**Solutions to Worksheet 3: String Pattern Matching and Web Crawling**

**Activity 1: Understanding String Pattern Matching**

**Sample Code Questions:**

1. **What does the pattern \d{3}-\d{3}-\d{4} represent?**
   * It matches a phone number format where:
     + \d{3} matches three digits.
     + - matches a hyphen.
     + \d{3} matches another three digits.
     + - matches another hyphen.
     + \d{4} matches four digits.
2. **How many phone numbers are found in the sample text?**
   * Two phone numbers are found: 123-456-7890 and 987-654-3210.
3. **Modify the code to extract only phone numbers starting with 987:**

**Modified Code:**

import re

text = "John's phone number is 123-456-7890. Call him at 987-654-3210."

pattern = r"987-\d{3}-\d{4}" # Match phone numbers starting with 987

matches = re.findall(pattern, text)

print("Phone numbers starting with 987:", matches)

**Problem Solution:**  
Write a script to extract all email addresses:

import re

text = "Contact us at support@example.com, sales@company.org, or info@domain.net."

pattern = r"[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}"

emails = re.findall(pattern, text)

print("Email addresses found:", emails)

**Output:**

Email addresses found: ['support@example.com', 'sales@company.org', 'info@domain.net']

**Activity 2: Extracting Data with Web Crawling**

**Sample Code Questions:**

1. **What is the purpose of the soup.find\_all('a') method?**
   * It extracts all <a> (anchor) tags from the HTML, which usually contain hyperlinks.
2. **How would you modify the code to extract all paragraphs (<p> tags) instead of links?**
   * Replace soup.find\_all('a') with soup.find\_all('p').

**Problem Solution:**  
Write a Python script to fetch the title (<title>) of a webpage:

import requests

from bs4 import BeautifulSoup

url = "https://example.com" # Replace with an actual URL

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

title = soup.title.text

print("Webpage Title:", title)

**Activity 3: Building a Simple Web Crawler**

**Sample Code Questions:**

1. **How does the for loop help in crawling multiple pages?**
   * The loop constructs URLs for multiple pages (e.g., /page/1, /page/2, etc.) and processes each page sequentially.
2. **What is the role of the class\_='article-title' argument in soup.find\_all()?**
   * It specifies the HTML class attribute to narrow the search to elements with the article-title class.
3. **Modified Code to Save Titles to a File:**

import requests

from bs4 import BeautifulSoup

base\_url = "https://example-blog.com"

titles = []

for page in range(1, 4):

url = f"{base\_url}/page/{page}"

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

articles = soup.find\_all('h2', class\_='article-title')

for article in articles:

titles.append(article.text)

# Save titles to a file

with open("titles.txt", "w") as file:

for title in titles:

file.write(title + "\n")

print("Article titles saved to 'titles.txt'")