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

**Answer:**  There is no feature in python for generating Regex objects

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

**Answer:**  For two reasons : 1.Disambiguating backlashes 2.Clarity and Maintainability

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

**Answer:** It returns **Match**  object if it finds match for regular expression otherwise return **None**

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

**Answer:** 1. Group() method 2.Using attribute access

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?

**Answer:**  Suppose the number is “123-456-7890”

Group 0 shows all the matched digits. i.e. “123-456-7890”

Group 1 shows 1st 3 matched digits i.e. “123”

Group 2 shows 2nd set of matched digits enclosed within parenthesis and the hyphen. I.e. “456-7890”

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?

**Answer:** Just use backlash . Suppose, to match lateral opening parenthesis( ( ), we use \( in our regex pattern. Similarly, to match lateral period (.), we use \. In regex pattern

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

**Answer:** If your regular expression pattern **does not contain any capturing groups** (parentheses), findall() will return a list of strings.

If your regular expression pattern includes **one or more capturing groups** (parentheses), findall() will return a list of tuples

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

**Answer:** either or

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

**Answer :** by using “.” Dot symbol denotes character except newline character

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

**Answer:** + is used for when to get character one or more occurrence where \* is for zero or more occurence

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

**Answer:** {4} preceding elements match for 4 times where {4,5} it will be 4 or 5 times

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

**Answer:** “\d” matches only decimal digits. “\w” matches word character. “\s” matches whitespace character

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

**Answer:** “\D” matches no digit character, “\W” matches not a word character, “\S” matches not a whitespace character

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

**Answer: “.\*”** matches any sequence of character(one or more occurrence) where “.\*?” is a non-greedy method where it tries to match fewest characters

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

**Answer:** [a-z0-9]

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

**Answer: 1.** Passing re.IGNORANCE flag 2. Using (?i) modifier

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

**Answer: “.”** Matches single character except newline character but when we use re.DOTALL flag as second argument then it also matches newline character

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

**Answer:** 'X drummers, X pipers, five rings, X hen'

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

**Answer:** To write more readable and maintainable regular expression

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)

**Answer:** r”^\d{1,3}(,\d{3})\*$”

import re

def is\_valid\_number(number\_str):

  pattern = r"^\d{1,3}(,\d{3})\*$"

  return bool(re.match(pattern, number\_str))

test\_cases = [

  "42",

  "1,234",

  "6,368,745",

  "12,34,567",

  "1234"

]

for case in test\_cases:

  if is\_valid\_number(case):

    print(f"'{case}' is a valid number with commas.")

  else:

    print(f"'{case}' is not a valid number with commas.")

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)

**Answer:** r”^[A-Z][a-zA-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.'

**Answer:** r”(?i)^(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|baseballs|cats)\.$”