**Regular Expressions (Regex) in Python**

Regular Expressions (regex) in Python are patterns used to search, match, and manipulate text. They are particularly useful for text validation, extraction, and replacement tasks.

Python provides regex functionality through the re module.

**1. Importing the re Module**

To use regex, first import the module:

import re

**2. Basic Regex Functions**

**a) re.match() – Matches only at the beginning of a string**

Checks if the pattern matches from the start of the string.

import re

pattern = r"hello"

text = "hello world"

match = re.match(pattern, text)

if match:

print("Match found!")

else:

print("No match.")

✅ **Output:** Match found!

**b) re.search() – Finds a match anywhere in the string**

text = "This is a hello world example."

search = re.search(r"hello", text)

if search:

print("Pattern found!")

✅ **Output:** Pattern found!

**c) re.findall() – Finds all occurrences in a string**

Returns a list of all matches.

text = "apple, banana, apple, orange"

matches = re.findall(r"apple", text)

print(matches)

✅ **Output:** ['apple', 'apple']

**d) re.sub() – Replaces occurrences in a string**

Replaces all matches with another string.

text = "I love apples and apples are great."

new\_text = re.sub(r"apples", "bananas", text)

print(new\_text)

✅ **Output:** "I love bananas and bananas are great."

**e) re.split() – Splits a string by the matched pattern**

Splits a string into a list based on a regex pattern.

text = "apple,banana,orange"

result = re.split(r",", text)

print(result)

✅ **Output:** ['apple', 'banana', 'orange']

**3. Special Characters in Regex**

| **Character** | **Meaning** | **Example** |
| --- | --- | --- |
| . | Any character except newline | a.b matches acb, a9b, etc. |
| ^ | Start of string | ^hello matches "hello world", but not "world hello" |
| $ | End of string | world$ matches "hello world" |
| \* | 0 or more occurrences | a\* matches "aaa", "", "a" |
| + | 1 or more occurrences | a+ matches "aaa", "a", but not "" |
| ? | 0 or 1 occurrence | colou?r matches "color" and "colour" |
| {n} | Exactly n occurrences | a{3} matches "aaa" |
| {n,} | At least n occurrences | a{2,} matches "aa", "aaa", etc. |
| {n,m} | Between n and m occurrences | a{2,4} matches "aa", "aaa", "aaaa" |
| [] | Character set | [aeiou] matches any vowel |
| \d | Any digit (0-9) | \d+ matches "123" in "abc123" |
| \D | Any non-digit | \D+ matches "abc" in "abc123" |
| \s | Any whitespace | "hello\s+world" matches "hello world" |
| \S | Any non-whitespace | "hello\S+world" matches "hello\_world" |
| \w | Any alphanumeric character (letters, digits, \_) | "hello\w+world" matches "hello123world" |
| \W | Any non-word character | hello\W+world matches "hello@world" |

**4. Using Groups in Regex**

You can use parentheses () to create groups.

text = "My phone number is 123-456-7890."

match = re.search(r"(\d{3})-(\d{3})-(\d{4})", text)

if match:

print("Full match:", match.group(0)) # Full match

print("First group:", match.group(1)) # First part (123)

print("Second group:", match.group(2)) # Second part (456)

print("Third group:", match.group(3)) # Third part (7890)

✅ **Output:**

Full match: 123-456-7890

First group: 123

Second group: 456

Third group: 7890

**5. Compiling Regex Patterns**

For efficiency, you can compile a regex pattern and reuse it.

pattern = re.compile(r"\d{3}-\d{3}-\d{4}")

text = "Contact me at 987-654-3210."

match = pattern.search(text)

print(match.group())

✅ **Output:** 987-654-3210

**6. Example: Validate an Email Address**

import re

def validate\_email(email):

pattern = r"^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$"

if re.match(pattern, email):

return "Valid email"

return "Invalid email"

print(validate\_email("test@example.com")) # ✅ Valid email

print(validate\_email("test@.com")) # ❌ Invalid email

**7. Example: Extracting Dates**

text = "Today's date is 2024-02-11 and yesterday was 2024-02-10."

dates = re.findall(r"\d{4}-\d{2}-\d{2}", text)

print(dates)

✅ **Output:** ['2024-02-11', '2024-02-10']

**Conclusion**

* re.match() → Match from the start
* re.search() → Match anywhere
* re.findall() → Get all matches
* re.sub() → Replace text
* re.split() → Split text