

111- [PF] - 실습 - 목록, 튜플, 딕셔너리

목록, 튜플, 딕셔너리 작업

실습 개요

Python 에서 문자열과 숫자 데이터 유형은 *모음*이라는 그룹으로 사용되는 경우가 많습니다. Python 에서 지원되는 3 가지 모음은 목록, 튜플, 딕셔너리입니다.

본 실습에서는 다음을 수행합니다.

- 목록 데이터 유형 사용
- 튜플 데이터 유형 사용
- 딕셔너리 데이터 유형 사용

예상 완료 시간

45 분

연습 1: 목록 데이터 유형 소개

AWS Cloud9 IDE 액세스

1. 이 지침의 상단으로 이동한 다음 **Start Lab** 을 선택하여 실습 환경을 시작합니다.

Start Lab 패널이 열리고 실습 상태가 표시됩니다.

2. *Lab status: ready* 라는 메시지가 표시되면 **X** 를 선택하여 **Start Lab** 패널을 닫습니다.
3. 지침의 맨 위에서 **AWS** 를 선택합니다.

새 브라우저 탭에서 AWS 관리 콘솔이 열립니다. 시스템에 자동으로 로그인됩니다.

참고: 새 브라우저 탭이 열리지 않는 경우 일반적으로 브라우저에서 팝업 창을 열 수 없음을 나타내는 배너 또는 아이콘이 브라우저 상단에 표시됩니다. 배너 또는 아이콘을 선택하고 **Allow pop ups** 를 선택합니다.

4. AWS 관리 콘솔에서 **Services > Cloud9** 을 선택합니다. **Your environments** 패널에서 **reStart-python-cloud9** 카드를 찾아 **Open IDE** 를 선택합니다.

AWS Cloud9 환경이 열립니다.

참고: *.c9/project.settings have been changed on disk* 라는 메시지가 담긴 팝업 창이 표시되면 **Discard** 를 선택하여 무시합니다. 마찬가지로, *Show third-party content* 라는 대화 창이 나타나면 **No** 를 선택하여 거절합니다.

Python 연습 파일 생성

5. 메뉴 모음에서 **File > New From Template > Python File** 을 선택합니다.

이 작업은 제목이 없는 파일을 생성합니다.

6. 템플릿 파일에 제공된 샘플 코드를 삭제합니다.
7. **File > Save As...**를 선택하고, 연습 파일에 적절한 이름(예: *collections.py*)을 입력한 다음 **/home/ec2-user/environment** 디렉터리에 저장합니다.

터미널 세션에 액세스

8. AWS Cloud9 IDE 에서 + 아이콘을 선택하고 **New Terminal** 을 선택합니다.

터미널 세션이 열립니다.

9. 현재 작동 중인 디렉터리를 표시하려면 `pwd` 를 입력합니다. 이 명령은 **/home/ec2-user/environment** 를 가리킵니다.
10. 이 디렉터리에서 이전 섹션에서 생성한 파일을 찾을 수 있어야 합니다.

이 활동에서는 Python 스크립트를 편집하여 과일 이름의 모음, 또는 과일 목록을 만듭니다.

11. IDE의 탐색 창에서 이전 *Python 연습 파일 생성* 섹션에서 생성한 **.py** 파일을 선택합니다.
12. 파일에 다음 코드를 입력합니다.

```
myFruitList = ["apple", "banana", "cherry"]  
print(myFruitList)  
print(type(myFruitList))
```

13. 파일을 저장하고 실행합니다.
14. 스크립트가 올바르게 실행되고 출력이 예상한 대로 표시되는지 확인합니다.

위치별 목록 액세스

위치별로 목록에 액세스할 수 있습니다. 이 활동에서는 목록의 각 항목을 위치별로 출력합니다.

15. 프로그래밍 언어에서 목록 위치는 0부터 시작합니다. 대괄호는 목록에서 사용자가 원하는 위치를 Python에 알립니다. **apple** 문자열에 액세스하려면 다음 코드를 입력합니다.

```
print(myFruitList[0])
```

16. **banana** 문자열에 액세스하려면 다음 내용을 입력합니다.

```
print(myFruitList[1])
```

17. **cherry** 문자열에 액세스하려면 다음 코드를 입력합니다.

```
print(myFruitList[2])
```

18. 파일을 저장하고 실행합니다.
19. 스크립트가 올바르게 실행되고 출력이 예상한 대로 표시되는지 확인합니다.

목록의 값 변경

목록의 값을 변경할 수 있습니다. 이 활동에서는 **cherry**를 **orange**로 변경합니다.

20. Python 에서 목록 위치는 0 부터 시작하므로 3 번째 위치에 액세스하려면 숫자 2를 사용해야 합니다. 다음 코드를 입력합니다.

```
myFruitList[2] = "orange"
```

21. 업데이트된 목록을 출력합니다.

```
print(myFruitList)
```

22. 파일을 저장하고 실행합니다.

23. 스크립트가 올바르게 실행되고 출력이 예상한 대로 표시되는지 확인합니다.

```
['apple', 'banana', 'cherry']  
<class 'list'>  
apple  
banana  
cherry  
['apple', 'banana', 'orange']
```

연습 2: 튜플 데이터 유형 소개

튜플 정의

튜플은 목록과 같지만 변경할 수 없습니다. 생성 후 변경할 수 없는 데이터 유형은 *변경 불가능*하다고 말합니다. 튜플을 정의하려면 대괄호([]) 대신 소괄호를 사용합니다.

24. 다음 코드를 입력하여 튜플을 생성합니다.

```
myFinalAnswerTuple = ("apple", "banana", "pineapple")  
print(myFinalAnswerTuple)  
print(type(myFinalAnswerTuple))
```

25. 파일을 저장하고 실행합니다.

26. 스크립트가 올바르게 실행되고 출력이 예상한 대로 표시되는지 확인합니다.

위치별 튜플 액세스

목록과 마찬가지로 튜플의 항목은 위치별로 액세스할 수 있습니다.

27. `apple` 문자열에 액세스하려면 다음 코드를 입력합니다.

```
print(myFinalAnswerTuple[0])
```

28. `banana` 문자열에 액세스하려면 다음 코드를 입력합니다.

```
print(myFinalAnswerTuple[1])
```

29. `pineapple` 문자열에 액세스하려면 다음 코드를 입력합니다.

```
print(myFinalAnswerTuple[2])
```

30. 파일을 저장하고 실행합니다.

31. IDE 창의 상단 근처에서 **Run**(Play) 버튼을 선택합니다.

32. 스크립트가 올바르게 실행되고 출력이 예상한 대로 표시되는지 확인합니다.

```
['apple', 'banana', 'cherry']  
<class 'list'>  
apple  
banana  
cherry  
['apple', 'banana', 'orange']  
( 'apple', 'banana', 'pineapple' )  
<class 'tuple'>  
apple  
banana  
pineapple
```

연습 3: 딕셔너리 데이터 유형 소개

딕셔너리 정의

딕셔너리는 위치(키)의 이름이 지정된 목록입니다. 목록에서 사람들이 좋아하는 과일을 나열한다고 생각해 보십시오.

33. Python 스크립트로 돌아가 다음 코드를 입력합니다.

```
myFavoriteFruitDictionary = {  
    "Akua" : "apple",  
    "Saanvi" : "banana",  
    "Paulo" : "pineapple"  
}
```

34. `print()` 함수를 사용하여 셸에 딕셔너리를 작성합니다.

```
print(myFavoriteFruitDictionary)
```

35. `type()` 함수를 사용하여 셸에 데이터 유형을 작성합니다.

```
print(type(myFavoriteFruitDictionary))
```

36. 파일을 저장하고 실행합니다.

37. 스크립트가 올바르게 실행되고 출력이 예상한 대로 표시되는지 확인합니다.

이름별 딕셔너리 액세스

이 활동에서는 숫자가 아닌 개인의 이름을 사용하여 그들이 좋아하는 과일을 가져옵니다.

38. Akua 가 좋아하는 과일에 액세스하려면 다음 코드를 입력합니다.

```
print(myFavoriteFruitDictionary["Akua"])
```

39. Saanvi 가 좋아하는 과일에 액세스하려면 다음 코드를 입력합니다.

```
print(myFavoriteFruitDictionary["Saanvi"])
```

40. Paulo 가 좋아하는 과일에 액세스하려면 다음 코드를 입력합니다.

```
print(myFavoriteFruitDictionary["Paulo"])
```

41. 파일을 저장하고 실행합니다.

42. 스크립트가 올바르게 실행되고 출력이 예상한 대로 표시되는지 확인합니다.

```
['apple', 'banana', 'cherry']  
<class 'list'>  
apple  
banana  
cherry  
['apple', 'banana', 'orange']  
( 'apple', 'banana', 'pineapple' )  
<class 'tuple'>  
apple  
banana  
pineapple  
{ 'Akua': 'apple', 'Saarvi': 'banana', 'Paulo': 'pineapple' }  
<class 'dict'>  
apple  
banana  
pineapple
```

축하합니다! Python 에서 목록, 튜플, 딕셔너리 데이터 유형을 작업했습니다.

실습 종료

축하합니다! 실습을 마치셨습니다.

43. 이 페이지의 상단에서 **End Lab** 을 선택한 다음 Yes 를 선택하여 실습 종료를 확인합니다.

*DELETE has been initiated... You may close this message box now.*라는 내용의 패널이 표시됩니다.

44. *Ended AWS Lab Successfully* 라는 메시지가 잠시 표시되어 실습이 종료되었음을 나타냅니다.

추가 리소스

AWS Training and Certification 에 대한 자세한

내용은 <https://aws.amazon.com/training/>을 참조하십시오.

여러분의 피드백을 환영합니다. 제안이나 수정 사항을 공유하려면 [AWS Training and Certification Contact Form](#) 에서 세부 정보를 제공해 주십시오.

© 2022 Amazon Web Services, Inc. 및 계열사. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다. 상업적인 복제, 대여 또는 판매는 금지됩니다.