

1. Regular Expressions (Regex)

1. Regular expressions are used to find patterns in text. (True) ✓
 2. The re module in Python is used for regex operations. (True) ✓
 3. \d matches only letters. (False) ✗ (It matches digits 0-9)
 4. \w matches only lowercase letters. (False) ✗ (It matches alphanumeric characters)
 5. \s matches any whitespace character, including tabs and newlines. (True) ✓
-

2. Space (\s)

6. \s+ matches one or more spaces. (True) ✓
 7. \s only matches single spaces but not tabs. (False) ✗ (It matches all whitespace characters)
 8. \S matches only digits. (False) ✗ (It matches any non-whitespace character)
 9. \s{2,} matches two or more spaces. (True) ✓
 10. The regex ^\s+|\s+\$ is used to remove leading and trailing spaces. (True) ✓
-

3. Caret (^)

11. ^ is used to match the start of a string. (True) ✓
 12. ^hello will match "hello world" but not "world hello". (True) ✓
 13. [^abc] matches only "a", "b", or "c". (False) ✗ (It matches everything except "a", "b", or "c")
 14. ^[0-9] matches a string starting with a digit. (True) ✓
 15. ^\$ matches an empty string. (True) ✓
-

4. Question Mark (?)

16. ? means zero or more occurrences of a character. (False) ✗ (It means zero or one occurrence)
 17. colour?r matches both "color" and "colour". (True) ✓
 18. a?b can match both "ab" and "b". (True) ✓
 19. ab?c will match "abc", "ac", and "abbc". (False) ✗ (It only matches "abc" and "ac")
 20. gr(e)?y matches both "gray" and "grey". (True) ✓
-

5. Kleene Star (*)

- 21. * means zero or more occurrences of a character. (True) ✓
 - 22. a*b matches "b", "ab", "aab", but not "bb". (False) ✗ (It also matches "b")
 - 23. .* can match an empty string. (True) ✓
 - 24. [A-Za-z]* matches any sequence of letters, including an empty string. (True) ✓
 - 25. .+ matches at least one character. (True) ✓
-

6. Wildcard (.)

- 26. . matches any single character except newlines. (True) ✓
 - 27. c.t can match "cat", "cot", and "cut". (True) ✓
 - 28. a.*b matches any string starting with "a" and ending with "b". (True) ✓
 - 29. ... matches exactly three characters. (True) ✓
 - 30. c.*t can match "cat", "carrot", and "ct". (True) ✓
-

7. Anchors (^, \$)

- 31. \$ is used to match the start of a string. (False) ✗ (It matches the end of a string)
 - 32. ^start.*end\$ ensures a string starts with "start" and ends with "end". (True) ✓
 - 33. ^hello\$ only matches "hello" as the full string. (True) ✓
 - 34. \bword\b matches "word" as a separate word. (True) ✓
 - 35. ^\$ matches any non-empty string. (False) ✗ (It matches an empty string)
-

8. Dollar (\$)

- 36. \$ ensures a pattern appears at the beginning of a string. (False) ✗ (It ensures the pattern appears at the end)
 - 37. \d\$ matches any string ending with a digit. (True) ✓
 - 38. ^[AEIOUaeiou].*[AEIOUaeiou]\$ matches words that start and end with vowels. (True) ✓
 - 39. ^a.*b\$ matches a string that starts with "a" and ends with "b". (True) ✓
 - 40. .+ does not match an empty string. (True) ✓
-

9. Disjunction (|)

- 41. | is used to match either one pattern or another. (True) ✓
 - 42. cat|dog matches "cat" or "dog". (True) ✓
 - 43. apple|banana matches "applebanana". (False) ✗ (It matches either "apple" or "banana", but not both together)
 - 44. (yes|no)? matches "yes", "no", or an empty string. (True) ✓
 - 45. (red|blue|green) matches any one of "red", "blue", or "green". (True) ✓
-

10. Grouping & Precedence

- 46. Parentheses () are used in regex for grouping expressions. (True) ✓
- 47. (ab)+ matches "ab", "abab", "ababab", etc. (True) ✓
- 48. (cat|dog)? matches "cat", "dog", or nothing. (True) ✓
- 49. (\b\w+\b)\s+\1 is used to detect repeated words. (True) ✓
- 50. Parentheses () have the lowest precedence in regex. (False) ✗ (They have the highest precedence)