual parentheses and period characters?

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

The findall() method returns a list of strings or a list of tuples of strings. What makes it return one or the other?

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.

What does the | character signify in regular expressions?

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

What two things does the ? character signify in regular expressions?

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

What is the difference between the + and \* characters in regular expressions?

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

What is the difference between {3} and {3,5} in regular expressions?

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

What do the \d, \w, and \s shorthand character classes signify in regular expressions?

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

What do the \D, \W, and \S shorthand character classes signify in regular expressions?

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

How do you make a regular expression case-insensitive?

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

What does the . character normally match? What does it match if re.DOTALL is passed as the second argument to re.compile()?

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.

What is the difference between these two: .**and .**?

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

What is the character class syntax to match all numbers and lowercase letters?

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

If numRegex = re.compile(r'\d+'), what will numRegex.sub('X', '12 drummers, 11 pipers, five rings, 3 hens') return?

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

What does passing re.VERBOSE as the second argument to re.compile() allow you to do?

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

How would you write a regex that matches a number with commas for every three digits? It must match the following:  
  
'42'  
  
'1,234'  
  
'6,368,745'  
  
but not the following:  
  
'12,34,567' (which has only two digits between the commas)  
  
'1234' (which lacks commas)

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

How would you write a regex that matches the full name of someone whose last name is Nakamoto? You can assume that the first name that comes before it will always be one word that begins with a capital letter. The regex must match the following:  
  
'Satoshi Nakamoto'  
  
'Alice Nakamoto'  
  
'Robocop Nakamoto'  
  
but not the following:  
  
'satoshi Nakamoto' (where the first name is not capitalized)  
  
'Mr. Nakamoto' (where the preceding word has a nonletter character)  
  
'Nakamoto' (which has no first name)  
  
'Satoshi nakamoto' (where Nakamoto is not capitalized)

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

How would you write a regex that matches a sentence where the first word is either Alice, Bob, or Carol; the second word is either eats, pets, or throws; the third word is apples, cats, or baseballs; and the sentence ends with a period? This regex should be case-insensitive. It must match the following:  
  
'Alice eats apples.'  
  
'Bob pets cats.'  
  
'Carol throws baseballs.'  
  
'Alice throws Apples.'  
  
'BOB EATS CATS.'  
  
but not the following:  
  
'Robocop eats apples.'  
  
'ALICE THROWS FOOTBALLS.'  
  
'Carol eats 7 cats.'

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