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

Ans: re.compile() function

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

Ans: Raw strings (using the r prefix) are often used in Regex objects to avoid unwanted escape characters. Since regular expressions often include backslashes to represent special characters, using a raw string ensures that backslashes are treated as literal characters and not as escape characters.

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

Ans: The return value of the search() method is a Match object if a match is found, or None if no match is found.

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

Ans: You can get the actual strings that match the pattern from a Match item using the ‘.group()’ method. For example, ‘match.group(0)’ returns the entire matched string, and ‘match.group(1)’ returns the string that matches the first group in the regex.

**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: In the regex ‘r'(\d\d\d)-(\d\d\d-\d\d\d\d)' ‘, group zero covers the entire matched string. Group 1 covers the first set of three digits, and group 2 covers the second set of three digits followed by a hyphen and the four digits after it.

**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: To match literal parentheses and periods in a regex, you need to escape them with a backslash. For example, ‘\(‘ matches a literal opening parenthesis, and ‘\.’ matches a literal period.

**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: The ‘findall()’ method returns a list of strings when the regex does not have any capturing groups. It returns a list of tuples containing strings when the regex has capturing groups.

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

Ans: In standard expressions, the ‘|’ character means "OR". It allows you to specify multiple alternatives in your regex pattern.

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

Ans: any character except a newline

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

Ans: In regular expressions, the + character means "one or more occurrences," while the \* character means "zero or more occurrences."

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

Ans: {4} in a regular expression matches exactly four occurrences of the preceding element. {4,5} matches between four and five occurrences of the preceding element.

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

Ans: In regular expressions, \d represents any digit (equivalent to [0-9]), \w represents any word character (equivalent to [a-zA-Z0-9\_]), and \s represents any whitespace character (spaces, tabs, newlines).

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

Ans: \D represents any non-digit character, \W represents any non-word character, and \S represents any non-whitespace character in regular expressions.

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

Ans: .\*? is a non-greedy qualifier that matches as few characters as possible, while .\* is a greedy qualifier that matches as many characters as possible.

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

Ans: The syntax for matching both numbers and lowercase letters with a character class is [0-9a-z] or [a-z0-9].

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

Ans: To make a regular expression case-insensitive, you can pass the ‘re.IGNORECASE’ flag (or ‘re.I’ shorthand) as the second 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 normally matches any character except a newline. If re.DOTALL is passed as the second argument to re.compile(), the . character 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: '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: Passing re.VERBOSE as the second argument to re.compile() allows you to include whitespace and comments within the regular expression to improve its readability.

**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: You can use the regex pattern r'^\d{1,3}(,\d{3})\*$' to match numbers with commas for every three digits.

**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: The regex pattern for matching the full name of someone whose last name is Watanabe with a capitalized first name is r'[A-Z][a-zA-Z]\* Watanabe'.

**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: The regex pattern for the specified sentence structure is r'^(Alice|Bob|Carol) (eats|pets|throws) (apples|cats|baseballs)\.$' with the re.IGNORECASE flag.