

## Python Advance Assignment\_17

Name : Bhavik Modi

E-mail : bmodi700@gmail.com

Website : <https://keenbm.github.io/>

Q1. Explain the difference between greedy and non-greedy syntax with visual terms in as few words as possible. What is the bare minimum effort required to transform a greedy pattern into a non-greedy one? What characters or characters can you introduce or change?

Answer- (1) Greedy Technique -The usual rule for matching in REs is sometimes called “left-most longest”: when a pattern can be matched at more than one place within a string, the chosen match will be the one that starts at the earliest possible position within the string, and then extends as far as possible. Normally Perl pattern matching is greedy. By greedy, we mean that the parser tries to match as much as possible. (2) In greedy approach regex pattern tend to consume maximum characters in a source string

Non-Greedy Technique -(1) A non-greedy match means that the regex engine matches as few characters as possible—so that it still can match the pattern in the given string. ... In other words, the non-greedy quantifiers give you the shortest possible match from a given position in the string. (2) In Non Greedy Technique regex pattern tend to consume minimum character in a source technique.

Q2. When exactly does greedy versus non-greedy make a difference? What if you're looking for a non-greedy match but the only one available is greedy?

Answer - Answer- (1) Greedy Technique -The usual rule for matching in REs is sometimes called “left-most longest”: when a pattern can be matched at more than one place within a string, the chosen match will be the one that starts at the earliest possible position within the string, and then extends as far as possible. Normally Perl pattern matching is greedy. By greedy, we mean that the parser tries to match as much as possible. (2) In greedy approach regex pattern tend to consume maximum characters in a source string

Non-Greedy Technique -(1) A non-greedy match means that the regex engine matches as few characters as possible—so that it still can match the pattern in the given string. ... In other words, the non-greedy quantifiers give you the shortest possible match from a given position in the string. (2) In Non Greedy Technique regex pattern tend to consume minimum character in a source technique.

If non-greedy available then we can use greedy with minimum number of qualifire character by manipulating parent String.

Q3. In a simple match of a string, which looks only for one match and does not do any replacement, is the use of a nontagged group likely to make any practical difference?

Answer - In such scenario it wont make any difference

Q4. Describe a scenario in which using a nontagged category would have a significant impact on the program's outcomes.

Answer - Non\_tagged is used where system is unable to use remember the scenario, basically when scenario is non-repetative.

Q5. Unlike a normal regex pattern, a look-ahead condition does not consume the characters it examines. Describe a situation in which this could make a difference in the results of your programme.

Lookahead and lookbehind, collectively called “lookaround”, are zero-length assertions just like the start and end of line, and start and end of word anchors explained earlier in this tutorial. The difference is that lookaround actually matches characters, but then gives up the match, returning only the result: match or no match. That is why they are called “assertions”. They do not consume characters in the string, but only assert whether a match is possible or not. Lookaround allows you to create regular expressions that are impossible to create without them, or that would get very longwinded without them.

Q6. In standard expressions, what is the difference between positive look-ahead and negative look-ahead?

Answer - Positive lookahead: `(?= «pattern»)` matches if pattern matches what comes after the current location in the input string. Negative lookahead: `(?! «pattern»)` matches if pattern does not match what comes after the current location in the input string.

Q7. What is the benefit of referring to groups by name rather than by number in a standard expression?

Answer - This is followed by an optional character and a space. The third named group is year. This consists of 1 or more digits. The advantage to named groups is that it adds readability and understandability to the code, so that you can easily see what part of a regular expression match is being referenced.

Q8. Can you identify repeated items within a target string using named groups, as in "The cow jumped over the moon"?

Answer - Yes we can identify repeated items within a target string using named groups.

Q9. When parsing a string, what is at least one thing that the Scanner interface does for you that the `re.findall` feature does not?

Answer - `re.search()` `re.search()` method either returns `None` (if the pattern doesn't match), or a `re.MatchObject` that contains information about the matching part of the string. This method stops after the first match, so this is best suited for testing a regular expression more than extracting data.

`re.findall()` Return all non-overlapping matches of pattern in string, as a list of strings. The string is scanned left-to-right, and matches are returned in the order found.

Q10. Does a scanner object have to be named scanner?

Answer - Yes scanner class need to be named as scanner.