

120-[PF] - 실습 - 문자열 시퀀스 및 인슐린 숫자 가 중치

Python에서 문자열 시퀀스 및 질량 수 작업

실습 개요

Python 기본 사항 모듈에서 변수, 주석, 산수, 연결, 예외를 학습했습니다. 이제 학습한 내용을 인간 인슐린의 실제 응용 분야에 적용해 보겠습니다.

인간 프리프로인슐린의 단백질 서열을 문자열 변수로, 프리프로인슐린의 질량을 int 및 float 변수로 저장합니다. 그런 다음, 코드를 설명하는 주석과 함께 이 변수를 콘솔에 출력합니다. 기본 산수 및 문자열 연결을 수행합니다.

본 실습에서는 다음을 수행합니다.

- 코드의 의도와 흐름을 설명하는 주석 추가
- `print()`를 사용하여 콘솔에 Python 코드의 요소 출력
- 문자열 조작을 사용하여 프리프로인슐린에서 인슐린 서열 확보
- 분자 질량 및 인슐린 서열에 대한 기본 산수 수행
- 문자열, int, float 변수를 인슐린의 질량을 나타내는 숫자에 할당
- Python 예외 살펴보기

예상 완료 시간

30 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: *string-insulin.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. 이 디렉터리에서 이전 섹션에서 생성한 파일을 찾을 수 있어야 합니다.

연습 1: 인간 인슐린의 서열 요소에 변수 할당

이 연습에서는 변수를 생성하고 이 변수에 문자열 값을 할당합니다.

10. IDE 의 탐색 창에서 이전 *Python 연습 파일 생성* 섹션에서 생성한 파일을 선택합니다.

.py 파일을 시작하는 방법

항상 주석과 함께 Python 파일을 시작해야 합니다. Python 주석은 우물정자 기호(#)로 시작한다는 사실을 기억하십시오.

첫 번째 주석은 다음을 제공합니다.

- 해당되는 경우 실행 파일에 대한 경로가 있는 Python 버전(*python3.6*)
- 파일의 인코딩(일반적으로 *coding: utf-8*)

11. 다음 줄에 다음과 같은 주석을 작성합니다.

```
# Store the human preproinsulin sequence in a variable called
preproinsulin:
```

12. 등호(=)를 할당 연산자로 하고 `preproInsulin =`을 변수 이름으로 입력하여 Python 파일에서 1 번째 변수를 만듭니다.
13. 등호(=) 뒤에 다음과 같이 입력합니다.

```
"malwmrllplllallwgpdpaaafvnqhlcgshlvealylvcgergfftytpktr" ₩
"reaedlqvggqvelggpgagslqplalegslqrgiveqcctsicslyqlenycn"
```

14. 해당 줄에서 첫 번째 변수를 확정하려면 ENTER 키를 누릅니다.

Python 파일 및 기타 PEP 표준의 최대 줄 길이

이전 단계의 변수 값에서 추적 역슬래시(₩)는 Python Enhancement Proposals(PEP) 8 스타일 가이드의 규정 준수를 유지하기 위해 사용됩니다. PEP 8 스타일 가이드는 줄 당 최대 79 자의 문자를 권장합니다. PEP 는 Python 모범 사례의 표준입니다. 줄 길이가 더 길어도 파일은 여전히 실행되지만, 권장 글자 수로 제한하면 간편성과 가독성이 높아집니다. 백슬래시(₩)를 사용하면 변수와 코드를 더 작은 블록으로 분할하여 79 자 문자 제한을 유지할 수 있습니다.

15. 파일에 주석을 작성합니다.

```
# Store the remaining sequence elements of human insulin in variables:
```

16. 단계를 반복하여 변수를 정의하고 다음 차트의 정보를 사용하여 값을 할당합니다. 변수 이름과 문자열 사이에 등호(=)를 사용합니다.

Variable Name

lsInsulin
blInsulin
alInsulin

String to Save to Variable

"malwmrllplllallwgpdpaaa"
"fvnqhlcgshlvealylvcgergfftypkt"
"giveqcctsicslyqlenycn"

`clnsulin`

`"rreaedlqvqqvelgggpgagslqplaleg`

참고: 일반적으로 Python 에서 변수 이름은 소문자인 단어로 시작하며, 이 다음으로 밑줄 또는 공백 없이 대문자가 옵니다. 일관된 방식으로 변수의 이름을 지정하십시오.

17. 마지막으로, 보다 작은 인슐린 그룹을 *인슐린*이라는 단일 변수로 병합합니다. 이렇게 하려면 새 줄에 `insulin = blnsulin + alnsulin` 을 입력합니다.

연습 3: `print()`를 사용하여 콘솔에 인간 인슐린의 서열 표시

이 연습에서는 내장된 `print()` 메서드를 사용하여 콘솔에 인간 인슐린의 서열 요소를 표시합니다.

18. 다음 줄에 다음과 같은 주석을 작성합니다.

```
# Printing "the sequence of human insulin" to console using successive  
print() commands:
```

19. Python 파일의 새 줄에 `print("The sequence of human preproinsulin:")`를 입력합니다.

20. ENTER 키를 누릅니다.

이 `print()` 스테이트먼트는 형식 지정 없이 직접적으로 나타내는 제공된 문자열을 출력합니다.

21. 스크립트의 변수에 포함된 문자열을 출력하려면 `print(preproInsulin)`를 입력합니다.

22. ENTER 키를 누릅니다.

23. 다음 주석을 입력합니다.

```
# Printing to console using concatenated strings inside the print function
(one-liner):
```

24. 문자열을 연결하려면 `print()` 스테이트먼트에서 더하기 기호(+)를 사용합니다.

```
print("The sequence of human insulin, chain a: " + aInsulin)
```

25. ENTER 키를 누릅니다.

참고: 내장된 `print()` 함수는 5 단계와 동일한 태스크를 수행할 수 있는 여러 인수를 취합니다.

```
print("The sequence of human insulin, chain a:", aInsulin)
```

26. 파일을 저장하고 실행합니다.

연습 4: 주어진 코드를 사용하여 인간 인슐린의 대략적인 분자 질량 계산

이 실습에서는 이후 실습에서 작업할 인슐린의 분자 질량을 계산합니다.

27. **.py** 파일이 열려 있는지 확인합니다.

28. 다음 코드를 복사하여 **.py** 파일의 끝에 붙여 넣습니다.

```
# Calculating the molecular weight of insulin
# Creating a list of the amino acid (AA) weights
aaWeights = {'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}
```

```
# 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))
```

29. 파일을 저장하고 실행합니다.

30. 출력한 결과를 확인합니다. 이 코드의 요소를 사용하여 다른 실습에서 루프와 함수 작업을 수행합니다. 따라서 코드의 작성 방식을 관찰하고 예상 출력을 따르도록 합니다.

참고: 인간 인슐린의 실제 분자 질량은 5807.63 입니다. 하지만 프로그램은 특정 연결 및 번역 후 처리를 무시하기 때문에 6696.42 를 전달합니다. 오류를 백분율을 계산하려면 $error\ percentage = (|measured - accepted| / accepted) * 100\%$ 식을 사용합니다.

31. 스크립트에 예를 입력하거나 복사합니다.

```
molecularWeightInsulinActual = 5807.63
print("Error percentage: " + str((((molecularWeightInsulin -
molecularWeightInsulinActual)/molecularWeightInsulinActual)*100))
```

32. 오류 백분율을 확인하려면 파일을 실행하고 저장합니다.

참고: 부동 소수점 계산에 문자열 연결을 사용하면 `print()` 함수가 오류를 반환합니다. 이 오류는 Python 에 특정 데이터 유형을 사용하라고 알리는 *캐스팅*이라는 메서드에 의해 처리됩니다. 앞서 `str()` 함수를 사용한 것이 캐스팅의 예시입니다.

축하합니다! 변수와 Python 함수의 여러 데이터 유형을 사용하여 작업했습니다.

실습 종료

축하합니다! 실습을 마치셨습니다.

33. 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.

34. *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.