Assignments

Assignment 1: Matching at the Beginning

Problem: Write a regular expression to match strings that start with "Python."

Sample Input:

```
1. Python is a programming language.
2. Java is not Python.
3. Pythonic code is elegant.
```

Sample Output:

```
1. Match found: Python
3. Match found: Pythonic
```

Explanation:

The regular expression `^Python` is used to match strings that start with "Python."

Assignment 2: Matching Email Addresses

Problem: Write a regular expression to match valid email addresses.

Sample Input:

```
1. john.doe@email.com
2. invalid.email@.com
```

3. alice@domain

4. support@company.org

```

**Sample Output:**

```

1. Match found: john.doe@email.com

4. Match found: support@company.org

```

**Explanation:**

The regular expression `\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b` can be used to match valid email addresses.

---

### Assignment 3: Extracting Dates

**Problem:** Write a regular expression to extract dates in the format YYYY-MM-DD from a text.

**Sample Input:**

```

1. Date: 2022-01-15

2. Event on 2023-12-31

3. No date in this text.

```

**Sample Output:**

```

1. Matched text: 2022-01-15

2. Matched text: 2023-12-31

```

**Explanation:**

The regular expression `(\d{4}-\d{2}-\d{2})` captures dates in the specified format.

---

### Assignment 4: Matching Phone Numbers

**Problem:** Write a regular expression to match valid phone numbers in the format (XXX) XXX-XXXX.

**Sample Input:**

```
1. (123) 456-7890
2. Invalid number: 987-654-3210
3. (555) 123-4567
4. No phone number here.
```

**Sample Output:**

```
1. Match found: (123) 456-7890
3. Match found: (555) 123-4567
```

Explanation:

The regular expression `\(\d{3}\) \d{3}-\d{4}` can be used to match phone numbers in the specified format.

```

Assignment 5: Non-Greedy Matching

Problem: Write a regular expression to match HTML tags in a non-greedy manner.

Sample Input:

```
1. <div>Content</div> <p>Paragraph</p>
2. <span>Text</span> <a href="#">Link</a>
```

Sample Output:

```
1. Match found: <div>
1. Match found: </div>
2. Match found: <span>
2. Match found: </span>
```

Explanation:

The regular expression `<.*?>` is used for non-greedy matching of HTML tags.

Assignment 6: Extracting Domain Names

Problem: Write a regular expression to extract domain names from URLs.

Sample Input:

```
1. https://www.example.com/page
2. Invalid URL: ftp://notadomain.org/file
```

3. Visit our site at http://www.company.net

```

**Sample Output:**

```

1. Matched text: www.example.com

3. Matched text: www.company.net

```

**Explanation:**

The regular expression `://(www\.[A-Za-z0-9.-]+)` captures domain names from URLs.

---

### Assignment 7: Matching Sentences with Specific Words

**Problem:** Write a regular expression to match sentences containing the words "Python" or "programming."

**Sample Input:**

```

1. Python is a powerful programming language.

2. Java and C++ are also programming languages.

3. The snake python is not a programming language.

```

**Sample Output:**

```

1. Match found: Python is a powerful programming language.

2. Match found: Java and C++ are also programming languages.

```
```

Explanation:

The regular expression `\b(?:Python|programming)\b` matches sentences containing the specified words.

Assignment 8: Extracting Currency Values

Problem: Write a regular expression to extract currency values (e.g., $20.50) from a text.

Sample Input:

```
1. Total cost: $50.75
2. Invalid: €30.00
3. Payment received: $100.25 USD
```

Sample Output:

```
1. Matched text: $50.75
3. Matched text: $100.25
```

Explanation:

The regular expression `(\$[\d.]+)` captures currency values in the specified format.

Assignment 9: Matching Lines Starting with a Number

Problem: Write a regular expression to match lines that start with a number.

Sample Input:

```
1. First line of text.
2. 42 is the answer.
3. Not a number: ABC
4. 12345 - another number.
```

Sample Output:

```
2. Match found: 42 is the answer.
4. Match found: 12345 - another number.
```

Explanation:
The regular expression `^\d` matches lines that start with a digit.

Assignment 10: Extracting Hashtags

Problem: Write a regular expression to extract hashtags from a text.

Sample Input:

```
1. #Python is awesome!
2. No hashtag here.
3. #coding is fun! #programming
```

Sample Output:

```
1. Matched text: #Python
3. Matched text: #coding
3. Matched text: #programming
```

Explanation:

The regular expression `#(\w+)` captures words starting with '#' as hashtags.

Assignment 1: Matching at the Beginning
Solution:
```python
import re

pattern = re.compile(r'^Python')

strings = ["Python is a programming language.", "Java is not Python.", "Pythonic code is elegant."]

for s in strings:
    match = pattern.search(s)
    if match:
        print(f"Match found: {match.group()}")
    else:
        print("No match")
```

Assignment 2: Matching Email Addresses
Solution:
```python
import re

pattern = re.compile(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b')

emails = ["john.doe@email.com", "invalid.email@.com", "alice@domain", "support@company.org"]

for email in emails:
    match = pattern.search(email)
    if match:
        print(f"Match found: {match.group()}")
    else:
```

```
        print("No match")
```

Assignment 3: Extracting Dates
Solution:
```python
import re

pattern = re.compile(r'(\d{4}-\d{2}-\d{2})')

texts = ["Date: 2022-01-15", "Event on 2023-12-31", "No date in this text."]

for text in texts:
    match = pattern.search(text)
    if match:
        print(f"Matched text: {match.group(1)}")
    else:
        print("No match")
```

Assignment 4: Matching Phone Numbers
Solution:
```python
import re

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

numbers = ["(123) 456-7890", "Invalid number: 987-654-3210", "(555) 123-4567", "No phone number here."]

for number in numbers:
    match = pattern.search(number)
    if match:
        print(f"Match found: {match.group()}")
    else:
        print("No match")
```

Assignment 5: Non-Greedy Matching
Solution:
```python
import re
```

```
pattern = re.compile(r'<.*?>')

html_text = "<div>Content</div> <p>Paragraph</p> <span>Text</span> <a href='#'>Link</a>"

matches = pattern.findall(html_text)

for match in matches:
    print(f"Match found: {match}")
```

Assignment 6: Extracting Domain Names
Solution:
```python
import re

pattern = re.compile(r'://(www\.[A-Za-z0-9.-]+)')

urls = ["https://www.example.com/page", "Invalid URL: ftp://notadomain.org/file", "Visit our site at http://www.company.net"]

for url in urls:
    match = pattern.search(url)
    if match:
        print(f"Matched text: {match.group(1)}")
    else:
        print("No match")
```

Assignment 7: Matching Sentences with Specific Words
Solution:
```python
import re

pattern = re.compile(r'\b(?:Python|programming)\b')

sentences = ["Python is a powerful programming language.", "Java and C++ are also programming languages.", "The snake python is not a programming language."]

for sentence in sentences:
    match = pattern.search(sentence)
    if match:
        print(f"Match found: {match.group()}")
    else:
        print("No match")
```

Assignment 8: Extracting Currency Values
Solution:
```python
import re

pattern = re.compile(r'(\$[\d.]+)')

texts = ["Total cost: $50.75", "Invalid: €30.00", "Payment received: $100.25 USD"]

for text in texts:
    match = pattern.search(text)
    if match:
        print(f"Matched text: {match.group(1)}")
    else:
        print("No match")
```

Assignment 9: Matching Lines Starting with a Number
Solution:
```python
import re

pattern = re.compile(r'^\d')

lines = ["First line of text.", "42 is the answer.", "Not a number: ABC", "12345 - another number."]

for line in lines:
    match = pattern.search(line)
    if match:
        print(f"Match found: {line}")
    else:
        print("No match")
```

Assignment 10: Extracting Hashtags
Solution:
```python
import re

pattern = re.compile(r'#(\w+)')

texts = ["#Python is awesome!", "No hashtag here.", "#coding is fun! #programming"]
```

```
for text in texts:
    matches = pattern.findall(text)
    for match in matches:
        print(f"Matched text: #{match}")
```