

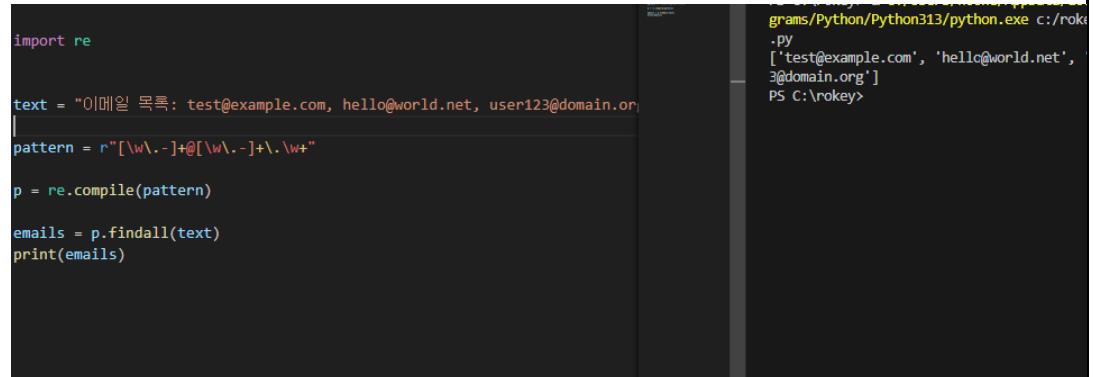
14차시	총 10 문제		연습: <input type="checkbox"/>	과제 : <input checked="" type="checkbox"/>	평가 : <input type="checkbox"/>
1. 정규표현식을 사용하려면 어떤 모듈을 임포트해야 하나요?					
a) regex					
b) re					
c) regular					
d) pattern					
2. re.match()와 re.search()의 주요 차이점은 무엇인가요?					
a) 둘 다 전체 문자열을 검색한다.					
b) re.match()는 문자열의 시작 부분부터 검사한다.					
c) re.search()는 문자열의 시작 부분부터 검사한다.					
d) 둘 다 다르지 않다.					
3. 다음 코드의 출력 결과는 무엇인가요?					
import re					
pattern = r'www+'					
text = "Hello, World!"					
print(re.findall(pattern, text))					
a) ['Hello', 'World']					
b) ['Hello', 'World!']					
c) ['Hello', ',', 'World', '!']					
d) ['Hello', ' ', 'World', '!']					
4. 정규식 ^abc는 무엇을 의미하나요?					
a) 문자열의 시작 부분에 abc가 있는지 확인					
b) 문자열의 끝 부분에 abc가 있는지 확인					
c) 문자열 전체가 abc인지 확인					
d) abc가 포함되어 있는지 확인					

5. 다음 코드의 출력 결과는 무엇인가요?

```
import re
pattern = r'(ab)+'
text = "ababab"
match = re.match(pattern, text)
print(match.group())
a) ab
b) ababab
c) aba
d) 에러 발생
```

6. 문자열에서 이메일 주소를 찾아 리스트로 반환하세요.

```
text = "이메일 목록: test@example.com, hello@world.net, user123@domain.org"
```



```
import re
text = "이메일 목록: test@example.com, hello@world.net, user123@domain.org"
pattern = r"[\w\.-]+@[\\w\.-]+\.\w+"
p = re.compile(pattern)
emails = p.findall(text)
print(emails)
```

PS C:\rokey> & C:/Users/hoone/AppData/Local/Programs/Python/Python313/python.exe c:/roke.py  
['test@example.com', 'hello@world.net', 'user123@domain.org']  
PS C:\rokey>

7. 주어진 문자열에서 한국식 전화번호(예: 010-1234-5678)를 찾아 리스트로 반환하세요.

```
text = "연락처: 010-1234-5678, 02-987-6543, 031-456-7890"
```



```
import re
text = "연락처: 010-1234-5678, 02-987-6543, 031-456-7890"
p = re.compile(r"\d{2,3}-\d{3,4}-\d{4}")
phones = p.findall(text)
print(phones)
```

PS C:\rokey> & C:/Users/hoone/AppData/Local/Programs/Python/Python313/python.exe c:/roke.py  
['010-1234-5678', '02-987-6543', '031-456-7890']  
PS C:\rokey>

8. 주어진 문자열에서 "Python"이 포함된 문장만 리스트로 반환하세요.

```
text = "I love Python. Java is also popular. Python is great for AI."
```

```
import re
text = "I love Python. Java is also popular. Python is great for AI."
pattern = r"[^.]*Python[^.]*\."
sentences = re.findall(pattern, text)
print(sentences)
```

```
PS C:\rokey> & C:/Users/hoone/AppData/Local/gram/Pthon/Python313/python.exe c:/roke.py
['I love Python.', ' Python is great for AI']
PS C:\rokey>
```

9. 문자열에서 숫자만 추출하여 리스트로 반환하세요.

```
text = "상품 코드: A123, B456, C789, 가격: 12000원"
```

```
import re
text = "상품 코드: A123, B456, C789, 가격: 12000원"
pattern = r"\d+"
numbers = re.findall(pattern, text)
print(numbers)
```

```
PS C:\rokey> & C:/Users/hoone/AppData/Local/gram/Pthon/Python313/python.exe c:/roke.py
[123, 456, 789, 12000]
PS C:\rokey>
```

10. 주어진 문자열에서 연속된 대문자(예: NASA, AI)만 찾아 리스트로 반환하세요.

```
text = "NASA is working on AI projects with IBM and Google."
```

```
import re
text = "NASA is working on AI projects with IBM and Google."
pattern = r"[A-Z]+"
result = re.findall(pattern, text)
print(result)
```

```
PS C:\rokey> & C:/Users/hoone/AppData/Local/gram/Pthon/Python313/python.exe c:/roke.py
['NASA', 'AI', 'IBM', 'G']
PS C:\rokey>
```