

Learning Script - 2

Python 1

보안기술 본부 김윤경

2018.11.05

The subject comes first, the medium second.

주제가 제일 중요하고 도구는 그 다음이다.

- Richard Prince

스크립트는 단지 하나의 도구입니다.

가장 중요한 것은 원래 하려는 목표를 정확히 이해하고, 관련 지식을 갖춰 적합한 과정으로 수행하는 것입니다.

스크립트는, 할 일의 프로세스가 결정되고 검증된 후, 그 일에 도움을 주려고 사용하는 도구중 하나입니다.

Script는 로 실행한다.

스크립트는 일종의 대본입니다.

목표를 달성하기 위한 작업 프로세스를, 명령 해석기의 문법에 맞춰 정의한 것입니다. 이 해석기를 인터프리터(Interpreter)라고 합니다.
인터프리터는 스크립트에 정의된 일련의 과정을 하나씩 해석하여 수행합니다.

- (1) Bash, Windows 파워셸 같이 특정 OS의 명령어를 해석하여 수행하는 인터프리터,
 - (2) Python, Ruby, JavaScript, TCL와 같이 프로그래밍 문법을 갖춘 전용 인터프리터
- 크게 두 분류가 있습니다.

인터프리터의 속성, 문법 또는 명령어세트를 아는 것이 스크립트 활용에 중요합니다.

Python – Console

버전 확인 및 Python Console

```
$ python -V
Python 3.6.5 :: Anaconda, Inc.

$ python
Python 3.6.5 |Anaconda, Inc.
>>> a = 1
>>> b = 100
>>> a + b
101
>>> c = a + b
>>> print(c)
101
>>> quit()
```

참고: python.org에서 python을 설치한 경우 IDLE 사용



01-5 파이썬 둘러보기
01-6 파이썬과 에디터
02장 파이썬 프로그래밍의 기초, 자료형
02-1 숫자형
02-2 문자열 자료형
02-3 리스트 자료형
02-4 튜플 자료형
02-5 딕셔너리 자료형
02-6 집합 자료형
02-7 불 자료형
02-8 자료형의 값을 저장하는 공간, 변수
03장 프로그램의 구조를 쌓는다! 제어문
03-1 if문
03-2 while문
03-3 for문
04장 프로그램의 입력과 출력은 어떻게 해야 할까?
04-1 함수
04-2 사용자 입력과 출력
04-3 파일 읽고 쓰기
05장 파이썬 날개달기

스크립트 활용에 필요한 핵심 지식:

- 자료형, 제어구조, 입출력
- 점프 투 파이썬(<https://wikidocs.net/book/1>) 2, 3, 4장 추천

- 리스트 정의

```
members = ["Kim", "Park", "Jang", "Ryu"]  
scores = [ 92, 99, 75, 87, 82]  
empty_list = []
```

- 요소 추가

```
members.append("Lee")
```

- 리스트 요소 출력

```
for m in members:  
    print(m)
```

```
Kim  
Park  
Jang  
Ryu  
Lee
```

- (참고)1에서 100까지 더하기

```
sum1 = 0  
for i in range(1,101):  
    sum1 = sum1 + i
```

평균점수?

```
scores = [ 92, 99, 75, 87, 82]
```

- 방법1

```
sum1 = 0
for score in scores:
    sum1 += score # sum1 = sum1 + score

avg = sum1/len(scores)
print("average = " + str(avg))
```

- 방법2

```
avg = sum(scores) / len(scores)
print("average = {}".format(avg))
```

정렬

```
scores.sort()
print(socres)
```

```
[75, 82, 87, 92, 99]
```

리스트 요소 추가

```
members = ["Kim", "Park", "Jang", "Ryu"]  
members.append("Kim")  
members.append("Kim")  
members.append("Lee")  
print(members)
```

Output:
['Kim', 'Park', 'Jang', 'Ryu', 'Kim', 'Kim', 'Lee']

집합 요소 추가

```
members = {"Kim", "Park", "Jang", "Ryu"}  
members.add("Kim")  
members.add("Kim")  
members.add("Lee")  
print(members)
```

Output:
{'Jang', 'Lee', 'Ryu', 'Kim', 'Park'}

차이점이 무엇인가요?

정렬

```
print(sorted(members))
```

['Jang', 'Kim', 'Lee', 'Park', 'Ryu']

Python — whois 조회하기

whois 모듈을 쓸수 있나요? python 콘솔에서 테스트 합니다.

```
>>> import whois
```

Import 오류가 나면 명령 프롬프트로 나와 다음과 같이 pip로 python-whois 모듈을 설치합니다.

```
>>> import whois
```

python 콘솔로 들어가 whois 모듈을 import하고 www.openbase.co.kr whois 조회를 해봅니다.

```
>>> import whois
>>> whois.whois("www.openbase.co.kr")
{'domain_name': 'openbase.co.kr', 'registrant_org': 'Openbase', 'registrant_address': '4F, Hibrand, 16 Maeheon-ro Seocho-gu, Seoul, KR', 'registrant_zip': '06771', 'admin_name': 'OPENBASE, Inc.', 'admin_email': 'isjang@openbase.co.kr', 'admin_phone': '02-3404-5613', 'creation_date': datetime.datetime(1998, 12, 7, 0, 0), 'updated_date': datetime.datetime(2017, 10, 16, 0, 0), 'expiration_date': datetime.datetime(2026, 10, 15, 0, 0), 'registrar': 'Inames Co., Ltd.(http://www.inames.co.kr)', 'name_servers': ['dns1.openbase.co.kr', 'dns.openbase.co.kr']}
```

팀장이 대량의 도메인 목록을 주고 whois 조회 결과를 정리하라고 합니다. 이제 실행을 반복해서 해야 합니다.

콘솔에서 타이핑하지 말고 일단 스크립트 파일로 만듭니다. 안 그러면 맨날 쳐야 하니까... 스크립트 파일로 만들면서 반복 실행하는 부분을 `whois_check()`라는 함수로 만듭니다.

```
import whois

def whois_check(address):
    result = whois.whois(address)
    return str(result)

ret1 = whois_check("www.openbase.co.kr")
print(ret1)
ret2 = whois_check("www.naver.com")
print(ret2)
ret3 = whois_check("www.google.com")
# 팀장이 준 약 3천개의 도메인을 하나씩 호출...?
```

팀장이 준 목록의 도메인에 대해 하나씩 다 체크 함수를 호출해야 할까요?

팀장은 도메인들을 `comma(,)`로 분리한 문서자료로 줬습니다. 그걸 활용해 보겠습니다.

팀장이 준 comma(,)로 분리된 도메인 목록을 리스트로 만듭니다. 특정 문자를 단위로 문자열을 잘라 리스트로 만드는 split()함수를 이용합니다.

그리고 리스트의 각 요소들을 돌면서 하나씩 whois 조회 함수를 호출합니다.

```
import whois

def whois_check(address):
    result = whois.whois(address)
    return str(result)

order =
"www.naver.com,www.google.com,www.openbase.co.kr,www.daum.net,www.seoul.go.kr,www.seoul.go.kr,w
ww.seoul.go.kr"
todo_list = order.split(',')

for address in todo_list:
    ret = whois_check(address)
    print(ret)
```

근데 팀장이 제대로 정리를 안 해서 목록에 같은 도메인이 여러 번 나옵니다. 어떻게 해야 할까요?

팀장이 준 도메인들을 리스트로 만든 `todo_list`를 `set`으로 변환합니다. `set`으로 변신하는 순간 중복자료가 제거됩니다.

```
import whois

def whois_check(address):
    result = whois.whois(address)
    return str(result)

order="www.naver.com,www.google.com,www.openbase.co.kr,www.daum.net,www.seoul.go.kr,www.seoul.go.kr,www.seoul.go.kr"

todo_list = order.split(',')
new_todo_list = set(todo_list)

for address in new_todo_list:
    ret = whois_check(address)
    print(ret)
```

근데 이제 보니 팀장이 목록을 제대로 안 만들어서 빈 칸도 있고, 공백도 있습니다. 제대로 조회가 안 되는 항목도 있겠네요. 어떻게 해결할까요? (다음시간에 계속)

```
"www.naver.com, , www.google.com, ,www.openbase.co.kr,www.daum.net,www.seoul.go.kr,www.seoul.go.kr"
```

자료 :

https://secuwave.github.io/secure3/learn_script/main