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

**Ans: The re module is responsible for generating Regex objects.**

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

**Ans: Raw strings are used in Regex objects because they allow backslashes to be interpreted literally instead of as escape characters.**

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

**Ans: The search() method returns a Match object if the pattern is found in the string, and None if the pattern is not found.**

1. **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 by calling the group() method on the Match object.**

1. **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 covers the entire match. Group 1 covers the first set of parentheses, which matches the first three digits. Group 2 covers the second set of parentheses, which matches the next three digits followed by a dash and the last four digits.**

1. **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 real parentheses and periods in a regex, you need to escape them using a backslash.**

1. **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 string tuples when the regular expression has more than one group. Otherwise, it returns a list of strings.**

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

**Ans: The | character means "or" in standard expressions. It matches either the expression before it or the expression after it.**

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

**Ans: The dot (.) character stands for any character except for a newline.**

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

**Ans The + character matches one or more occurrences of the preceding expression, while the \* character matches zero or more occurrences of the preceding expression.**

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

**Ans: {4} matches exactly four occurrences of the preceding expression, while {4,5} matches between four and five occurrences of the preceding expression.**

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

**Ans: \d matches any digit character, \w matches any word character (alphanumeric characters and underscores), and \s matches any whitespace character.**

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

**Ans: \D matches any non-digit character, \W matches any non-word character, and \S matches any non-whitespace character.**

1. **What is the difference between .? and .?**

**Ans: .\* matches any character (except for a newline) zero or more times, while .\*? matches any character (except for a newline) zero or more times, but as few times as possible.**

1. **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].**

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

**Ans: To make a normal expression in regex case insensitive, you can pass the re.IGNORECASE flag as the second argument to re.compile().**

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

**Answer: The "." character in regular expressions normally matches any character except for a newline character. If the re.DOTALL flag is passed as the second argument to re.compile(), then the "." character will match any character including a newline.**

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

**Answer: The code will replace any match of one or more digit characters with the string "X". Therefore, the returned string will be "X drummers, X pipers, five rings, X hen".**

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

**Answer: Passing re.VERBOSE as the second argument to re.compile() allows the use of whitespace and comments in the regular expression pattern, which can make the pattern more readable and easier to understand.**

1. **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)**

**Answer: The regular expression that matches a number with commas for every three digits is: r'^\d{1,3}(,\d{3})\*$'. This pattern matches any string that starts with one to three digits, followed by zero or more occurrences of a comma and three digits. It does not match strings with more or less than three digits between commas or without any commas.**

1. **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)**

**Answer: The regular expression that matches the full name of someone whose last name is Watanabe is: r'[A-Z][a-zA-Z]\*\sWatanabe'. This pattern matches any string that starts with a capitalized word, followed by a space and the word "Watanabe". It does not match strings with a non-letter character preceding "Watanabe", strings without a first name, or strings where "Watanabe" is not capitalized.**

1. **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.'**

**Answer: The regular expression that matches the described sentence pattern is: r'^(Alice|Bob|Carol)\s(eats|pets|throws)\s**