정규표현식 2

정규식 객체

- findall(string, flags)
  - String에서 패턴을 만족하는 문자열을 리스트로 반환

```
str = '''Window
Unix
Linux
Solaris'''
p = re.compile('^.+', re.M)
print(p.findall(str))
```

```
str = '''Window
Unix
Linux
Solaris'''
p = re.compile('^.+', re.S)
result = p.search(str)
```

정규식 객체

- Group에 이름 지정하기

```
m = re.match(r"(?P<first_name>₩w+) (?P<last_name>₩w+)",
"Malcolm Reynolds")
print(m.group('first_name', 'last_name'))
print(m.groups())
```

- Group에 디폴트값 주기

```
m = re.match(r"(₩d+)₩.?(₩d+)?", "24")
print(m.groups())
print(m.groups(0))
```

- Group을 dictionary로 출력하기

```
m = re.match(r"(?P<first_name>₩w+) (?P<last_name>₩w+)",
"Malcolm Reynolds")
print(m.groupdict())
```

정규식 객체

- Lookahead assertion

```
p = re.compile(".+:")
m = p.search("http://google.com")
print(m.group())
```

```
p = re.compile(".+(?=:)")
m = p.search("http://google.com")
print(m.group())
```

  - 해당 정규식과 매치되어도 해당 문자열은 통과되지 않음

```
p = re.compile('.*[.](?!bat$|exe$).*$')
```

  - 해당 정규식과 매치시 해당 조건이 아닌 경우만 매치함

정규식 객체

- Lookbehind assertion

```
p = re.compile("(?<=abc)def")
m = p.search("abcdef")
print(m.group())
```

  - 해당 스트링 앞에 있다면 매치를 시킴. 즉 해당 스트링 뒤에 있는 값과 매칭

```
m = re.search('(?<=-)₩w+', 'spam-egg')
print(m.group())
```

    - 하이픈 뒤의 문자열 찾음

정규식 객체

- start() / end()
  - 매치되는 스트링의 시작과 끝 위치의 인덱스를 반환

```
email = "tony@tiremove_thisger.net"
m = re.search("remove_this", email)
result = email[:m.start()] + email[m.end():]
print(result)
```

정규식 표현 예제

- 출력 함수

```
def displaymatch(match):
    if match is None:
        return None
    return '<Match: %r, groups=%r>' % (match.group(), match.groups())
```

%r : repr()을 사용하여 변환
%s : str()을 사용하여 변환

- 포커 (a, 1,2~9, 10(t), j, q, k)

```
valid = re.compile(r"^[a2-9tjqk]{5}$")
displaymatch(valid.match("akt5q"))
displaymatch(valid.match("akt5e"))
displaymatch(valid.match("akt"))
displaymatch(valid.match("727ak"))
```

Making a Phonebook

```
text = """Ross: McFluff: 834.345.1254: 155 Elm Street
Ronald: Heathmore: 892.345.3428 436: Finley Avenue
Frank: Burger: 925.541.7625 662: South Dogwood Way
Heather: Albrecht: 548.326.4584 919: Park Place"""

entries = re.split("₩n", text)
result = [re.split(":?", entry, 4) for entry in entries]
print(result)
```

Urllib

- Urllib.request
  - Url을 오픈하기 위한 라이브러리
  - **urlopen(url)**

```
import urllib.request
with urllib.request.urlopen('http://www.python.org/') as f:
    print(f.read())
    print(f.read(300)) #300 byte
    print(f.read(300).decode("utf-8")) #encoding mode utf-8
```

Urllib

- Urllib.parse
  - Url의 컴포넌트를 파싱하기 위한 라이브러리
    - scheme://netloc/path;parameters?query#fragment
  - Urlparse(url)

```
from urllib.parse import urlparse
result = 
urlparse('http://search.naver.com/search.naver?where=nexearch&query=urllib.parse&sm=top_hty&fbm=1&ie=utf8')
print(result)
print(result.scheme)
print(result.geturl())
```