1. What is the name of the feature responsible for generating Regex objects?  
   Solution: re.compile() function is responsible for generating Regex objects.

**2**. Why do raw strings often appear in Regex objects?

Solution: Raw strings are used so that backslashes do not have to be escaped.  
**3**. What is the return value of the search() method?  
Solution: The search() method returns the first match of a substring found in the string.  
**4**. From a Match item, how do you get the actual strings that match the pattern?  
Solution: The group() method returns strings of the matched text.  
**5**. In the regex which created from the r'(\d\d\d)-(\d\d\d-\d\d\d\d)', what does group zero cover? Group 2? Group 1?  
Solution: Group 0 covers the entire match, group 1 covers the first set of parentheses, and group 2 covers the second set of parentheses.  
**6**. In standard expression syntax, parentheses and intervals have distinct meanings. How can you tell a regex that you want it to fit real parentheses and periods?  
Solution: We can use backslash to escape parentheses and periods, like: \., \(, and \).  
**7.** The findall() method returns a string list or a list of string tuples. What causes it to return one of the two options?  
Solution: 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**. In standard expressions, what does the | character mean?  
Solution: In standard expressions, the pipe (|) character means bitwise or operation.  
**9**. In regular expressions, what does the | character stand for?  
Solution: It indicates that a match can be one of the two terms on either side of the pipe.  
**10**.In regular expressions, what is the difference between the + and \* characters?  
Solution: Each of them are regular expression meta characters, the asterisk character ( \* ) means that the preceding expression can match zero or more times while the plus character( + ) indicates that the preceding expression must match at least one time or multiple times.  
**11**. What is the difference between {4} and {4,5} in regular expression?  
Solution: {4} matches the previous token exactly 4 times as many times as possible while {4,5} matches the previous token between 4 and 5 times, as many times as possible.  
12. What do you mean by the \d, \w, and \s shorthand character classes signify in regular expressions?  
Solution: \d matches a digit (equivalent to [0-9]), \w matches any word character (it’s equivalent to [a-zA-Z0-9\_] and \s matches any whitespace character (equivalent to [\r\n\t\f\v ]  
13. What do means by \D, \W, and \S shorthand character classes signify in regular expressions?  
Solution: \D matches any character that's not a digit (equivalent to [^0-9]), \W matches any non-word character (equivalent to [^a-zA-Z0-9\_]) and \S matches any non-whitespace character (equivalent to [^\r\n\t\f\v ])  
14. What is the difference between .\*? and .+? ?  
Solution: .\*? matches any character except for line terminators between zero and unlimited times while .+? matches any character except for line terminators between one and unlimited times.  
  
  
15. What is the syntax for matching both numbers and lowercase letters with a character class?  
Solution: [a-z0-9]  
16. What is the procedure for making a normal expression in regex case insensitive?  
Solution: Writing the following expression (?i) before the expression that is to be made insensitive. E,g., (?i)[a-z] will match lowercase as well as upper case letters.  
17. What does the . character normally match? What does it match if re.DOTALL is passed as 2nd argument in re.compile()?  
Solution: 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. If numReg = re.compile(r’\d+’;), what will numRegex.sub(‘X’, ‘11 drummers, 10 pipers, five rings, 4 hen’) return?  
Solution: X drummers, X pipers, five rings, X hen  
19. What does passing re.VERBOSE as the 2nd argument to re.compile() allow to do?  
Solution: The re. VERBOSE argument allows you to add whitespace and comments to the string passed to re.  
20. How would you write a regex that match a number with comma for every three digits? It must match the given 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)  
Solution: r’\d{1,3}(,\d{3})\*$’  
21. How would you write a regex that matches the full name of someone whose last name is

Watanabe? 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:

‘Haruto Watanabe’

‘Alice Watanabe’

‘RoboCop Watanabe’

but not the following:

‘haruto Watanabe’ (where the first name is not capitalized)

‘Mr. Watanabe’ (where the preceding word has a nonletter character)

‘Watanabe’ (which has no first name)

‘Haruto watanabe’ (where Watanabe is not capitalized)  
  
Solution: r‘[A-Z]{1}[A-Za-z]\*\s[A-Z]{1}[A-Za-z]\*’  
22. 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.’

Solution: r’(?i)(alice|bob|carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.’