Rounded Brackets

In regular expressions, rounded brackets, also known as parentheses or capturing groups, serve several important purposes:

- **Grouping**: Parentheses are primarily used to group parts of the regular expression together. This is useful for applying quantifiers (such as *, +, ?) to multiple characters or character classes, or for applying alternation (using the pipe symbol "|") to a group of characters.
  Example:
  - (abc)+ matches one or more occurrences of the sequence "abc".
  - (a|b)c matches either "ac" or "bc".
- **Capturing Groups**: Parentheses can also create capturing groups, which capture the substring matched by the enclosed part of the pattern. The substrings captured by these groups can be referenced later in the regex or in replacement patterns when performing search-and-replace operations.
  Example:
  regex

  Copy code
  ```
  (\d{3})-(\d{3})-(\d{4})
  ```

  This pattern captures a phone number with three groups: the area code, the prefix, and the line number.
- **Back References**: Capturing groups can be referenced later in the regular expression or in replacement patterns using backreferences. This allows you to match repeated occurrences of the same text or enforce repetition of a previous match.
  Example:
  regex

  Copy code
  ```
  (\w)\1
  ```

  This pattern matches a single word character followed by the same character (e.g., "aa", "bb", etc.).
- **Non-Capturing Groups**: Sometimes, you may want to group parts of a regular expression for applying quantifiers or alternations without capturing the matched substring. In such cases, you can use non-capturing groups, which are denoted by (?:...).
  Example:
  regex

```
Copy code
(?:https?|ftp)://\S+
```

> This pattern matches URLs starting with either "http://", "https://", or "ftp://", without capturing the protocol.

Overall, parentheses are a fundamental component of regular expressions, providing grouping, capturing, and referencing capabilities that enhance the expressiveness and functionality of the patterns.