Python Project Day 3 – Email Extractor📄

By LearnWithBhawana

Problem Statement💭

You are given a block of messy text that may contain valid and invalid email addresses. Your task is to write a Python script that extracts only valid emails using regular expressions (regex).

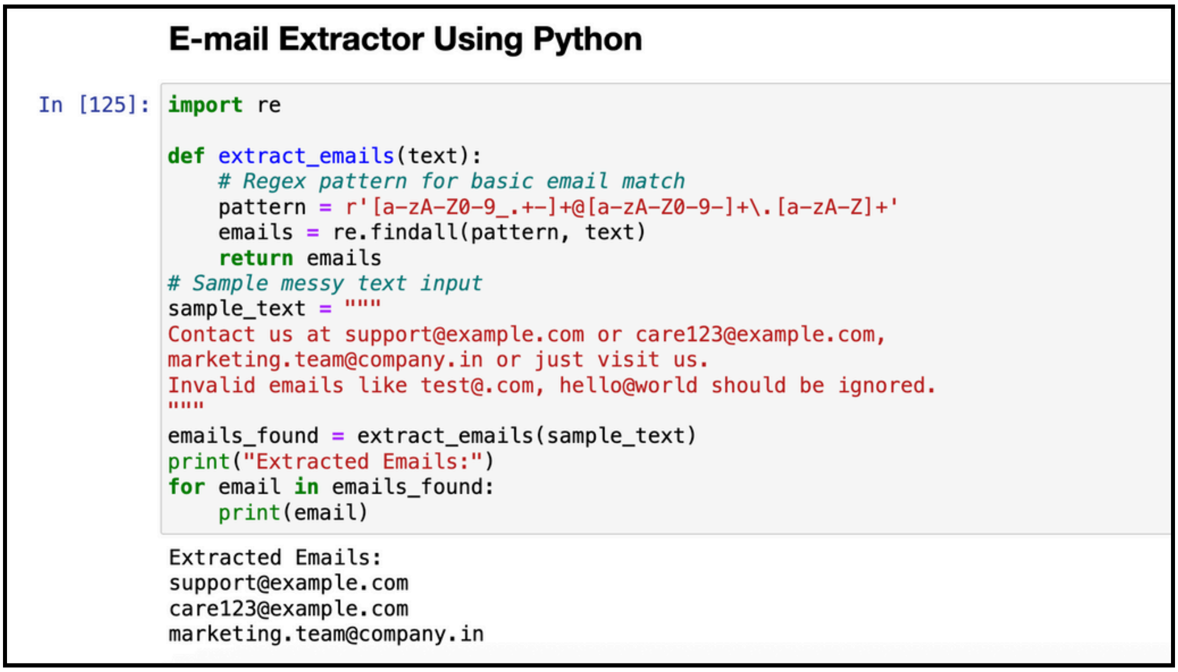
Python Code 💻

```
import re
def extract_emails(text):
   # Regex pattern for basic email match
   pattern = r'[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z]+'
   emails = re.findall(pattern, text)
   return emails
# Sample messy text input
sample_text = """
Contact us at support@example.com or care123@example.com,
marketing.team@company.in or just visit us.
Invalid emails like test@.com, hello@world should be ignored.
```

```
"""
emails_found = extract_emails(sample_text)
print("Extracted Emails:")
for email in emails_found:
    print(email)
```

✅ Sample Output:

```
Extracted Emails:
support@example.com
care123@example.com
marketing.team@company.in
```



E-mail Extractor Using Python

```
import re

def extract_emails(text):
    # Regex pattern for basic email match
    pattern = r'[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z]+'
    emails = re.findall(pattern, text)
    return emails
# Sample messy text input
sample_text = """
Contact us at support@example.com or care123@example.com,
marketing.team@company.in or just visit us.
Invalid emails like test@.com, hello@world should be ignored.
"""
emails_found = extract_emails(sample_text)
print("Extracted Emails:")
for email in emails_found:
    print(email)
```

```
Extracted Emails:
support@example.com
care123@example.com
marketing.team@company.in
```

Step-by-Step Code Explanation 🪜

1. **import re – Imports Python's built-in regex module.**
2. **def extract_emails(text) – Creates a function to extract emails from any text.**
3. **pattern = r'...' – This is the regex pattern to match email formats.**
4. **re.findall(pattern, text) – Finds and returns all matching email substrings.**
5. **The script uses a sample messy paragraph and prints only valid emails.**

Regex Pattern Breakdown :

pattern = r'[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z]+'

- **[a-zA-Z0-9_.+-]+**
  Matches the username part (letters, digits, _, ., +, -)

- @
  Matches the literal at symbol
- [a-zA-Z0-9-]+
  Matches the domain name like gmail, company-site, etc.
- \.
  Escaped dot (.) between domain and extension
- [a-zA-Z]+
  Matches the domain extension like com, in, org, etc.