```
Lab. search() 다루기

1. 사용 tool
   -Jupyter Notebook
   -Microsoft Visual Studio Code

2. Code
   #re.search()
   #문자열의 일부분이 정규 표현식과 matching되는지 확인하는 method
   #첫번째로 pattern을 찾으면 match 객체 반환
   #못찾으면 None 반환
   #matching되는 문자열의 앞부분에 있지 않다면 match() 대신에
   search()를 사용하는 것이 좋다.

   import re

   result = re.search(r'abc', 'abcdef')
   print(type(result))   #<class 're.Match'>

   print(result.start())   #0
   print(result.end())     #3
   print(result.group())  #abc

   result = re.search(r'abc', '123abcdef')
   print(result.start())   #3
   print(result.end())     #6
   print(result.group())  #abc

   result = re.search(r'abc', '123abdef')
   print(result)   #None


```

```
32  result = re.search(r'\d\d', '123abcdef321')
33  print(result)   #<re.Match object; span=(0, 2),
    match='12'>
34
35  result = re.search(r'\d\d\d\d', '123abcdef321')
36  print(result)   #None
37
38  result = re.search(r'\d\d\d\w', '123abcdef321')
39  print(result)   #<re.Match object; span=(0, 4),
    match='123a'>
40
41  result = re.search(r'..\w\w', '@#$%ABCDabcd')
42  print(result)    #<re.Match object; span=(2, 6),
    match='$%AB'>
43
44
45  #Metacharacters [] 다루기
46  result = re.search(r'[cbm]at', 'cat')
47  print(result)    #<re.Match object; span=(0, 3),
    match='cat'>
48
49  result = re.search(r'[cbm]at', 'bat')
50  print(result)   #<re.Match object; span=(0, 3),
    match='bat'>
51
52  result = re.search(r'[0-9]hello', '4hello')
53  print(result)   #<re.Match object; span=(0, 6),
    match='4hello'>
54
55  result = re.search(r'[0-7]hello', '9hello')
56  print(result)    #None
57
```

```
58  result = re.search(r'[abc.^]amera', 'camera')
59  print(result)    #<re.Match object; span=(0, 6),
    match='camera'>
60
61  result = re.search(r'[abc.^]amera', '.amera')
62  print(result)    #<re.Match object; span=(0, 6),
    match='.amera'>
63
64  result = re.search(r'[abc.^]amera', 'damera')
65  print(result)    #None
66
67  result = re.search(r'[^abc]amera', 'camera')
68  print(result)    #None
69
70  result = re.search(r'[^abc]amera', 'damera')
71  print(result)    #<re.Match object; span=(0, 6),
    match='damera'>
72
73
74  #Special Character Classes \ 다루기
75  result = re.search(r'\sand ', 'Apple and Banana')
76  print(result)    #<re.Match object; span=(5, 10), match='
    and '>
77
78  result = re.search(r'\Sand ', 'Apple and Banana')
79  print(result)    #None
80
81  result = re.search(r'\Sand ', 'Apple sand Banana')
82  print(result)    #<re.Match object; span=(5, 11),
    match='sand '>
83
84
```

```
85  # .(모든문자) 다루기
86  result = re.search(r'.and', 'land')
87  print(result)    #<re.Match object; span=(0, 4),
    match='land'>
88
89  result = re.search(r'\.and', 'land')
90  print(result)    #None
91
92  result = re.search(r'd.g', 'dog')
93  print(result)    #<re.Match object; span=(0, 3),
    match='dog'>
94
95
96  #Repetition Cases(반복패턴) 다루기
97  result = re.search(r'a[bcd]*b', 'abcbdccb')
98  print(result)    #<re.Match object; span=(0, 8),
    match='abcbdccb'>
99
100 result = re.search(r'b\w+a', 'banana')
101 print(result)    #<re.Match object; span=(0, 6),
    match='banana'>
102
103 result = re.search(r'i+', 'piigiii')
104 print(result)    #<re.Match object; span=(1, 3),
    match='ii'>
105
106 result = re.search(r'pi+g', 'piig')
107 print(result)    #<re.Match object; span=(0, 4),
    match='piig'>
108
109 result = re.search(r'pi+g', 'pg')
110 print(result)    #None
```

```
111
112     result = re.search(r'pi*g', 'pg')
113     print(result)     #<re.Match object; span=(0, 2),
        match='pg'>
114
115     result = re.search(r'https?', 'https://www.google.com')
116     print(result)     #<re.Match object; span=(0, 5),
        match='https'>
117
118     result = re.search(r'https?', 'httpk://www.google.com')
119     print(result)     #<re.Match object; span=(0, 4),
        match='http'>
120
121     result = re.search(r'n\w+a', 'carnival')
122     print(result)     #<re.Match object; span=(3, 7),
        match='niva'>
123
124
125     #^, $ 다루기
126     result = re.search(r'^n\w+a', 'carnival')
127     print(result)     #None
128
129     result = re.search(r'^c\w+a', 'carnival')
130     print(result)     #<re.Match object; span=(0, 7),
        match='carniva'>
131
132     result = re.search(r'c\w+al$', 'carnival')
133     print(result)     #<re.Match object; span=(0, 8),
        match='carnival'>
134
135     result = re.search(r'c\w+a$', 'carnival')
136     print(result)     #None
```

```
137
138
139     #grouping () 다루기
140     result = re.search(r'\w+@.+', 'javaexpert@nate.com')
141     print(result)     #<re.Match object; span=(0, 19),
        match='javaexpert@nate.com'>
142     print(result.group())  #javaexpert@nate.com
143
144     result = re.search(r'(\w+)@(.+)', 'javaexpert@nate.com')
145     print(result.group(1))  #javaexpert
146     print(result.group(2))  #nate.com
147     print(result.group(0))  #javaexpert@nate.com
148
149
150     #{} 다루기
151     result = re.search(r'car*al', 'carrrrral')
152     print(result)     #<re.Match object; span=(0, 9),
        match='carrrrral'>
153
154     result = re.search(r'car{3}al', 'carrrrral')
155     print(result)     #None
156
157     result = re.search(r'car{3}al', 'carrral')
158     print(result)     #<re.Match object; span=(0, 7),
        match='carrral'>
159
160     result = re.search(r'car{3,5}al', 'carrrrral')
161     print(result)     #<re.Match object; span=(0, 9),
        match='carrrrral'>
162
163
164     #Minimum matching
```

```
165   result = re.search(r'<.+>', '<body>hello</body>')
166   print(result)     #<re.Match object; span=(0, 18),
      match='<body>hello</body>'>
167
168   result = re.search(r'<.+?>', '<body>hello</body>')
169   print(result)     #<re.Match object; span=(0, 6),
      match='<body>'>
170
171   result = re.search(r'a{3,5}', 'aaaaa')
172   print(result)     #<re.Match object; span=(0, 5),
      match='aaaaa'>
173
174   result = re.search(r'a{3,5}?', 'aaaaa')
175   print(result)     #<re.Match object; span=(0, 3),
      match='aaa'>
176
177
178   #flag 다루기
179   result = re.search(r'[a-z]+', '0010010 Has at least one 010
      letter 0010010', re.I)
180   print(result)  #<re.Match object; span=(8, 11),
      match='Has'>
181
182   result = re.search(r'[a-z]+', '0010010 Has at least one 010
      letter 0010010')
183   print(result)  #<re.Match object; span=(9, 11),
      match='as'>
184
185   line = "Cats are smarter than dogs";
186   searchObj = re.search( r'(.*) are (.*?) .*', line, re.M|re.I)
187
188   if searchObj:
```

```
189       print("searchObj.group() : ", searchObj.group())
190       print("searchObj.group(1) : ", searchObj.group(1))
191       print("searchObj.group(2) : ", searchObj.group(2))
192    else:
193       print("Nothing found!!")
```