

## Q1. Regex

Task: Write a regex to find

1. **U.S. ZIP codes** (disjunction + token boundaries)

Match 12345 or 12345-6789 or 12345 6789 (hyphen or space allowed for the +4 part).  
Make sure you only match whole tokens (not inside longer strings).

Sol: `\b\d{5}(:[- ]\d{4})?\b`

2. **Negation in disjunction** (word start rules)

Find all words that do not start with a capital letter. Words may include internal apostrophes/hyphens like don't, state-of-the-art.

Sol: `\b[a-z][a-zA-Z']*\b`

3. **Convenient aliases** (numbers, a bit richer)

Extract all numbers that may have:

- a. optional sign (+/-),
- b. optional thousands separators (commas),
- c. optional decimal part,
- d. optional scientific notation (e.g., 1.23e-4).

Sol: `[+-]?(?:\d{1,3}(?:,\d{3})*\d+)(?:\.\d*)(?:[eE][+-]?\d+)?`

4. **More disjunction** (spelling variants)

Match any spelling of "email": email, e-mail, or e mail. Accept either a space or a hyphen (including en-dash -) between e and mail, and be case-insensitive.

Sol: `e[ -]?mail`

5. **Wildcards, optionality, repetition** (with punctuation)

Match the interjection go, goo, gooo, ... (one or more o), as a word, and allow an optional trailing punctuation mark ! . , ? (e.g., gooo!).

Sol: `(?<!\w)go+[!.,?](?!\\w)`

6. **anchors** (line/sentence end with quotes)

Match lines that end with a question mark possibly followed only by closing quotes/brackets like ") "' ] and spaces.

Sol: `\\?[)'"\\s]*$`