126- [PF] - 실습 - 인슐린에 대한 정보를 검색하기 위한 파일 핸들러 및 모듈

인슐린에 대한 정보를 검색하기 위한 파일 핸들러 및 모듈 생성

실습 개요

본 실습에서는 다음을 수행합니다.

- 모듈 생성
- 파일을 열고 Python 에 내장된 JSON 모듈을 사용하여 포함된 JSON 데이터 로드
- JSON 구조를 구문 분석하여 인슐린 데이터에 액세스
- 주어진 코드를 사용하여 인간 인슐린의 대략적인 분자 질량을 계산(실습 **Python 에서 문자열 시퀀스 및 질량 수 작업**과 유사함)

예상 완료 시간

25 분

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 파일에서 샘플 코드를 삭제합니다.
7. **File -> Save As...**를 선택하고, 연습 파일에 적절한 이름(예: calc_weight_json.py)을 입력한 다음 **/home/ec2-user/environment** 디렉터리에 저장합니다.
8. 두 번째 파일을 생성하고 jsonFileHandler.py 라는 이름을 지정합니다.

   참고: **.py** 는 Python 파일의 확장자입니다.

9. **files** 라는 디렉터리를 생성합니다.

터미널 세션에 액세스

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

    터미널 세션이 열립니다.

11. 현재 작동 중인 디렉터리를 표시하려면 `pwd` 를 입력합니다. 이 명령은 **/home/ec2-user/environment** 를 가리킵니다.
12. 이 디렉터리에서 이전 섹션에서 생성한 파일을 찾습니다.

연습 1: JSON 분자 데이터 파일 생성

이 JSON 문서는 인슐린 분자, 아미노산의 질량 수, 인슐린 분자의 실제 질량과 같은 이전 실습의 모든 정보를 저장합니다.

13. 메뉴 모음에서 **File -> New File** 을 선택합니다.
14. 다음 코드를 복사하여 새로 생성한 파일에 붙여 넣습니다.

```
{
    "molecules":{
        "lsInsulin":"malwmrllpllallalwgpdpaaa",
        "bInsulin":"fvnqhlcgshlvealylvcgergffytpkt",
        "aInsulin":"giveqcctsicslyqlenycn",
        "cInsulin":"rreaedlqvgqvelgggpgagslqplalegslqkr"
    },
    "weights":{
        "A":89.09,
        "C":121.16,
        "D":133.10,
        "E":147.13,
        "F":165.19,
        "G":75.07,
        "H":155.16,
        "I":131.17,
        "K":146.19,
        "L":131.17,
        "M":149.21,
        "N":132.12,
        "P":115.13,
        "Q":146.15,
        "R":174.20,
        "S":105.09,
        "T":119.12,
        "V":117.15,
        "W":204.23,
        "Y":181.19
    },
```

```
    "molecularWeightInsulinActual":5807.63
}
```

15. **files** 폴더에 **insulin.json** 으로 파일을 저장하려면 **File -> Save As...**를 선택합니다.
16. **Save As** 팝업 창에서 **Filename:**에 `insulin.json` 을 입력합니다.
17. **Folder:**에 `files` 를 입력하거나 **files** 폴더를 선택합니다.

연습 2: JSON 파일 핸들러 모듈 생성

이 과제에서는 JSON 파일을 읽고 JSON 문서를 반환하는 모듈을 생성합니다.

18. **jsonFileHandler.py** 파일을 선택합니다.
19. JSON 을 가져와 작업을 시작합니다.

```
import json
```

20. 파일을 읽을 함수를 정의합니다.

```
def readJsonFile(fileName):
```

21. 파일 정의 아래에서 데이터 변수를 빈 문자열로 추가합니다.

```
data=""
```

22. 함수 본문에 대해 **open** 함수를 사용하여 json 파일을 열고, **json.load** 를 사용하여 파일을 구문 분석합니다.

```
def readJsonFile(fileName):
    data = ""
    with open('files/insulin.json') as json_file:
        data = json.load(json_file)
    return data
```

open 은 **files/insulin.json** 파일을 파일 핸들러에 반환합니다.

json.load 는 JSON 파일을 읽고 해당 콘텐츠를 Python 딕셔너리로 반환합니다.

23. **try/except** 블록을 추가하여 이 함수를 더욱 신뢰할 수 있도록 만듭니다.

```
import json

def readJsonFile(fileName):
    data = ""
    try:
        with open(fileName) as json_file:
            data = json.load(json_file)
    except IOError:
        print("Could not read file")
    return data
```

파일을 열 수 없는 경우 프로그램은 *Could not read file* 오류를 표시합니다.

반환된 **data** 문자열은 파일 열기 메서드가 실패할 경우 비어 있습니다.

다른 Python 파일로 가져와 **readJsonFile** 함수에 액세스할 수 있는 **jsonFileHandle** 모듈을 생성했습니다.

연습 3: 메인 프로그램 생성

이전 실습에서처럼 JSON 데이터를 구문 분석하고 문자 질량을 계산하는 메인 프로그램을 만듭니다.

24. 먼저 **jsonFileHandle** 모듈을 가져옵니다. **calc_weight_json.py** 파일을 열고 다음을 추가합니다.

```
import jsonFileHandler
```

25. JSON 데이터를 검색하여 **data** 변수에 저장합니다.

```
data = jsonFileHandler.readJsonFile('files/insulin.json')
```

26. 반환된 데이터가 비어 있지 않은지 테스트하고 인슐린 데이터를 얻습니다.

```
if data != "" :
    bInsulin = data['molecules']['bInsulin']
    aInsulin = data['molecules']['aInsulin']
    insulin = bInsulin + aInsulin
    molecularWeightInsulinActual = data['molecularWeightInsulinActual']
    print('bInsulin: ' + bInsulin)
    print('aInsulin: ' + aInsulin)
    print('molecularWeightInsulinActual: ' + str(molecularWeightInsulinActual))
else:
    print("Error. Exiting program")
```

27. 프로그램을 실행하여 데이터가 제대로 검색되었는지 확인할 수 있습니다. 결과는 다음과 같아야 합니다.

```
bInsulin: fvnqhlcgshlvealylvcgergffytpkt
aInsulin: giveqcctsicslyqlenycn
molecularWeightInsulinActual: 5807.63
```

28. 파일을 찾을 수 없는 경우 무엇이 발생했는지 테스트할 수도 있습니다. 예를 들어, 파일 이름을 **'files/insuline.json'**으로 변경하고 프로그램을 실행합니다. 다음과 같은 메시지가 표시됩니다.

```
Could not read file
Error. Exiting program
```

29. 파일 이름을 다시 **files/insulin.json** 으로 되돌리도록 마지막 변경을 실행 취소합니다.
30. 코드의 **if** 섹션에서 마지막 **print** 아래에 다음 코드를 추가합니다.

```
# Calculating the molecular weight of insulin
# Getting a list of the amino acid (AA) weights
aaWeights = data['weights']
# Count the number of each amino acids
aaCountInsulin = ({x: float(insulin.upper().count(x)) for x in ['A','C','D', 'E', 'F', 'G', 'H', 'I', 'K', 'L', 'M', 'N', 'P', 'Q', 'R','S', 'T','V', 'W', 'Y']})
```

```
# Multiply the count by the weights
molecularWeightInsulin = sum({x: (aaCountInsulin[x]*aaWeights[x]) for x in
['A', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L', 'M', 'N', 'P', 'Q', 'R','S', 'T', 'V', 'W', 'Y']}.values())
print("The rough molecular weight of insulin: " +
str(molecularWeightInsulin))
print("Percent error: " + str(((molecularWeightInsulin -
molecularWeightInsulinActual)/molecularWeightInsulinActual)*100))
```

31. 프로그램을 실행합니다. 다음과 같은 결과가 표시됩니다.

```
bInsulin: fvnqhlcgshlvealylvcgergffytpkt
aInsulin: giveqcctsicslyqlenycn
molecularWeightInsulinActual: 5807.63
The rough molecular weight of insulin: 6696.420000000001
Percent error: 15.30383306099047
```

실습 종료

축하합니다! 실습을 마치셨습니다.

32. 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.

33. *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.