

122- [PF] - 실습 - Python 목록 및 루프를 사용하여 인슐린의 순전하 계산

Python 목록 및 루프를 사용하여 인슐린의 순전하 계산

실습 개요

흐름 제어 모듈에서는 if-else 스테이트먼트, while 루프, 목록 및 for 루프에 대해 배웠습니다. 이제 학습한 내용을 인간 인슐린의 실제 응용 분야에 적용해 보겠습니다.

여기에서는 lists, for, while 루프와 기본 산수를 사용하여 pH0~pH14 사이의 인슐린 순전하를 계산합니다.

본 실습에서는 다음을 수행합니다.

- 순전하 계산에 사용할 pKa 값(산의 강도를 나타냄)의 딕셔너리 생성
- count() 메서드를 사용하여 아미노산의 수 얻기
- while 루프를 사용하여 pH0~pH14 사이의 인슐린 순전하 계산

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: *net-charge.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.

연습 1: 변수, 목록 및 딕셔너리 할당

11. IDE의 탐색 창에서 이전 *Python 연습 파일 생성* 섹션에서 생성한 파일을 선택합니다.
12. 다음 코드를 복사하여 파일에 붙여 넣고 파일을 저장합니다.

```
# Python3.6
# Coding: utf-8
# Store the human preproinsulin sequence in a variable called preproinsulin:
preproInsulin =
"malwmrllplllallwgpdpaaafvnqhlcgshlvealylvcgergfftyptktrreaedlqvgqvelgggpgagslqplaleg
slqkrgiveqcctsicslyqlenycn"
# Store the remaining sequence elements of human insulin in variables:
lsInsulin = "malwmrllplllallwgpdpaaa"
bInsulin = "fvnqhlcgshlvealylvcgergfftypkt"
aInsulin = "giveqcctsicslyqlenycn"
cInsulin = "rreaedlqvgqvelgggpgagslqplalegslqkr"
insulin = bInsulin + aInsulin
```

13. 다음 줄에서 `pKR = {}` 식을 입력하여 새 딕셔너리를 생성합니다.
14. 키-값 페어로 딕셔너리를 채우려면 1 번째 키 `y`를 `10.07` 값과 함께 삽입합니다. 중괄호 안에 커서를 놓고 `'y': 10.07,`을 입력합니다.

참고: 나머지 키-값 페어를 추가할 수 있도록 값 뒤에 쉼표를 포함했습니다.

15. 코드 세그먼트를 일치시키려면 다음 키-값 페어를 딕셔너리에 추가합니다.

- `'c': 8.18`
- `'k': 10.53`
- `'h': 6.00`
- `'r': 12.48`
- `'d': 3.65`
- `'e': 4.25`

딕셔너리의 모습은 다음 코드와 유사합니다.

```
pKR = {'y':10.07,'c': 8.18,'k':10.53,'h':6.00,'r':12.48,'d':3.65,'e':4.25}
```

참고: *Y, C, K, H, R, D, E*는 순전하 계산과 관련이 있는 유일한 아미노산입니다.

연습 2: count()를 사용하여 각 아미노산의 수 세기

이 연습에서는 count() 메서드 및 목록 내포를 사용하여 Y, C, K, H, R, D, E 아미노산의 수를 셉니다. 이러한 아미노산은 순전하에 기여합니다.

16. 목록 내 항목의 수를 파악하기 위해 count() 메서드를 사용할 수 있습니다. 인슐린에서 얼마나 많은 수의 아미노산이 Y인지 확인하려면 insulin.count("Y")를 입력하여 count() 메서드를 사용합니다.
17. 다음으로, count() 메서드가 반환한 변수를 float 로 캐스팅하여 insulin.count() 줄을 업데이트합니다(float(insulin.count("Y"))).
18. 단일 엔터티를 파악하는 기본적인 내용을 알아보았으니 이제 이 메서드를 사용하여 목록에서 모든 엔터티를 찾을 수 있습니다. 이 프로세스는 목록 내포를 사용하여 수행할 수 있습니다. 전체 줄에 대해 다음을 입력합니다. seqCount = ({x: float(insulin.count(x)) for x in ['y','c','k','h','r','d','e']})

참고: 이 연습의 처음 두 단계는 3 번째 단계의 선행 단계입니다.

연습 3: 순전하 수식 작성

이 연습에서는 순전하 수식을 구성합니다. Python 기반 순전하 수식에서 제공된 netCharge 변수를 사용합니다. 수식의 함수에는 pH 변수가 14 와 같거나 보다 낮을 때 순전하를 출력할 while 루프가 포함됩니다.

19. pH라는 변수를 만들고 pH = 0 을 입력하고 ENTER 키를 눌러 0 으로 초기화합니다.
20. while (pH <= 14):를 입력하고 ENTER 키를 눌러 while 루프를 생성합니다.
21. 다음 netCharge 변수를 복사하여 while 루프의 앞에 붙여 넣습니다.

```
netCharge = (  
    +(sum({x: ((seqCount[x]*(10**pKR[x]))/((10**pH)+(10**pKR[x]))})  
    for x in ['k','h','r'].values()))  
    -(sum({x: ((seqCount[x]*(10**pH))/((10**pH)+(10**pKR[x]))})  
    for x in ['y','c','d','e'].values()))
```

22. pH의 netCharge 변수를 출력하려면 더 나은 가독성을 위해 형식 문자열을 사용합니다. print('{0:.2f}'.format(pH), netCharge)를 입력하고 Enter 키를 누릅니다.
23. 마지막으로, pH +=1 을 입력하고 ENTER 키를 눌러 pH 변수를 증분합니다.

24. 파일을 저장하고 실행합니다.

Python 에서 들여쓰기와 공백에 주의

Python 코드의 하위 집합은 들여쓰기와 공백으로 구성됩니다. Python 에서 단 하나의 들여쓰기 또는 공백만 잘못되어도 예외 또는 기타 오류가 발생할 수 있습니다. 예를 들어 코드가 작동하려면 `while` 루프 내의 모든 항목이 적절히 들여쓰기되도록 해야 합니다.

축하합니다! Python 함수의 목록 및 루프 작업을 수행했습니다.

실습 종료

축하합니다! 실습을 마치셨습니다.

25. 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.

26. *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.