

Regular Expressions

① `^[A-Z]{1}.*?(?:,)`

This regular expression asserts that the string is at the start of a line.

The following are two groups indicated by parentheses. For the first group, it must start with only one capital letter, followed any one or more characters.

The `?` at the end makes the `.*` lazy. This means the expression will accept the first match and not continue looking for more.

The second group represents matching a comma but is preceded by `?:`.

`?:` indicates to find the comma but not capture it. It is not in the returned string.

The first group stops matching once a comma is found.

② `(?:,|\s)([A-Z]{1}.*?)(?:\s|(?:,|:))`

The regular expression consists of three groups. The first is a non-capturing group that matches a comma or comma followed by a space. It does not return the,

The second group starts with exactly one uppercase letter followed by 0 or more of any character. With the `?` at the end of the group, it is lazy and will stop looking for matches once a satisfactory match is found. The last group is a non-capturing group that looks for either a space and open parenthesis `\s\(|` or a colon `:` or a comma `,`.

③ `\$.+`

This regular expression looks for strings that start with `$`, then is followed any one or more characters. Since the dollar sign is a special character in regular expression it needs to be escaped with the back slash to be treated as an actual dollar sign.

Summary:

The first expression gets all the text before the first comma, provided it starts with a capital letter.

The second expression gets the text after the first comma, and stops at either the first open parenthesis, colon or comma.

The third expression looks for a dollar sign, then gets everything after, provided there is text after.