



Python Basic Review I: 자료구조의 이해와 활용

AI빅데이터프로그래밍

Prof.Hyunsil Moon (hsmoon@kookmin.ac.kr)

1.

리스트(List)

1 자료 구조 (Data Structure)

- 데이터를 관리하고 저장하는 방법
- Python에서 기본 제공하는 자료 구조
 - 리스트
 - 튜플
 - 사전
 - 집합

2 리스트의 정의

- Python에서 기본 제공하는 자료 구조 중 하나
 - 타 프로그래밍 언어의 배열(Array)과 유사
 - 하나의 변수에 여러 값을 할당하는 자료형 = 데이터 구조
 - 학생 100명의 성적을 채점한다면 몇 개의 변수가 필요?
- 가변적 컨테이너
- 순서가 있는 자료를 관리하기에 편리
 - 시퀀스 자료형

2 리스트의 정의

- Colors = ['red', 'blue', 'green'] 과 같은 형태로 사용

colors = ['red', 'blue', 'green']

colors →

'red'	'blue'	'green'
-------	--------	---------

- 대괄호를 사용하여 정의
- 세 개의 값을 순서대로 가진 리스트
- 하나의 자료형만이 아니라, 여러 자료형을 섞어서 사용 가능
- 리스트의 길이 : len()

- 리스트에서 특정 위치의 값에 접근하는 방법
 - `x = [1, True, 'hello']`
 - `x[0]` : `x`의 0 번째, 앞에서부터 첫번째
 - `x[1]` : `x`의 1 번째, 앞에서부터 두번째
- 0부터 시작하는 이유?
 - 이진수 관점에서 메모리 절약
 - 비주얼 베이직, 매트랩, R등은 1부터 인덱싱

- 역인덱싱
 - 리스트에서 역순으로 접근하는 방법
 - `x[-1]` : 끝에서 첫번째(0은 없다)
 - `x[-2]` : 끝에서 두번째

값	['서울', '부산', '인천', '대구', '대전', '광주', '울산', '수원']							
인덱스	-8	-7	-6	-5	-4	-3	-2	-1

- 리스트에서 특정 범위를 가리킴
 - `x[1:3]` : `x` 의 1번부터 3번 전까지 (3 번은 포함 X)
 - `x[:2]` : `x`의 시작부터 2번 전까지
 - `x[1:]` : `x` 의 1 번부터 끝까지
 - `x[:]` : `x`의 모든 값을 반환
- 슬라이싱에서 인덱스값이 넘어가더라도 작동
 - 자동으로 처음과 끝으로 맞춰서 지정
 - `x[-50:2]`
 - `x[1:50]`

슬라이싱

- 스텝(증가값)
 - 슬라이싱에서 간격을 지정할 수 있음
 - `x[1:7:2]` : 1에서 7까지 2 간격으로 역방향 스텝
 - `x[::-1]` : 뒤부터 순서대로

5 리스트의 값 바꾸기

- $y = [1, 2, 3, 4, 5]$
- $y[0] = 2$
 - x 의 0번째 값을 2로 바꾼다
- 여러 개 값 바꾸기
 - 변경할 값을 리스트로 전달
 - $y[:3] = [7, 8, 9]$
 - $y[:3] = [0, 10] \rightarrow$ 모자라게 전달한다면?
 - $y[:3] = [7, 8, 9, 10] \rightarrow$ 넘치게 전달한다면?

리스트의 연산

- 덧셈연산

- $xs + ys$: 두 리스트를 연결한 새 리스트를 만든다

- 곱셈연산

- $xs * n$: 리스트 xs 를 n 번 반복한 새 리스트를 만든다

- in 연산

- $y \text{ in } xs$: 값 y 가 리스트 xs 에 포함되어 있는지 확인한다

- not in 연산

- $y \text{ in } xs$: 값 y 가 리스트 xs 에 포함되어 있지 않은지 확인한다

패킹(Packing)과 언패킹(Unpacking)

- `a, b = [1, 2]`
 - 변수 `a` 에는 1, `b` 에는 2 가 할당된다
- 언패킹 시 할당 받는 변수의 개수가 적거나 많으면 모두 에러 발생
 - 모자를 경우에는 `_` 변수 사용
- 특별한 변수 `_`
 - Python 인터프리터는 마지막으로 반환된 결과를 변수 `_`에 자동 할당

8 리스트 주요 메서드와 함수

- 메서드 (method): 값에 소속된 형태의 함수
- 리스트 항목 추가
 - `xs.append(y)` : 리스트 `xs` 의 끝에 값 `y` 를 추가한다
 - `xs.extend(ys)` : 리스트 `xs` 에 리스트 `ys` 의 값들을 추가한다 = 덧셈연산
 - `append`와 `extend`의 차이를 잘 구분해야 함
 - `xs.insert(idx, y)` : 리스트 `xs` 의 위치 `idx` 에 값 `y` 를 추가한다
 - `idx` 위치에 새로운 값을 추가하면서 `idx` 위치를 기준으로 뒤쪽의 인덱스가 하나씩 밀림

8 리스트 주요 메서드와 함수

- 리스트 항목 삭제
 - `xs.remove(y)` : 리스트 `xs` 에서 값 `y` 를 찾아 삭제한다
 - 여러 개일 경우 가장 먼저 나오는 값 1개만 삭제
 - `del` : 리스트 또는 특정 인덱스값을 삭제
 - `xs.pop()` : 리스트의 마지막 항목을 삭제하면서 그 항목을 반환

8 리스트 주요 메서드와 함수

- 리스트 정렬하기
 - 리스트의 정렬은 비교가 가능한 경우에만 사용가능
 - 예) 숫자끼리 또는 문자끼리 구성되어 있을 경우에만 사용 가능
 - `xs.sort()` : 기본은 오름차순 정렬
 - `reverse=True` 옵션을 사용하면 내림차순 정렬
 - `sorted()`
 - `reverse=True` 옵션을 사용하면 내림차순 정렬

리스트의 연산을 활용한 문자열 처리

- 파이썬에서 문자열은 변경 불가능한(immutable) 구조를 가진 리스트
 - 내부적으로 리스트와 동일하게 취급
 - 하지만 값 변경은 불가능(immutable)
- 문자열의 길이와 인덱싱
 - `len('hello')` : `len`함수를 이용해 문자열의 길이 파악 가능
 - 리스트와 같은 방식으로 인덱싱과 슬라이싱

리스트의 연산을 활용한 문자열 처리

- 'hello' + ' world'
 - 두 문자열을 더해 'helloworld' 가 된다
- 'hello' * 2
 - 'hello'를 2회 반복하여 'hellohello'가 된다
- 'e' in ' hello'
 - 'e'가 'hello'에 포함되어 있는지 확인한다
- 'e' not in 'world'
 - 'e' 가 'world' 에 포함 안되어 있는지 확인한다

2.

기타 자료 구조

튜플 (Tuple)

- 여러 값을 소괄호 ()로 묶어서 표현 : (1, True, 'hello')
- 리스트와 동일하나 변경 불가능(immutable)
 - 추가, 삭제, 수정 X

2 사전(Dictionary)

- 열쇠(key)와 값(value)이 짝을 이룬 형태의 자료 구조
 - 중괄호 { }를 사용하여 표현

```
d = {'apple': 1, 'banana': 2}
d['apple']
```
 - apple이 열쇠, 1이 그 값
 - 순서보다 열쇠가 중요한 경우(예: 주소록) 사용

2 사전(Dictionary)

- 사전에 없는 열쇠를 사용하면 `KeyError`가 발생
 - `.get` 메서드를 사용하면 없는 열쇠의 경우 `None`값을 반환
 - `.get` 메서드에 값을 주면 없는 열쇠에는 기본값을 반환
- 특정 열쇠의 삭제
 - `del d['apple']`
- 특정 열쇠의 추가
 - `d['apple'] = 3`

집합(Set)

- 수학의 집합과 비슷한 자료 구조
 - 사용 예제 : {1, 2, 3}
 - 사전과 표기법이 비슷하나 키와 값의 짝이 없음
 - 리스트나 튜플과 비슷하나 중복을 허용하지 않고 순서를 보존하지 않음
 - 중복 제거에 활용

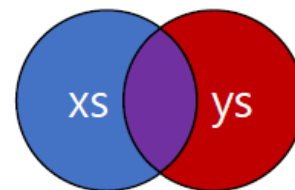
집합(Set)

- 원소 추가 : `.add()`
- 원소 삭제
 - `.remove()` : 값이 없으면 에러 발생
 - `.discard()` : 값이 없어도 에러는 발생되지 않음
 - `.clear()` : 모든 항목 삭제

집합(Set)

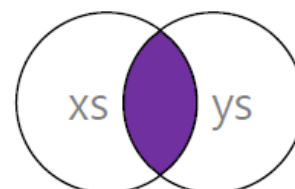
- $xs \mid ys$

합집합



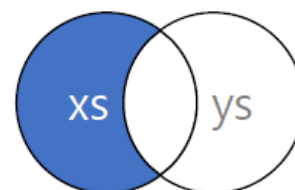
- $xs \& ys$

교집합



- $xs - ys$

차집합



- $xs < ys$

집합 xs 가 집합 ys 의 부분집합인지 확인

- $x \text{ in } xs$

원소 x 가 집합 xs 에 포함되는지 확인

4 자료 구조의 변환

- 함수를 이용해서 자료 구조의 변환 가능
 - `list()` : 리스트
 - `tuple()` : 튜플
 - `dict()` : 사전
 - `set()` : 집합
- 단, 사전은 다른 자료 구조로 바꿀 때 열쇠(key)만 보존
 - 값(value)을 보존하고 싶은 경우에는 `변수명.values()` 메서드 사용

A decorative border at the top of the slide features a repeating pattern of white line-art icons on a blue background. The icons include a tag, puzzle piece, magnifying glass, smartphone, document, envelope, speech bubble, target, gear, pie chart, lightbulb, clock, checkmark, and presentation board.

Q&A

Thank you for your listening