

Assignment-7

1. What is the name of the feature responsible for generating Regex objects?

Ans. The feature responsible for generating Regex objects is the `re.compile()` function in Python's `re` module. It compiles a regular expression pattern into a regex object.

2. Why do raw strings often appear in Regex objects?

Ans. Raw strings (e.g., `r"pattern"`) are used to prevent Python from interpreting backslashes (`\`) as escape characters. In regular expressions, backslashes are frequently used, so raw strings make it easier to write and understand the regex pattern.

3. What is the return value of the `search()` method?

Ans. The `search()` method returns a Match object if a match is found. If no match is found, it returns `None`.

4. From a Match item, how do you get the actual strings that match the pattern?

Ans. You can get the actual matched strings using the `.group()` method of the Match object. For example, `match.group(0)` returns the entire match, and `match.group(1)` returns the match for the first capturing group.

5. In the regex created from `r'(\d\d\d)-(\d\d\d-\d\d\d\d\d)'`, what does group zero cover? Group 2? Group 1?

Ans.

- Group 0 covers the entire match: `(\d\d\d)-(\d\d\d-\d\d\d\d\d)` (e.g., `'123-456-7890'`).
- Group 1 covers the first part before the hyphen: `\d\d\d` (e.g., `'123'`).
- Group 2 covers the second part: `\d\d\d-\d\d\d\d\d` (e.g., `'456-7890'`).

6. In standard expression syntax, parentheses and intervals have distinct meanings. How can you tell a regex that you want it to fit real parentheses and periods?

Ans. To match literal parentheses and periods, you need to escape them with a backslash. For example, to match `(', ')`, and `.'`, use `\(, \)`, and `\.` in your regex.

7. The `findall()` method returns a string list or a list of string tuples. What causes it to return one of the two options?

Ans.

-If the regex pattern contains no capturing groups, `findall()` returns a list of strings.

- If the pattern contains capturing groups, `findall()` returns a list of tuples, with each tuple containing the matched groups.

8. In standard expressions, what does the `|` character mean?

Ans. The `|` character represents alternation (logical OR). It allows you to match one pattern or another. For example, `a|b` matches either "a" or "b".

9. In regular expressions, what does the `?` character stand for?

Ans. The `?` character can have two meanings:

1. It makes the preceding element optional, meaning it will match 0 or 1 occurrence.

2. In the context of non-greedy matching (e.g., `.*?)`, it tells the regex engine to match as few characters as possible.

10. In regular expressions, what is the difference between the `+` and `*` characters?

Ans.

`+` matches **one or more** occurrences of the preceding element.

`*` matches **zero or more** occurrences of the preceding element.

11. What is the difference between `{4}` and `{4,5}` in regular expressions?

Ans.

`{4}` matches **exactly 4** occurrences of the preceding element.

`{4,5}` matches **between 4 and 5** occurrences of the preceding element.

12. What do the `\d`, `\w`, and `\s` shorthand character classes signify in regular expressions?

Ans.

- `\d` matches any **digit** (equivalent to `[0-9]`).
- `\w` matches any **alphanumeric character** (equivalent to `[a-zA-Z0-9_]`).
- `\s` matches any **whitespace character** (spaces, tabs, and line breaks).

13. What do `\D`, `\W`, and `\S` shorthand character classes signify in regular expressions?

Ans.

- `\D` matches any **non-digit** character.
- `\W` matches any **non-word** character.
- `\S` matches any **non-whitespace** character.

14. What is the difference between `.*` and `.*?`?

Ans.

- `.*` is a greedy match, which means it matches as many characters as possible.
- `.*?` is a non-greedy or lazy match, meaning it matches as few characters as possible.

15. What is the syntax for matching both numbers and lowercase letters with a character class?

Ans. You can use a character class like `[a-z0-9]` to match both lowercase letters and numbers.

16. What is the procedure for making a normal expression in regex case-insensitive?

Ans. To make a regex case-insensitive, you can pass the `re.IGNORECASE` or `re.I` flag to the `re.compile()` function.

17. What does the `.` character normally match? What does it match if `re.DOTALL` is passed as the second argument in `re.compile()`?

Ans.

- Normally, the `.` character matches any character except newline.
- When `re.DOTALL` is passed, `.` matches any character, including newlines.

18. If ``numRegex = re.compile(r'\d+')``, what will ``numRegex.sub('X', '11 drummers, 10 pipers, five rings, 4 hen')`` return?

Ans. It will return: 'X drummers, X pipers, five rings, X hen'. All sequences of digits are replaced with 'X'.

19. What does passing ``re.VERBOSE`` as the second argument to ``re.compile()`` allow you to do?

Ans. It allows you to write multiline regex patterns with comments, making the pattern easier to read and understand.

20. How would you write a regex that matches a number with commas for every three digits?

Ans.

- The regex would be: ``^\d{1,3}(\,\d{3})*$``

- This matches numbers like ``'42'``, ``'1,234'``, and ``'6,368,745'``, but not ``'12,34,567'`` or ``'1234'``.

21. How would you write a regex that matches the full name of someone whose last name is Watanabe?

Ans.

- The regex would be: ``^[A-Z][a-z]*\sWatanabe$``

- This matches names like ``'Haruto Watanabe'``, ``'Alice Watanabe'``, but not ``'haruto Watanabe'``, ``'Mr. Watanabe'``, or ``'Watanabe'``.

22. How would you write a regex that matches a sentence where the first word is either Alice, Bob, or Carol; the second word is either eats, pets, or throws; and the third word is apples, cats, or baseballs; and the sentence ends with a period?

Ans.

- The regex would be: ``^(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.$``

- Adding ``re.IGNORECASE`` would allow it to match case-insensitive sentences like ``'Alice throws Apples.'``, ``'BOB EATS CATS.'``, etc.