국민대학교 인공지능 연구실 김인경

파이썬기초1주차

목차



INTRO

- ▶ 참고 강의:
 - ▶ Inflearn '김왼손의 미운코딩새끼: 4시간만에 끝내는 파이썬 기초'
- ▶ 참고 서적:
 - ▶ 파이썬으로 배우는 머신러닝의 교과서(한빛미디어)
- ▶ 시간 및 요일
 - ▶ 수요일 20:00~ 21:00
- ▶ 연락처
 - kim_jenn@kookmin.ac.kr
- ▶ 1주차 colab 실습 링크
 - https://colab.research.google.com/drive/1wgHQKrv2Q6bAjhfKZFItB785nGuEVw-Q

파이썬 설치하기

- ▶ '김왼손의 미운코딩새끼'의 첫번째 강의에 파이썬 설치하기
- ▶ '파이썬으로 배우는 머신러닝의 교과서' 1장 참고(아나콘다 설치)
- 주피터 노트북 설치하기
 - https://m.blog.naver.com/itisik/221781721674
- ▶ 구글 colab 사용법
 - https://datascienceschool.net/view-notebook/f9d9fddb7cc7494a9e4be99f0e137be0/

숫자와 사칙연산

- ▶ 숫자의 종류
 - ▶ int: 정수를 표현하는 자료형(ex. 123)
 - ▶ float: 실수를 표현하는 자료형(ex. 1.23)
 - ▶ complex: 복소수를 표현하는 자료형(ex. 5 + 4i)
- 사칙연산
 - 덧셈(+), 뺄셈(-), 곱셈(*), 나눗셈(/)
 - 나누기 연산 몫(//), 나머지(%)
 - ▶ 다양한 축약형태 지원(+=, -=, *=, /=, //=, %=)

숫자와 사칙연산

- '='와 '=='의 차이점
 - ▶ '='는 할당 연산자를 의미
 - › '=='는 좌항과 우항이 동일함을 의미
 - Ex) a = 0
 - ▶ a에 0을 저장한다는 의미 즉, 좌항에 우항의 값을 대입
 - Ex) a == 0
 - ▶ a의 값이 0과 동일한지

입력과 출력

- input()
 - ▶ 사용자로부터 입력받는 함수
- print()
 - ▶ 주피터 노트북에서 셀의 마지막 행은 출력되지만 중간에 출력하고 싶다면 print함수를 사용 해야 함
 - ▶ format함수를 사용하여 문자열과 수치를 조합할 수 있음
 - ▶ print() 내부에 <'문자열'.format(x)>는 <문자열 중 {0} 부분을 x의 내용으로 대체한다>는 것을 의미함

변수

- ▶ 알파벳을 이용하여 변수를 나타낼 수 있음
- ▶ 변수에는 값을 저장할 수 있기 때문에 이를 사용하여 계산할 수 있음
- ▶ 식별자
 - ▶ 파이썬에서 이름을 붙일 때 사용하는 단어
 - ▶ 변수, 함수, 모듈, 클래스, 객체를 식별하는데 사용되는 이름
 - ▶ 문자, 숫자, ''로만 이루어짐
 - ▶ 대문자와 소문자는 구별됨(teacher와 TEAHCER는 다른 식별자)
 - ▶ 시작은 문자와 '_'만 가능 ─> 4fun (불가능)

변수

- ▶ 식별자
 - ▶ 중간에 스페이스를 넣어서 생성할 수 없음 & '.'를 제외한 다른 특수기호는 사용할 수 없음
 - ▶ 스네이크 케이스(snake_case)
 - ▶ 단어 사이에 '_' 기호를 붙여서 만든 식별자
 - ▶ 주로 변수, 함수 이름으로 사용
 - ▶ 캐멀 케이스(CamelCase)
 - ▶ 단어들의 첫 글자를 대문자로 만든 식별자
 - ▶ 주로 클래스 이름으로 사용
- ▶ 키워드, 예약어(keyword, reserved word)
 - ▶ 파이썬이 만들어질 때 정해진 단어들로 변수 명으로 사용하지 않음

문자열

- ▶ 문자열을 다루는 데에는 str형이 사용됨
- ▶ 작은 따옴표 혹은 큰따옴표로 둘러싸면 문자열로 인식함
- ▶ 문자열에도 사칙연산을 사용할 수 있음
 - ▶ 하지만 자료형을 동일하게 해야 함 ─> 숫자 + 문자열의 형태를 출력할 수 없음
 - > colab 예시 참고

- ▶ 아이템(item)들을 모은 순차 데이터 타입 ─> 아이템은 int, str, 다른 리스트 등 모든 종류의 데이터가 가능
- ▶ []를 사용해서 리스트를 생성함
- ▶ 리스트의 인덱스는 0부터 시작함
 - ex. myList = [1, 2, 3, 4, 5] —> myList[0] —> 1
- ▶ 리스트 자르기
 - ▶ 리스트이름[a:b] ─> a번째부터 (b-1)번째 아이템으로 구성된 부분 리스트가 결과값으로 얻어짐
 - ex. myList[2: 4] —> [3, 4]
 - ex. myList[1:] —> [2, 3, 4, 5]
 - ex. myList[: 3] —> [1, 2, 3]
 - ▶ ex. myList[:] —> [1, 2, 3, 4, 5] (리스트 복사)

- ▶ 리스트 연산자
 - +:두리스트를 합침
 - *: 리스트를 숫자만큼 반복
- ▶ 리스트 크기
 - len(리스트)
- ▶ 리스트 수정하기
 - ▶ 인덱스를 사용해서 대상 아이템을 변경할 수 있음

- range
 - ▶ range(시작숫자, 끝숫자 + 1)을 사용하면 시작 숫자부터 끝 숫자까지의 정수 데이터를 만들 수 있음
 - \rightarrow ex. range(5, 10) \longrightarrow 5, 6, 7, 8, 9
 - range형은 list형을 사용하여 요소를 수정할 수 있는 list형으로 변환할 수 있음
 - ex. list(range(5, 10)) —> [5, 6, 7, 8, 9]

- ▶ append()사용
 - ▶ 사용법: listName.append(item)
 - ▶ 리스트 뒤에 아이템을 추가
- ▶ insert()사용
 - ▶ 사용법: listName.insert(index, item)
 - ▶ 리스트 내 원하는 위치에 아이템 추가
- ▶ extend()사용
 - ▶ 사용법: listName.extend([item1, item2, item3])
 - ▶ 여러 아이템을 리스트 뒤에 추가

- ▶ remove()사용
 - ▶ 사용법: listName.remove(item)
 - ▶ 리스트의 특정 값을 제거, 만약 해당 값이 복수 개인 경우 제일 먼저 발견되는 값을 제거
- del사용
 - ▶ 사용법: del listName[index]
 - ▶ 리스트의 특정 인덱스에 있는 요소 제거
- ▶ pop()사용
 - ▶ 사용법: listName.pop()
 - ▶ 리스트의 마지막 항목을 제거 및 리턴(return)
- ▶ clear()사용
 - ▶ 사용법: listName.clear()
 - ▶ 리스트 내부의 모든 요소를 제거

- ▶ 정렬
 - ▶ 리스트의 아이템들을 작은 순서 혹은 큰 순서로 정렬할 수 있음
 - ▶ default는 숫자 크기 순서, 사전 등장 순서
 - ▶ 사용법1: listName.sort()
 - ▶ sort(reverser=True)를 하면 정렬 순서를 반대로 할 수 있음
 - ▶ sort()를 적용하면 원본이 유지되지 않고 정렬이 되기 때문에 원본을 유지하고 싶으면 복사본을 사용해서 정렬해야 함 ─> [:] 사용 가능
 - ▶ 사용법2: sorted(listName)
 - ▶ 정렬된 복제 리스트를 리턴 ─> 원본 유지
 - ▶ 때문에 리턴 값을 받을 변수를 지정한 후 사용해야 함
 - ex. newList = sorted(myList)

- ▶ 2차원 배열 형태를 만들 수 있음 ─> 리스트로 구 성된 리스트(행렬)
 - ▶ 요소의 참조를 변수명[i][j]와 같이 입력함
 - ▶ 총 3개의 리스트를 생성할 수 있음
 - ▶ Bob의 점수 리스트, Alice의 점수리스트, Carole의 점수리스트
 - ▶ 셋의 리스트를 합친 모든 사람의 점수를 포함 하는 리스트를 생성
 - ▶ 인덱스를 통해서 접근 가능

점수표

	90	수학	컴퓨터	물리
Bob	80	70	90	60
Alice	90	60		75
Carole	95	90	95	90

▶ AllScore[1][2]는 AliceScore[2]와 같음

```
1 #행렬
2 BobScore = [80, 70, 90, 60]
3 AliceScore = [90, 60, 80, 75]
4 CaroleScore = [95, 90, 95, 90]
5
6 AllScore = [BobScore, AliceScore, CaroleScore]
```

점수표						
	영어	수학	컴퓨터	물리		
Bob	[0][0]	[0][1]	[0][2]	[0][3]		
Alice	[1][0]	[1][1]	[1][2]	[1][3]		
Carole	[2][0]	[2][1]	[2][2]	[2][3]		