**Assignment-7**

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

Sol: The re.compile() function is responsible for generating Regex objects.

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

Sol: Raw strings are used in Regex objects to avoid Python's interpretation of escape characters like \n and \t. This is because escape characters may be used in Regex patterns, but they have a different meaning than in Python strings.

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

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

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

Sol: You can get the actual strings that match the pattern from a match object using the group() method or the groups() 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?

Sol: Group zero (\0) covers the entire match, group 1 (\1) covers the first set of parentheses, and group 2 (\2) covers the second set of parentheses.

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?

Sol: To fit real parentheses and periods in a regex pattern, you can use a backslash (\) to escape the character. For example, to match a literal left parenthesis, you can use \(, and to match a literal period, you can use \..

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

Sol: The findall() method returns a list of all non-overlapping matches of the pattern in the string. If the pattern contains groups, then it returns a list of tuples, where each tuple contains the matching groups.

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

Sol: The | character in regular expressions is used to indicate alternation, meaning "either this or that". It matches either the expression before or after the |.

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

Sol: In regular expressions, the . character is a wildcard that matches any character except a newline character (\n).

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

Sol: In regular expressions, the + character matches one or more occurrences of the preceding character or group, while the \* character matches zero or more occurrences of the preceding character or group.

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

Sol: {4} matches exactly 4 occurrences of the preceding character or group, while {4,5} matches between 4 and 5 occurrences of the preceding character or group.

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

Sol: In regular expressions, \d matches any digit character (equivalent to [0-9]), \w matches any alphanumeric character (equivalent to [a-zA-Z0-9\_]), and \s matches any whitespace character (equivalent to [ \t\n\r\f\v]).

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

Sol: In regular expressions, \D matches any non-digit character (equivalent to [^0-9]), \W matches any non-alphanumeric character (equivalent to [^a-zA-Z0-9\_]), and \S matches any non-whitespace character (equivalent to [^ \t\n\r\f\v]).

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

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

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

Sol: To match both numbers and lowercase letters with a character class, you can use the expression [0-9a-z] or [a-z0-9].

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

Sol: To make a regular expression case-insensitive, you can pass the re.IGNORECASE or re.I flag 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()?

Sol: The . character in regular expressions normally matches any character except a newline character (\n). If re.DOTALL is passed as the second argument to re.compile(), then the . character will match any character, including newline characters.

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

Sol: The numRegex.sub('X', '11 drummers, 10 pipers, five rings, 4 hen') call will return the string 'X drummers, X pipers, five rings, X hen'.

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

Sol: The numRegex.sub('X', '11 drummers, 10 pipers, five rings, 4 hen') will return the string 'X drummers, X pipers, five rings, X hen'. The sub() method of a regular expression object replaces all occurrences of the pattern with the given replacement string, in this case, 'X'. The pattern r'\d+' matches one or more digit characters, so the method replaces all numbers in the input string with 'X'.

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)

Sol: The regex that matches a number with a comma for every three digits is: r'^\d{1,3}(,\d{3})\*$'. This regex matches:

* ^\d{1,3}: One to three digits at the beginning of the string.
* (,\d{3})\*: Zero or more groups of a comma followed by exactly three digits.
* $: End of the string.

This regex ensures that there is at least one digit at the beginning of the string, and then allows for any number of comma-separated groups of 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)

Sol: The regex that matches the full name of someone whose last name is Watanabe is: r'[A-Z][a-z]\* Watanabe'. This regex matches:

* [A-Z][a-z]\*: A capital letter followed by any number of lowercase letters, representing the first name.
* Watanabe: The last name.

This regex ensures that the first name begins with a capital letter and is followed by any number of lowercase letters, and that the last name is always "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.'

Sol: The 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 either apples, cats, or baseballs; and the sentence ends with a period is: r'^(Alice|Bob|Carol) (eats|pets|throws) (apples|cats|baseballs)\.$'. This regex matches:

* ^(Alice|Bob|Carol): The beginning of the string followed by one of the three names.
* (eats|pets|throws): One of the three verbs.
* (apples|cats|baseballs): One of the three objects.
* \.$: A period at the end of the string.