

please explain the below code

```
res = bool(re.fullmatch(r'^[-+]?[0-9]*[.][0-9]+', n))  
print(res)
```

Sure, I'll explain the code line by line.

- The code imports the re module, which provides regular expression operations in Python <https://docs.python.org/3/library/re.html>.
- The code defines a variable n, which is a string that represents a number.
- The code defines a regular expression pattern, which is a string that specifies a set of strings that matches it. The pattern is enclosed in r' ' to indicate that it is a raw string, meaning that backslashes are not treated as escape characters <https://docs.python.org/3/library/re.html>.
- The pattern is composed of the following elements:
  - ^: matches the start of the string <https://docs.python.org/3/library/re.html>.
  - [-+]?: matches an optional sign, either - or + <https://docs.python.org/3/library/re.html>.
  - [0-9]\*: matches zero or more digits from 0 to 9 <https://docs.python.org/3/library/re.html>.
  - [.]: matches a literal dot character <https://docs.python.org/3/library/re.html>.
  - [0-9]+: matches one or more digits from 0 to 9 <https://docs.python.org/3/library/re.html>.
  - \$: matches the end of the string <https://docs.python.org/3/library/re.html>.
- The code calls the re.fullmatch (pattern, string) function, which returns a match object if the whole string matches the pattern, or None otherwise <https://www.geeksforgeeks.org/re-fullmatch-function-in-python/>. The function takes two arguments: the pattern and the string to be matched.
- The code assigns the result of the function call to a variable res, which is either a match object or None.
- The code converts the res variable to a boolean value using the bool () function, which returns True if the argument is truthy (has some value) and False if it is falsy (has no value) <https://vimsky.com/examples/usage/re-fullmatch-function-in-python.html>. This means that res will be True if there is a match and False otherwise.
- The code prints the value of res to the standard output using the print () function <https://www.pythontutorial.net/python-regex/python-regex-fullmatch/>.

To summarize, the code checks if the string n is a valid decimal number with an optional sign

and prints True or False accordingly. For example, if n is '3.14', res will be True; if n is '3.14a', res will be False.

please explain

```
regex_pattern = r'(<)([a-zA-Z])([a-zA-Z0-9-~.\_]+)\@([a-zA-Z]+)\.([a-zA-Z]{1,3})(>)$'
```

Sure, I'll explain the regular expression pattern. The pattern is a raw string, meaning that backslashes are not treated as escape characters. The pattern is composed of the following elements:

- `r'`: indicates the start of the raw string.
- `(<)`: matches a literal left angle bracket character.
- `([a-zA-Z])`: matches one character that is either a lowercase or uppercase letter from a to z. This is a capturing group, meaning that the matched character can be accessed later.
- `([a-zA-Z0-9-~.\_]+)`: matches one or more characters that are either letters, digits, hyphens, dots, or underscores. This is another capturing group.
- `@`: matches a literal at sign character.
- `([a-zA-Z]+)`: matches one or more characters that are letters. This is another capturing group.
- `.`: matches a literal dot character.
- `([a-zA-Z]{1,3})`: matches one to three characters that are letters. This is another capturing group.
- `(>)`: matches a literal right angle bracket character.
- `$`: matches the end of the string.

The pattern is designed to match an email address enclosed in angle brackets, such as `<alice@example.com>`. The email address must have a letter as the first character, followed

by one or more alphanumeric or special characters, followed by an at sign, followed by one or more letters, followed by a dot, followed by one to three letters. The pattern has five capturing groups that can be used to extract the different parts of the email address.

For more information about regular expressions in Python, you can refer to Python RegEx - W3Schools [https://www.w3schools.com/python/python\\_regex.asp](https://www.w3schools.com/python/python_regex.asp) or Regular Expression HOWTO — Python 3.12.0 documentation <https://docs.python.org/3/howto/regex.html>.