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

Ans: import re

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

Ans:

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

Ans: Search() is a method in re module which takes two args and returns the match object if found. Otherwise returns None.

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

Ans: re.search(‘match\_item’, ‘actual string’)

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?

Ans: group(0) cover the entire string

group(2) returns the 2nd part of the string that matches with the given pattern.  
Group(1) returns the 1st part of the string that matches with the given pattern.

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?

Ans: by using backslash ‘\’.

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

Ans: findall() function takes 2 arguments : 1. pattern and 2. strring  
if the given pattern is a simple string without using any parentheses then findall() will return a list of strings that matches the pattern.  
For example,

|  |  |
| --- | --- |
| Code | Output |
| import re  s = "black, blue and brown"  pattern = r'bl\w+'  matches = re.findall(pattern,s) print(matches) | ['black', 'blue'] |

But when the pattern has multiple groups using parentheses then the output will be list of tuples. for example,

|  |  |
| --- | --- |
| Code | Output |
| import re  s = "black, blue and brown"  pattern = r'(bl(\w+))'  matches = re.findall(pattern,s) print(matches) | [('black', 'ack'), ('blue', 'ue')] |

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

Ans:

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

Ans: in RegEx, ‘|’ stands for choice. It matches the expression before it or after it.

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

Ans: in RegEx, ‘+’ matches the preceding element one or more than one time.  
For example, ‘ab+c’ matches ‘abc’ or ‘abbc’ or ‘abbbc’ and so on but not ‘ac’.

While ‘\*’ matches the preceding element or group of elements one or more times.  
For example, (xy)\* matches ‘xy’, ‘xyxy’, ‘xyxyxy’, etc.

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

Ans: {4} matches the preceding character atleast 4 times where as {4,5} matches the preceding character atleast 4 times but not more than 5 times.

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

Ans: \d stands for digits from 0 to 9.  
\w stands for alphabets from a to z in lowercase  
\s stands for whitespace.

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

Ans: \D stands for not a digit  
\W stands for not a word  
\S not a wwhitespace

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

Ans: .\*? = any character zero or more times.  
  
.+? = any character one or more times.

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

Ans: [a-z0-9] or [0-9a-z]

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

Ans: re.IGNORECASE or re.I as the 2nd argument to re.compile().

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

Ans: the .character usually matches any character except newline character. If re.DOTALL is passed as the 2nd 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?

Ans: Output : X drummers, X pipers, five rings, X hen

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

Ans: re.VERBOSE argument allows us to add whitespace and comments to the string passed to re.compile().  
For example,  
regex\_email = re.compile(r"""  
 ^([a-z0-9\_\.-]+) # local Part  
 @ # single @ sign  
 ([0-9a-z\.-]+) # Domain name  
 \. # single Dot .  
 ([a-z]{2,6})$ # Top level Domain  
 """,re.VERBOSE | re.IGNORECASE)

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)

Ans: x = 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)

Ans: 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.'

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