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

re.compile() generates the regex objects

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

It is because the backslashes have to escape from escape charaters.

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

It returns a match if there is a match within the string else it return none.

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

We will get the actual string by using the following code:

re.match(r“string”,line).group[0]

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?**

Group 0: Returns 3 numbers from 0 to 9

Group 2: returns 4 numbers from 0 to 9

Group 1: returns 3 numbers from 0 to 9

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?**

By using “r” before the syntax of regex. Here, r specifies a “raw string”.

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

If only one group is present in the pattern, then it will return any one of them.

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

This means “OR”

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

The character stands for a single word in a group of words. For example: in “apple”, characters are ‘a’, ’p’, ’p’, ’l’, ’e’

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

If we write “+” in the line with any regex symbol, then it means that there can be other characters, while if we use “\*” symbol it means that there can be zero or more same characters of words.

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

{4} this means only 4

And, {4,5} means number 4 and 5 both

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

\d means that numbers from 0 to 9

\w means characters including a to z, A to Z, 0-9, \_ i.e. Alphanumeric numbers.

\s means non white space characters

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

\D means non-numeric

\W means only symbols like $, %, #, etc.

\S means white space characters.

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

.\*? will find only first match and return the same while .\*will find till the end of the word.

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

\w

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

We can use (?!)G[a-z](?!).\* or [gG][a-bA-B].\* to make a part of regex case insensitive

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

. character matches any character except the newline character, and if re.DOTALL is passed as 2nd argument then it will include all the characters including the newline characters.

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

This will return X when a number is encountered

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

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

re.VERBOSE allows the regex to look more readable.

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)**

(^\d{1-3}(,\d{3})\*$)

Here, ^ tells that the number may have 1-3 characters and thereafter 3 characters there should be a comma and after that there should be 3 characters or more in the pair of 3.

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)**

((^[A-Z][a-z]\*$)(\S^[Watnable]))

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.'**

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