In [1]:
```python
import re

def extract_emails(input_string):
    """
    Extract all email addresses from the given input string using regular expre

    Args:
    input_string (str): The string containing email addresses.

    Returns:
    list: A list of extracted email addresses.
    """
    # Define the regular expression pattern for email addresses
    email_pattern = r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'

    # Use re.findall to extract all email addresses
    emails = re.findall(email_pattern, input_string)

    return emails

# Test the function
test_input = 'Contact us at support@example.com and sales@example.org.'
result = extract_emails(test_input)
print("Extracted Emails:", result)
```

```
Extracted Emails: ['support@example.com', 'sales@example.org']
```

In [2]:
```
import re

def extract_emails(input_string):
    """
    Extract all valid email addresses from the given input string using advance

    Args:
    input_string (str): The string containing email addresses.

    Returns:
    list: A list of extracted valid email addresses.
    """
    # Advanced email regex to handle various cases
    email_pattern = r'''
        (?<!\S)             # Ensure no non-space character before the email (s
        [a-zA-Z0-9._%+-]+   # Local part: Alphanumeric and valid special charac
        @                   # At symbol
        (?:[a-zA-Z0-9-]+\.)+# Domain part: One or more subdomains ending with
        [a-zA-Z]{2,}        # Top-level domain: At least two characters
        (?!\S)              # Ensure no non-space character after the email (en
    '''

    # Compile the pattern with re.VERBOSE for readability and re.IGNORECASE for
    regex = re.compile(email_pattern, re.VERBOSE | re.IGNORECASE)

    # Find all matches
    emails = regex.findall(input_string)

    # Filter duplicates (if needed) and return
    return list(set(emails))

# Test the function
test_input = """Contact us at support@example.com and sales@example.org.
For inquiries, email <help@subdomain.example.co.uk>.
Invalid ones like @example should not be captured."""
result = extract_emails(test_input)

print("Extracted Emails:")
for email in result:
    print(email)
```

```
Extracted Emails:
support@example.com
```

In [ ]:
```
```