

 kim hanwoong 내용 추가함.

ae7c839 24 days ago

2 contributors  

101 lines (83 sloc) | 4.29 KB

레귤러 익스프레션을 사용하여 경로 추출하기

일반적인 경로 분리방법은 실제로 잘 사용하지 않습니다. 프로젝트 중간에 경로가 추가될 수도 있고 규칙이 바뀔 수 있기 때문입니다. 그래서 실제 경로에서 데이터를 불러올 때는 레귤러 익스프레션을 일반적으로 선호하기도 합니다. 다른 개발자들이 안심할 수 있도록 테스트 코드를 작성해두면 더 신뢰받는 함수가 될것입니다.

우리가 실습하는 문자열은 아래와 같습니다.

```
리눅스 : /project/circle/shot/F00/0010/comp/F00_0010_comp_v001.nk
윈도우 : \\10.20.30.40\project\circle\shot\F00\0010\F00_0010_comp_v001.nk
```

참고 : 일반적으로 기업중 컴퓨터가 많이 필요한 회사는 내부망 네트워크 설정시 사설IP A 클래스 대역을 많이 사용합니다. A 클래스의 특징은 네자리의 IP 주소 중에서 두번째, 세번째, 네번째 주소를 회사 마음대로 부여할 수 있다는 점입니다. 회사 내부에 컴퓨터가 몇대가 될지 모르지만 굉장히 많은 컴퓨터가 미래에 필요할 것 같다면 보통 최초 A 클래스로 셋팅합니다. A 클래스는 255 x 255 x 255 = 총 16,581,375개의 주소(10.0.0.0 ~ 10.255.255.255)를 가질 수 있습니다. 하지만 보통 제한된 네트워크로 구성하여 사용합니다.

예전과 다른 방식으로 프로젝트 문자열을 가지고 오는 함수를 작성하겠습니다. 이번엔 파이썬 레귤러 익스프레션을 사용하겠습니다.

실제 실무에서는 프로젝트를 가지고 오기 위해서 아래와 같은 코드를 작성합니다. 레귤러 익스프레션을 사용하면 프로그래머 입장에서 좀더 명확하게 경로 정보를 가지고 올 수 있습니다.

pathapi.py

```
#!/usr/bin/env python
#coding:utf8
import sys
import re

path = "/project/circle/shot/F00/0010/comp/F00_0010_comp_v001.nk"

def project(path):
    p = re.findall('/project/(\\w+)', path.replace("\\\\", "/"))
    if len(p) != 1:
        return "", "project 정보를 경로에서 가지고 올 수 없습니다."
    return p[0], None

if __name__ == "__main__":
    project, err = project(path)
    if err:
        sys.stderr.write(err+"\n")
        sys.exit(1)
    print project
```

- () 내부에 적혀있는 정규식과 일치하는 문자열 그룹을 반환할 때 사용합니다.
- \\S 를 사용한 이유는 공백이 없는 유니코드가 프로젝트이름 조건과 같기 때문입니다.
- [^/] 가 붙어있는 이유는 /project/circle/ 형태로 입력을 받을 때 맨 뒤 / 를 제거하기 위함 입니다. [] 문자 내부에 ^ 문자를 사용하면 예외시키라는 뜻이 있습니다.

pathapi_test.py

```
#!/usr/bin/env python
#coding:utf8
import unittest
from pathapi import *

class Test_path(unittest.TestCase):
    def test_project(self):
        self.assertEqual(project("/project/circle"), ("circle",None))
        self.assertEqual(project("/project/circle/"), ("circle",None))
        self.assertEqual(project("\\\\10.20.30.40\\project\\circle\\"), ("circle",None))
        self.assertEqual(project("/project/circle/shot/F00/0010/comp/F00_0010_comp_v001.nk"), ("circle",

if __name__ == "__main__":
    unittest.main()
```

아래처럼 python test시 OK가 출력되면 테스트가 무사히 통과된 함수가 됩니다.

```
$ python pathapi_test.py
.
-----
Ran 1 test in 0.000s

OK
```

실습

- pathapi.py 를 작성합니다.
- project 를 가지고 오는 레귤러 익스프레션 추가
- 테스트 코드 작성하기.
- seq 를 가지고 오는 레귤러 익스프레션 추가
- shot 을 가지고 오는 레귤러 익스프레션 추가
- task 를 가지고 오는 레귤러 익스프레션 추가
- 파일명 또는 경로에서 버전을 가지고 오는 함수 추가

```
import re
p = "F00_0010_comp_v001.nk"
re.findall("_v(\\d+)",p)
```

- 시퀀스 넘버를 가지고 오는 익스프레션 추가

```
import re
f = "F00_0010_comp_v001.1001.exr"
re.findall("\\.(\\d+)\\. ",f)
```

- 나머지 각 함수에 대해서 테스트코드 작성

Reference

- <https://docs.python.org/2/library/re.html>
- <https://www.xpressengine.com/tip/22635583>