1. The Main difference between Greedy and Non Greedy Match Syntax is The Greedy Match will try to match as many repetitions of the quantified pattern as possible Whereas The Non Greedy Match will try to match as few repetitions of the quantified pattern as possible

import re

print(re.findall("v\*", "vvvvvv")) # Greedy Match Synatx

print(re.findall("v\*?", "vvvvvv")) # Non Greddy Syntax

['vvvvvv', '']

['', 'v', '', 'v', '', 'v', '', 'v', '', 'v', '', 'v', '']

1. The Greedy Match will try to match as many repetitions of the quantified pattern as possible. The Non Greedy Match will try to match as few repetitions of the quantified pattern as possible. If only Non Greedy Match is available, we can use other filtering or pattern matching methods of regex and further identify the required pattern.
2. In this Case The Non Tagged Group will not make any diffrence in this case.

import re

phoneNumRegex = re.compile(r'\d\d\d')

num = phoneNumRegex.search('My number is 234-567-8901.')

print(f'Phone number found -> {num.group()}') # Non Tagged group

print(f'Phone number found -> {num.group(0)}') # Tagged Group

Phone number found -> 234

Phone number found -> 234

1. Here in the below Code Snippet . decimal is not tagged or captured. It will useful in scenarios where the separator of value in a string is of no use and we need to capture only the values.

import re

text='135.456'

pattern=r'(\d+)(?:.)(\d+)'

regobj=re.compile(pattern)

matobj=regobj.search(text)

matobj.groups()

('135', '456')

1. While counting the number of multiple lines or mutiple sentence in a string the positive look ahead makes a difference, without which we wont get the correct count of lines or sentences in a string.
2. Positive Lookahead allows to add a condition for what follows. Negative Lookahead is similar, but it looks behind. That is, it allows to match a pattern only if there’s something before it.
3. Referring to groups by name rather than by number in a standard expression helps to keep the code clear and easy to understand.
4. import re

text = "The cow jumped over the moon"

regobj=re.compile(r'(?P<w1>The)',re.I)

regobj.findall(text)

['The', 'the']

1. re.findall() module is used to search for all occurrences that match a given pattern. In contrast, re.search() will only return the first occurrence that matches the specified pattern. re.findall() will iterate over all the lines of the file and will return all non-overlapping matches of pattern in a single step.
2. Yes, It may have any name.