

```

1 Lab. Using search()
2
3 1. 사용 tool
4 -Jupyter Notebook
5 -Microsoft Visual Studio Code
6
7
8 2. Code
9 #re.search()
10 #문자열의 일부분이 정규 표현식과 matching되는지 확인하는 method
11 #첫번째로 pattern을 찾으면 match 객체 반환
12 #못찾으면 None 반환
13 #matching되는 문자열의 앞부분에 있지 않다면 match() 대신에 search()를 사용하는 것이 좋다.
14
15 import re
16
17 result = re.search(r'abc', 'abcdef')
18 print(type(result)) #<class 're.Match'>
19
20 print(result.start()) #0
21 print(result.end()) #3
22 print(result.group()) #abc
23
24 result = re.search(r'abc', '123abcdef')
25 print(result.start()) #3
26 print(result.end()) #6
27 print(result.group()) #abc
28
29 result = re.search(r'abc', '123abdef')
30 print(result) #None
31
32
33 result = re.search(r'\d\d', '123abcdef321')
34 print(result) #<re.Match object; span=(0, 2), match='12'>
35
36 result = re.search(r'\d\d\d\d', '123abcdef321')
37 print(result) #None
38
39 result = re.search(r'\d\d\d\w', '123abcdef321')
40 print(result) #<re.Match object; span=(0, 4), match='123a'>
41
42 result = re.search(r'\. \w\w', '@#%$ABCDabcd')
43 print(result) #<re.Match object; span=(2, 6), match=' $%AB'>
44
45
46 #Metacharacters [] 다루기
47 result = re.search(r'[cbm]at', 'cat')
48 print(result) #<re.Match object; span=(0, 3), match='cat'>
49
50 result = re.search(r'[cbm]at', 'bat')
51 print(result) #<re.Match object; span=(0, 3), match='bat'>
52
53 result = re.search(r'[0-9]hello', '4hello')
54 print(result) #<re.Match object; span=(0, 6), match='4hello'>
55
56 result = re.search(r'[0-7]hello', '9hello')
57 print(result) #None
58
59 result = re.search(r'[abc.^]amera', 'camera')
60 print(result) #<re.Match object; span=(0, 6), match='camera'>
61
62 result = re.search(r'[abc.^]amera', '.amera')
63 print(result) #<re.Match object; span=(0, 6), match='.amera'>
64
65 result = re.search(r'[abc.^]amera', 'damera')
66 print(result) #None
67
68 result = re.search(r'^[abc]amera', 'camera')
69 print(result) #None
70
71 result = re.search(r'^[abc]amera', 'damera')
72 print(result) #<re.Match object; span=(0, 6), match='damera'>
73
74
75 #Special Character Classes \ 다루기
76 result = re.search(r'\sand ', 'Apple and Banana')
77 print(result) #<re.Match object; span=(5, 10), match=' and '>
78
79 result = re.search(r'\Sand ', 'Apple and Banana')
80 print(result) #None
81
82 result = re.search(r'\Sand ', 'Apple sand Banana')
83 print(result) #<re.Match object; span=(5, 11), match='sand '>
84

```

```

85
86 #.(모든문자) 다루기
87 result = re.search(r'.and', 'land')
88 print(result) #<re.Match object; span=(0, 4), match='land'>
89
90 result = re.search(r'\.and', 'land')
91 print(result) #None
92
93 result = re.search(r'd.g', 'dog')
94 print(result) #<re.Match object; span=(0, 3), match='dog'>
95
96
97 #Repetition Cases(반복패턴) 다루기
98 result = re.search(r'a[bcd]*b', 'abcdcccb')
99 print(result) #<re.Match object; span=(0, 8), match='abcdcccb'>
100
101 result = re.search(r'b\w+a', 'banana')
102 print(result) #<re.Match object; span=(0, 6), match='banana'>
103
104 result = re.search(r'i+', 'piigiii')
105 print(result) #<re.Match object; span=(1, 3), match='ii'>
106
107 result = re.search(r'pi+g', 'piig')
108 print(result) #<re.Match object; span=(0, 4), match='piig'>
109
110 result = re.search(r'pi+g', 'pg')
111 print(result) #None
112
113 result = re.search(r'pi*g', 'pg')
114 print(result) #<re.Match object; span=(0, 2), match='pg'>
115
116 result = re.search(r'https?', 'https://www.google.com')
117 print(result) #<re.Match object; span=(0, 5), match='https'>
118
119 result = re.search(r'https?', 'httpk://www.google.com')
120 print(result) #<re.Match object; span=(0, 4), match='http'>
121
122 result = re.search(r'n\w+a', 'carnival')
123 print(result) #<re.Match object; span=(3, 7), match='niva'>
124
125
126 #^, $ 다루기
127 result = re.search(r'^n\w+a', 'carnival')
128 print(result) #None
129
130 result = re.search(r'^c\w+a', 'carnival')
131 print(result) #<re.Match object; span=(0, 7), match='carniva'>
132
133 result = re.search(r'c\w+a$', 'carnival')
134 print(result) #<re.Match object; span=(0, 8), match='carnival'>
135
136 result = re.search(r'c\w+a$', 'carnival')
137 print(result) #None
138
139
140 #grouping () 다루기
141 result = re.search(r'\w+@.+', 'javaexpert@nate.com')
142 print(result) #<re.Match object; span=(0, 19), match='javaexpert@nate.com'>
143 print(result.group()) #javaexpert@nate.com
144
145 result = re.search(r'(\w+)@(.+)', 'javaexpert@nate.com')
146 print(result.group(1)) #javaexpert
147 print(result.group(2)) #nate.com
148 print(result.group(0)) #javaexpert@nate.com
149
150
151 #{ } 다루기
152 result = re.search(r'car*al', 'carrrrral')
153 print(result) #<re.Match object; span=(0, 9), match='carrrrral'>
154
155 result = re.search(r'car{3}al', 'carrrrral')
156 print(result) #None
157
158 result = re.search(r'car{3}al', 'carrral')
159 print(result) #<re.Match object; span=(0, 7), match='carrral'>
160
161 result = re.search(r'car{3,5}al', 'carrrrral')
162 print(result) #<re.Match object; span=(0, 9), match='carrrrral'>
163
164
165 #Minimum matching
166 result = re.search(r'<.+>', '<body>hello</body>')
167 print(result) #<re.Match object; span=(0, 18), match='<body>hello</body>'>
168

```

```

169 result = re.search(r'<.+?>', '<body>hello</body>')
170 print(result)    #<re.Match object; span=(0, 6), match='<body>'>
171
172 result = re.search(r'a{3,5}', 'aaaaa')
173 print(result)    #<re.Match object; span=(0, 5), match='aaaaa'>
174
175 result = re.search(r'a{3,5}?', 'aaaaa')
176 print(result)    #<re.Match object; span=(0, 3), match='aaa'>
177
178
179 #flag 다루기
180 result = re.search(r'[a-z]+', '0010010 Has at least one 010 letter 0010010', re.I)
181 print(result)    #<re.Match object; span=(8, 11), match='Has'>
182
183 result = re.search(r'[a-z]+', '0010010 Has at least one 010 letter 0010010')
184 print(result)    #<re.Match object; span=(9, 11), match='as'>
185
186 line = "Cats are smarter than dogs";
187 searchObj = re.search( r'(.*) are (.*?) .*', line, re.M|re.I)
188
189 if searchObj:
190     print("searchObj.group() : ", searchObj.group())
191     print("searchObj.group(1) : ", searchObj.group(1))
192     print("searchObj.group(2) : ", searchObj.group(2))
193 else:
194     print("Nothing found!!")

```