

109- [PF] - 실습 - 숫자 데이터 유형

숫자 데이터 유형 작업

실습 개요

Python 을 사용하여 간편하게 계산할 수 있습니다. 실제로 Python 은 대규모 데이터를 분석해야 하는 데이터 과학자 사이에서 인기 있는 언어입니다. 이 실습에서는 숫자 값을 저장하는 데 사용되는 기본 데이터 유형을 탐색해 봅니다.

본 실습에서는 다음을 수행합니다.

- Python 셸 사용
- int 데이터 유형 사용
- float 데이터 유형 사용
- 복소수 데이터 유형 사용
- bool 데이터 유형 사용

예상 완료 시간

60 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: *numeric-data.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: Python 셸 사용

터미널 탭에서 다음 명령을 입력하여 Python 셸을 시작할 수 있습니다.

```
python3
```

Python 셸은 다음 예와 유사해야 합니다.

```
Python 3.6.12 (default, Aug 31 2020, 18:56:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-28)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

3 개의 보다 큰 기호(>>>)가 사용자가 Python 명령을 입력할 수 있는 프롬프트를 나타냅니다. 다음 활동에서는 몇몇 숫자 명령을 실행하여 Python 셸 사용을 연습합니다.

더하기

11. 다음과 같이 입력합니다.

```
2 + 2
```

12. ENTER 키를 누릅니다.

13. 4가 출력으로 표시되는지 확인합니다.

빼기

14. 다음과 같이 입력합니다.

```
4 - 2
```

15. ENTER 키를 누릅니다.

16. 2가 출력으로 표시되는지 확인합니다.

곱하기

곱셈을 수행하려면 * 기호를 사용합니다.

17. 다음과 같이 입력합니다.

```
2 * 2
```

18. ENTER 키를 누릅니다.

19. 4가 출력으로 표시되는지 확인합니다.

나누기

나누기를 수행하려면 / 기호를 사용합니다.

20. 다음과 같이 입력합니다.

```
4 / 2
```

21. ENTER 키를 누릅니다.

22. 2.0이 출력으로 표시되는지 확인합니다.

Python 셸 종료

23. Python 셸을 종료하려면 다음 명령을 입력합니다.

```
quit()
```

연습 2: int 데이터 유형 소개

데이터 유형에 대해 자세히 알아보기 위해 내장된 함수를 사용합니다. *함수*는 이름으로 재사용할 수 있는 코드 조각입니다. 다음을 수행하여 함수를 사용합니다.

- 이름으로 호출
- 소괄호 안에 묶인 *인수*라는 하나 이상의 입력 목록을 포함

Python에는 보다 유용한 프로그램을 작성하는 데 사용할 수 있는 여러 함수가 내장되어 있습니다.

이 함수 모음을 *라이브러리*라고 합니다. Python에 내장된 함수 모음은 *Python 표준 라이브러리*라고 합니다.

Python 파일 편집

Python 셸에 명령을 하나씩 입력하는 대신, 명령 시퀀스를 포함하는 텍스트 파일을 편집합니다.

24. IDE의 탐색 창에서 이전 *Python 연습 파일 생성* 섹션에서 생성한 파일을 선택합니다.

25. 파일에 다음 코드를 입력합니다.

```
print("Python has three numeric types: int, float, and complex")
```

26. 파일을 저장하려면 **File > Save**를 선택합니다.

27. IDE 창의 상단에서 **Run(Play 버튼)**을 선택합니다.

28. IDE의 하단(콘솔) 창에서 프로그램이 *Python has three numeric types: int, float, and complex*라는 메시지를 출력하는지 확인합니다.

참고: 콘솔 출력을 보려면 위로 스크롤해야 할 수 있습니다.

29. 터미널 탭에서 이 실습을 위해 생성한 파일 이름인 *<lab-python-file-name>*으로 다음 명령을 입력하여 프로그램을 실행할 수도 있습니다.

```
python3 <lab-python-file-name>.py
```

30. 작성한 텍스트가 표준 출력에 나타나는지 확인합니다.

31. ~ \$ `python3 <lab-python-file-name>.py`

```
Python has three numeric types: int, float, and complex
```

변수 생성

변수는 정보를 저장하는 레이블이 지정된 상자과 같습니다. 상자의 내용을 변경할 수는 있지만, 레이블은 동일하게 유지됩니다. 이 활동에서는 변수 이름 *myValue*를 사용하고 해당 레이블이 지정된 상자에 다른 데이터 유형을 저장합니다.

31. Python 파일로 돌아가 새로운 줄에서 다음 코드를 입력합니다.

```
myValue=1
```

32. `print()` 함수를 사용하여 셀에 변수 값을 작성합니다. 프로그래밍 맥락에서 쓰기/란 셀에 정보를 추가하는 것을 의미합니다.

```
print(myValue)
```

33. 변수의 데이터 유형을 가져오려면 내장된 `type()` 함수를 사용합니다.

```
print(type(myValue))
```

34. 숫자와 텍스트를 결합하려면 인수를 *문자열*이라고 하는 문자의 모음으로 변환하는 내장된 `str()` 함수를 사용합니다. 이 경우에는 `int`(정수) 데이터 유형을 *문자열* 데이터 유형으로 변환합니다.

```
print(str(myValue) + " is of the data type " + str(type(myValue)))
```

35. 파일을 저장합니다.

36. 파일을 실행하려면 **Run** 을 선택합니다.

37. IDE 의 하단 창에서 올바른 출력이 표시되는지 확인합니다.

38. Python has three numeric types: int, float, and complex

39. 1

40. <class 'int'>

41. 1 is of the data type <class 'int'>

~ \$

참고: 출력을 보려면 위로 스크롤해야 할 수 있습니다.

연습 3: float 데이터 유형 소개

`int` 데이터 유형은 정수만을 저장합니다. 3.14와 같이 소수가 포함된 숫자를 저장하려면 *float* 라는 새로운 데이터 유형이 필요합니다.

38. Python 파일로 돌아가 새로운 줄에서 다음 코드를 입력합니다.

```
myValue=3.14
```

39. 셀에 변수 값을 작성하려면 `print()` 함수를 사용합니다.

```
print(myValue)
```

40. 내장된 `type()` 함수를 사용하여 변수의 데이터 유형을 가져옵니다.

```
print(type(myValue))
```

41. 숫자와 텍스트를 결합하려면 내장된 `str()` 함수를 사용합니다.

```
print(str(myValue) + " is of the data type " + str(type(myValue)))
```

42. 파일을 저장합니다.

43. 파일을 실행하려면 **Run** 을 선택합니다.

44. IDE 의 하단 창에서 다음 출력이 표시되는지 확인합니다.

```
Python has three numeric types: int, float, and complex
1
<class 'int'>
1 is of the data type <class 'int'>
3.14
<class 'float'>
3.14 is of the data type <class 'float'>
~ $
```

참고: 출력을 보려면 위로 스크롤해야 함을 기억하십시오.

연습 4: 복소수 데이터 유형 소개

고급 산수에서 허수는 허수 단위 j 로 곱하여 실수로 작성할 수 있는 복소수입니다. 복소수 데이터 유형은 $5j$ 와 같이 문자와 숫자를 나타내야 하기 때문에 복잡합니다.

45. Python 파일로 돌아가 다음 코드를 입력합니다.

```
myValue=5j
```

46. `print()` 함수를 사용하여 변수 값을 작성합니다.

```
print(myValue)
```

47. `type()` 함수를 사용하여 변수의 데이터 유형을 가져옵니다.

```
print(type(myValue))
```

48. 숫자와 텍스트를 결합하려면 내장된 `str()` 함수를 사용합니다.

```
print(str(myValue) + " is of the data type " + str(type(myValue)))
```

49. 파일을 저장합니다.

50. 파일을 실행하려면 **Run** 을 선택합니다.

51. IDE 의 하단 창에서 올바른 출력이 표시되는지 확인합니다.

```
Python has three numeric types: int, float, and complex
```

```
1
```

```
<class 'int'>
```

```
1 is of the data type <class 'int'>
```

```
3.14
```

```
<class 'float'>
```

```
3.14 is of the data type <class 'float'>
```

```
5j
```

```
<class 'complex'>
```

```
5j is of the data type <class 'complex'>
```

```
~ $
```

참고: 출력을 보려면 위로 스크롤해야 함을 기억하십시오.

연습 5: bool 데이터 유형 소개

`bool`(부울) 데이터 유형은 *True* 와 *False* 라는 영구 이름으로 구성되며, 이들은 숫자 *1* 과 *0* 으로 나타냅니다. 여기서 *1 = True* 이며 *0 = False* 입니다. `bool` 데이터 유형은 `int` 의 하위 집합으로

구현되며 실제 데이터 유형으로 간주되지 않습니다. 그러나 일부 프로그래밍 언어에서는 다른 데이터 유형으로 구현됩니다. 이 연습에서는 Python bool 이 *가짜 데이터 유형*을 호출합니다.

52. 텍스트 파일로 돌아가 다음 코드를 입력합니다.

```
myValue=True
```

53. print() 함수를 사용하여 셀에 변수 값을 작성합니다.

```
print(myValue)
```

54. 내장된 type() 함수를 사용하여 변수의 데이터 유형을 가져옵니다.

```
print(type(myValue))
```

55. 숫자와 텍스트를 결합하려면 내장된 str() 함수를 사용합니다.

```
print(str(myValue) + " is of the data type " + str(type(myValue)))
```

56. 파일을 저장합니다.

57. **Run(Play)** 버튼을 선택합니다.

58. IDE 의 하단 창에서 올바른 출력을 표시하는지 확인합니다.

59. .py 파일로 돌아가 다음 코드를 입력합니다.

```
myValue=False
```

60. print() 함수를 사용하여 셀에 변수 값을 작성합니다.

```
print(myValue)
```

61. 변수의 데이터 유형을 가져오려면 내장된 type() 함수를 사용합니다.

```
print(type(myValue))
```

62. 숫자와 텍스트를 결합하려면 내장된 str() 함수를 사용합니다.

```
print(str(myValue) + " is of the data type " + str(type(myValue)))
```

63. 파일을 저장합니다.

64. **Run(Play** 버튼)을 선택합니다.

65. IDE 의 하단 창에서 올바른 출력이 표시되는지 확인합니다.

```
Python has three numeric types: int, float, and complex
```

```
1
```

```
<class 'int'>
```

```
1 is of the data type <class 'int'>
```

```
3.14
```

```
<class 'float'>
```

```
3.14 is of the data type <class 'float'>
```

```
5j
```

```
<class 'complex'>
```

```
5j is of the data type <class 'complex'>
```

```
True
```

```
<class 'bool'>
```

```
True is of the data type <class 'bool'>
```

```
False
```

```
<class 'bool'>
```

```
False is of the data type <class 'bool'>
```

```
~ $
```

축하합니다! int, float, 복합이라는 Python 의 3 가지 숫자 데이터 유형을 학습했습니다.

또한 *bool*이라는 Python 가짜 데이터 유형을 소개했습니다. bool 은 실제로 *True* 및 *False* 값을 나타내는 숫자 0과 1 입니다.____

실습 종료

축하합니다! 실습을 마치셨습니다.

66. 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.

67. *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.