

예약어

class for while if else except elif 등

연산자

- a//b
- a/b
- a%b
- a**b
- a-=3

함수 정리

[문자열]

a.upper()

a.replace('a','b') : a를 b로 바꿔라

format 함수

split('a') : 특정한 문자열을 구분하여 문자열의 리스트로 치환

''.join(a) : a 리스트 안 문자열에 ' '를 삽입하여 새로운 문자열로 합치기

예) ''.join(a) -> ['a','b'] -> ['ab']

예) '-'.join(a) -> ['a','b'] -> ['a-b']

[리스트] 멤버 함수

a.append(1) : 리스트 끝에 항목 추가

a.split('w') : 'w'를 기준으로 문자열 나눠서 리스트화

a.upper() : 대문자로 변환

a.extend(b) : 리스트 a와 리스트 b를 합침 (a += b, a+b 가능

a.insert(1,40) : a[1]에 40이란 요소를 추가

a.remove() : 값으로 항목 삭제

a.pop(2) : a[2]반환 + 삭제

index(1) : 1의 인덱스(위치)를 반환

in 키워드 : value in [list]. 리스트 내에 키워드가 존재하는지 확인 / true false 형으로 반환

sort() : list 정렬 *a.sort(reverse=True) : 내림차순으로 정렬

sorted() : list의 정렬된 복사본을 반환

[dict] 관련 함수

- a.update(b) : a에 딕셔너리 병합 + 겹치는 key의 value는 over-write (update) 됨
- key 삭제
 - 1. del a['key']
 - 2. a.pop('key')
- a.clear() : a의 모든 요소 삭제
- 'b' in a : 'b'라는 key가 a에 있느냐 dict에서 in 키워드는 딕셔너리 크기와 관계없이 항상 연산의 속도가 일정(매우 빠름. key만 바로 검색)list와 같이 인덱스가 있는 경우, 'in' 키워드 검색 시, 연산 매우 많이 해야함. 성능의 저하 초래가능
- value access
 - 1. a['key']
 - 2. a.get['key']
- 모든 keys, values 접근
 - 1. a.keys() * list(a.keys())로, 리스트형으로 변환해 사용 많이함.
 - 2. a.values() * list(a.values())로, 리스트형으로 변환해 사용 많이함.
 - 3. a.items() : key:value를 튜플로 반환

[set] 메소드

a.union(b) : 합집합

a.intersection(b) : 교집합

a.difference(b) : 차집합

a.issubset(b) : 부분집합

1. 문자열은 불변객체 (리스트는 변동가능객체이지만, 문자열만큼은 예외)

리스트는 변동 가능 객체이나

문자열은 불변객체

문자열은 리스트로 저장이 되어서, 바뀌네기가 가능할 것 같으나, 불가

바꾸고자 하면,

- 1. 바뀌서 전체를 새로운 문자열로 저장
- 2. replace 함수 이용. b = a.replace('h','j') 로 새로 b에 저장

문자열 인덱싱 - increment

a[0:4:2] 하면, a[0], [2] 가 출력됨 (0<= a <4, 2씩 건너뛴)

2. 튜플 : 왜 쓰나?

- 데이터 컬렉션으로서의 기능 X. 튜플의 값을 차례대로 변수에 대입 (tuple unpacking. 변수에 숫자들을 나온 순서대로 차례차례 대입함)

3. set

- 중복 허용 X + 순서가 없음(인덱싱 X. dict형과 같은 성질)
- 집합과 같은 개념. dict의 key만 모아놓은 것. 인덱싱 X.

- 교집합, 차집합, 합집합 구하기 좋음

잠깐 체크

[] : 리스트

() : 튜플

{ } : set

{ ' ': ' ', ' ': ' ', ... } : dict

4. dict

- { 'key' : 'value', ... } 구조
- 순서가 없음 (인덱싱 X)
- key가 해시형으로 저장됨 (key로 호출, value가 꺼려나온다. 굉장히 빠르다. key만 해쉬구조로 저장)

a['key'] = 'value' 로 추가 a = {'key':'value', 등으로 추가}

In []: