الاسم : احمد السيد عبده على فرج | سكشن 1

Chapter 4

Multiple Choice Questions

1. Which function in Python checks only at the beginning of a string?
a) re.match()
b) re.search()
c) re.findall()
d) re.sub()

2. What does the regex pattern \d+ match?
a) One or more letters
b) One or more digits
c) Exactly one digit
d) Zero or more digits

3. Which regex will match any string ending with "ing"?
a) ^ing
b) ing$
*c) .*ing*
d) (ing)?

4. The output of re.findall(r"[aeiou]", "Python Programming") is:
a) ['a', 'o', 'o', 'a']
b) ['o', 'o', 'a', 'i']
c) ['y', 'a', 'i']
d) []

5. Which regex matches a valid variable name in Python (letters, numbers, underscores, not starting with digit) ?

a) ^\d\w$*
b) ^[A-Za-z_]\w$*
c) ^\w+$
d) ^[A-Z]\d$*

6. The metacharacter ^ inside brackets [^...] means:
a) Start of string
b) End of string
c) Negation (not these characters)
d) Match newline

7. What will be the result of: re.split(r"\s+", "Python is easy")
a) ['Python is easy']
b) ['Python', 'is', 'easy']
c) ['Python', ' is easy']
d) ['', 'Python', 'is', 'easy']

8. Which function is best for replacing substrings using regex?
a) re.match()
b) re.sub()ROGRAMMING WITH PYTHON
c) re.search()
d) re.findall()

True / False Questions

1. re.match() scans the entire string for a pattern.
Answer : F
2. The regex . matches any character except a newline.
Answer : T
3. Regex \w+ matches only uppercase letters.
Answer : F
4. The regex \d{3} matches exactly three digits.
Answer : T
5. re.sub() can be used for both searching and replacing.
Answer : T
6. Regex patterns in Python are case-sensitive unless re.IGNORECASE is used.
Answer : T
7. re.findall() returns only the first match found.
Answer : F
8. ^Python$ matches the string "I love Python".
Answer : F

```
# import re
# from collections import Counter

#----------------------
#----  problem 1  -----
#----------------------

# pattern = r"^[A-Za-z0-9._]+@[A-Za-z0-9.-
]+\.(com|org|edu)$"
# emails = ["user@example.com", "bad-email",
"test@domain.org"]
# valid = [email for email in emails if
re.match(pattern, email)]

# print(valid)

# ########################################################
#----------------------
#----  problem 2  -----
#----------------------

# Texts = "I love #Python and #AI"
# hashtags = re.findall(r"#\w+", Texts)
# print(hashtags)

# ########################################################
#----------------------
#----  problem 3  -----
#----------------------

# phones = ["+1-555-1234", "5551234", "123-456-7890"]
# pattern = r"^\+?\d{1,3}[-]?\d{3}[-]?\d{4}$"

# for p in phones:
#     print(f"{p}: {bool(re.match(pattern, p))}")

# ########################################################
#----------------------
#----  problem 4  -----
#----------------------

# Str = "Python, Python! AI is great; Python AI."
# words = re.findall(r"\w+", Str)
# freq = dict(Counter(words))

# print(freq)
```

```
# ######################################################
#----------------------
#----  problem 5  -----
#----------------------

# Str = "This is is a test test"
# duplicates = re.findall(r"\b(\w+)\s+\1\b", Str)

# print([f"{w} {w}" for w in duplicates])

# ######################################################
#----------------------
#----  problem 6  -----
#----------------------

# event = "The events are on 2023-05-12 and 2024-01-01."
# pattern = r"\d{4}-\d{2}-\d{2}"
# event_dates = re.findall(pattern, event)

# print(event_dates)

# ######################################################
#----------------------
#----  problem 7  -----
#----------------------

# credit_Card = "Card: 1234-5678-9012-3456"
# masked_Card = re.sub(r"\d(?=[\d-]*\d{4})",
"*",credit_Card)
# print(masked_Card)

# ######################################################
#----------------------
#----  problem 8  -----
#----------------------

# Str = "I know Python, Java, and C++ but not Ruby."
# pattern =
r"(?<!\w)(Python|Java|C\+\+|C#|Ruby|JavaScript|Go|Rust|S
wift|Kotlin)(?!\w)"
# langs = re.findall(pattern,Str,re.IGNORECASE)

# print(langs)
```