1. What is the name of the feature responsible for generating Regex objects?

re.complie()

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

because regular expression have "\" as syntax, like \d

3. What is the return value of the search() method?

match object to desired pattern

4. From a Match item, how do you get the actual strings that match the pattern?

by using group method

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?

group zero- \d\d\d-\d\d\d-\d\d\d\d

Group 2- (\d\d\d-\d\d\d\d)

Group 1 - (\d\d\d)

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?

using \ for escaping these characters.

7. The findall() method returns a string list or a list of string tuples. What causes it to return one of the two options?

groups

8. In standard expressions, what does the | character mean?

pipe, to match multiple expression

9. In regular expressions, what does the character stand for?

\ , ? , + , \* etc they are used denote pattern to be searched

10.In regular expressions, what is the difference between the + and \* characters?

+ searches for 1 or more instances of pattern

\* searches for zero or more.

11. What is the difference between {4} and {4,5} in regular expression?

{4} will search for 4 repititions

{4,5} will search for 4 or 5 repititions

12. What do you mean by the \d, \w, and \s shorthand character classes signify in regular expressions?

\d for digits

\w for all digits, alphbets, underscore

\s space, tab, new line

13. What do means by \D, \W, and \S shorthand character classes signify in regular expressions?

\D for all other than digits

\W for all other than digits, alphbets, underscore

\S for all other than space, tab, new line

14. What is the difference between .\*? and .\*?

first searches in non greedy manner, ie it searches shortest possible string

.\* searches in greedy manner ie it gives longest possible string.

15. What is the syntax for matching both numbers and lowercase letters with a character class?

\w

16. What is the procedure for making a normal expression in regax case insensitive?

re.compile(r'xyz',re.IGNORECASE)

17. What does the . character normally match? What does it match if re.DOTALL is passed as 2nd argument in re.compile()?

it will match any character except for new line.

it will also match new line character \n

18. If numReg = re.compile(r'\d+'), what will numRegex.sub('X', '11 drummers, 10 pipers, five rings, 4 hen') return?

'XX drummers, XX pipers, five rings, X hen'

19. What does passing re.VERBOSE as the 2nd argument to re.compile() allow to do?

it allow first arguement of re.complie to be written in multiple lines and thus make it easy to read.

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)

re.compile (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)

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

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.'

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