**Practice Questions**

|  |  |
| --- | --- |
| Q: | 1. What is the function that creates Regex objects?   re.complie() |
| Q: | 1. Why are raw strings often used when creating Regex objects?   There are character escape, special character in the string that may affect the string so we use raw string to avoid those special charaters |
| Q: | 1. What does the search() method return?   Return the Match object |
| Q: | 1. How do you get the actual strings that match the pattern from a Matchobject?   Using group() method |
| Q: | 1. In the regex created from r'(\d\d\d)-(\d\d\d-\d\d\d\d)', what does group 0cover? Group 1? Group 2?   Group 0 will cover entire the match. |
| Q: | 1. Parentheses and periods have specific meanings in regular expression syntax. How would you specify that you want a regex to match actual parentheses and period characters?   Using back slash before parenthes and period: ex \( or \) or \. |
| Q: | 1. The findall() method returns a list of strings or a list of tuples of strings. What makes it return one or the other?   findall() method returns a list of string when there is no group in regex such as  re.compile(r'\d\d\d-\d\d\d-\d\d\d\d')  finall() method returns a tuples when there is the group in regext such as  re.compile(r'(\d\d\d)-(\d\d\d)-(\d\d\d\d)') |
| Q: | 1. What does the | character signify in regular expressions?   This character is called pipe , this will return the first match string in regex |
| Q: | 1. What two things does the ? character signify in regular expressions?   This is the optional matching -> will find the match or not for the optional matching string |
| Q: | 1. What is the difference between the + and \* characters in regular expressions?   The \* mean match zero or more  The + mean match one or more |
| Q: | 1. What is the difference between {3} and {3,5} in regular expressions?   The {3} mean at least 3 times matching  The {3,5} mean matching from 3-5 times |
| Q: | 1. What do the \d, \w, and \s shorthand character classes signify in regular expressions?   \d match if the character is number  \w match if the character is letter, number or underscore character  \s match if the character is space , tab or newline |
| Q: | 1. What do the \D, \W, and \S shorthand character classes signify in regular expressions?   \D match if the character is not number  \W match if the character is not letter, number or underscore character  \S match if the character is not space, tab or newline |
| Q: | 1. How do you make a regular expression case-insensitive?   We could use re.IGNORECASE or re.I as second argument of the regex. |
| Q: | 1. What does the . character normally match? What does it match if re.DOTALLis passed as the second argument to re.compile()?   . character is called wildcard -> will match any character except newline |
| Q: | 1. What is the difference between these two: .\* and .\*?   The .\* is greedy mode which try to match as much as we can  The .\*? is non-greedy mode which try to match the shortest string if possible |
| Q: | 1. What is the character class syntax to match all numbers and lowercase letters?   [a-z0-9] |
| Q: | 1. If numRegex = re.compile(r'\d+'), what will numRegex.sub('X', '12 drummers, 11 pipers, five rings, 3 hens') return?   X drummers, X pipers, five rings, X hens |
| Q: | 1. What does passing re.VERBOSE as the second argument to re.compile() allow you to do?   To allow the regex ignore white space and comment on pattern |
| Q: | 20. How would you write a regex that matches a number with commas for every three digits? It must match the 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)   Re.compile(r’^\d{1,3}(,\d{3})\*$’) |
| Q: | 21. How would you write a regex that matches the full name of someone whose last name is Nakamoto? You can assume that the first name that comes before it will always be one word that begins with a capital letter. The regex must match the following:   * 'Satoshi Nakamoto' * 'Alice Nakamoto' * 'Robocop Nakamoto'   but not the following:   * 'satoshi Nakamoto' (where the first name is not capitalized) * 'Mr. Nakamoto' (where the preceding word has a nonletter character) * 'Nakamoto' (which has no first name) * 'Satoshi nakamoto' (where Nakamoto is not capitalized)   Re.compile(r’^[A-Z].\*[^\W]\sNakamoto$’) |
| Q: | 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.' |

Re.compile(r’(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs).$’,re.I)