Chapter 4: Regular Expressions (Page 64)

:Multiple Choice (MCQs)

1. **(a** re.match() (Section 4.6).
2. **(b** One or more digits (Section 4.4).
3. **(b** ing$ (Section 4.3).
4. **(b** ['o', 'o', 'a', 'i'] (Section 4.3).
5. **(b** ^[A-Za-z_]\w*$ (Standard variable naming rules).
6. **(c** Negation (Section 4.3).
7. **(b** ['Python', 'is', 'easy'] (Splitting by whitespace).
8. **(b** re.sub() (Section 4.6).

:True/False

1. **False** (match checks only the start; search checks everywhere).
2. **True** (The dot . matches anything except newline).
3. **False** (\w matches alphanumeric + underscore, not just uppercase).
4. **False** (\d{3} matches exactly 3 digits).
5. **False** (sub is specifically for substitution/replacing).
6. **True** (Case-sensitive by default unless re.IGNORECASE is used).
7. **False** (findall returns all non-overlapping matches).

True ($ anchors to the end).

```
# ========================================
# Chapter 4: Regular Expressions
```

```python
# ========================================
print("\n--- Chapter 4 Solutions ---")

# 1. Validate Email
email_pattern = r"^[A-Za-z0-9_.]+@[A-Za-z0-9.-]+\.(com|org|edu)$"
emails = ["user@example.com", "bad-email"]
for e in emails:
    print(f"Email '{e}' valid: {bool(re.match(email_pattern, e))}")

# 2. Extract Hashtags
text = "I love #Python and #AI"
print(f"Hashtags: {re.findall(r'#\w+', text)}")

# 3. Validate Phone
phone_pattern = r"^\+?\d{1,3}-?\d{3}-?\d{4}$"
phones = ["+1-555-1234", "5551234"]
for p in phones:
    print(f"Phone '{p}' valid: {bool(re.match(phone_pattern, p))}")

# 4. Word Frequency
text = "Python, Python! AI is great; Python AI."
words = re.findall(r"\b\w+\b", text)
print(f"Word Freq: {{w: words.count(w) for w in set(words)}}")

# 5. Duplicate Words
text = "This is is a test test"
print(f"Duplicates: {re.findall(r'\\b(\\w+)\\s+\\1\\b', text)}")

# 6. Extract Dates
text = "The events are on 2023-05-12 and 2024-01-01."
print(f"Dates: {re.findall(r'\d{4}-\d{2}-\d{2}', text)}")

# 7. Mask Sensitive Data
card = "Card: 1234-5678-9012-3456"
print(f"Masked: {re.sub(r'\d(?=.*\d{4})', '*', card)}")

# 8. Extract Languages
text = "I know Python, Java, and C++ but not Ruby."
print(f"Languages: {re.findall(r'\\b(Python|Java|C\\+\\+|Ruby)\\b', text)}")
```