DAS 사전교육

- Python 기초 -

기본 개념2

- 1. 컬렉션 자료형
- 2. 리스트 내포
- 3. 정규 표현식

1. 컬렉션 자료형

- 컬렉션: 여러 개의 값을 하나의 변수에 담을 수 있고 변수 안에 공간을 여러 개 가지며, 변수 안에 서로 다른 공간을 찾는 방법

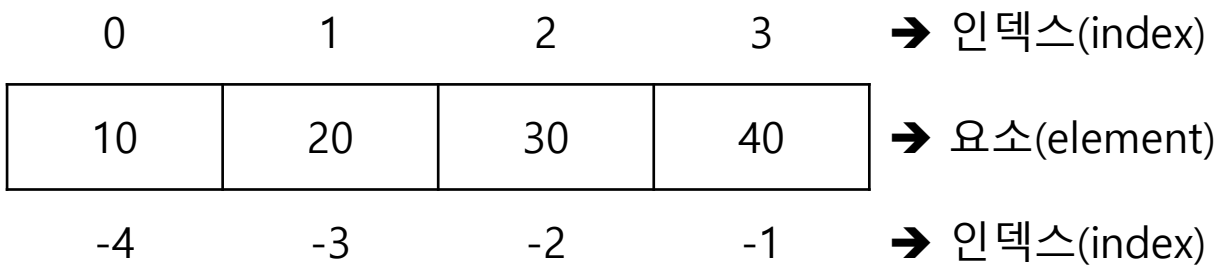
컬렉션 자료형 종류	생성 방법
리스트 List	[ ]
튜플 Tuple	( )
딕셔너리 Dictionary	{key : value}
세트 Set	{ }

1.1 리스트 자료형

- 대괄호 [ ] 안에 서로 다른 자료형의 값을 콤마(,)로 구분
- 대괄호 [ ]에 넣는 자료를 요소(element)
- 요소들은 순서를 가지고 있고 인덱스를 사용하여 참조 가능

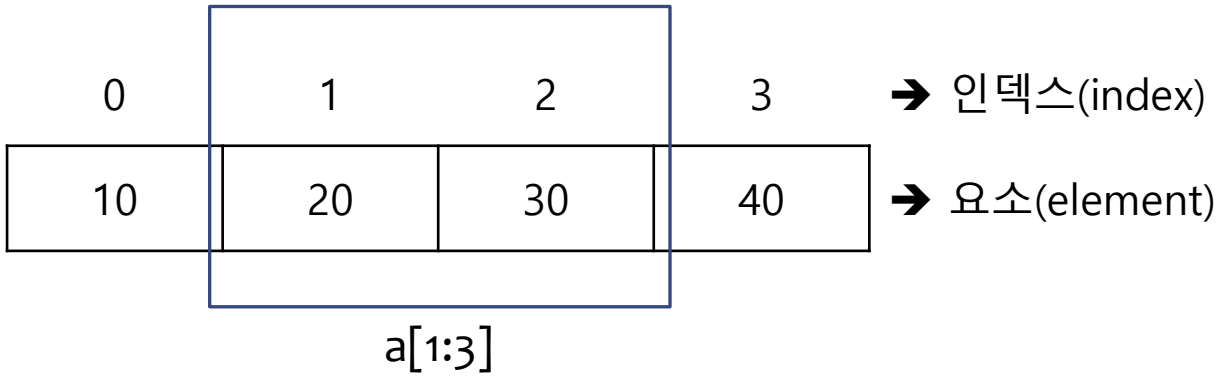
1.1 리스트 자료형

- 리스트 인덱싱(Indexing)
  - 0에서 시작
  - 마지막부터 접근할 때는 -1부터



1.1 리스트 자료형

- 리스트 슬라이싱(Indexing)
  - 리스트명[시작 : 끝+1]



1.1 리스트 자료형

- 리스트 값 변경하기
  - 인덱스를 사용해서 그 위치의 값을 변경
- 리스트 제어 함수

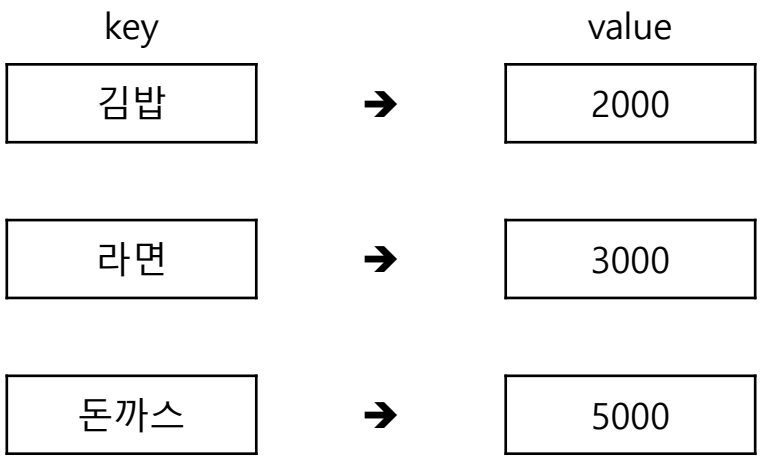
함수	설명	사용법
append( )	리스트에 요소를 마지막 위치에 새로 추가	리스트.append(값)
insert( )	리스트의 해당 위치에 요소를 새로 삽입	리스트.insert(위치, 값)
sort( )	오름차순 정렬 내림차순 정렬	리스트.sort() 리스트.sort(reverse=True)
reverse( )	현재의 리스트를 그대로 거꾸로 뒤집기	리스트.reverse()
pop( )	리스트 제일 뒤의 항목을 빼내고, 빼낸 항목은 삭제 제거할 위치에 있는 요소를 제거	리스트.pop() 리스트.pop(위치)
remove( )	해당 요소를 찾아 삭제	리스트.remove(삭제할값)
count( )	해당 요소의 개수를 반환	리스트.count(찾을값)
index( )	리스트에 위치 값이 있으면 위치값을 반환	리스트.index(값)
len( )	리스트 총 요소 개수를 반환	len(리스트)

1.2 튜플 자료형

- 리스트는 도중 바뀔 수 있는 요소 저장에 알맞은 자료형
- 튜플은 한 번 저장된 값은 수정할 수 없는 자료형
- 괄호 ( ) 안에 서로 다른 자료형의 값을 콤마(,)로 구분
- 인덱싱과 슬라이싱은 리스트와 동일

1.3 딕셔너리 자료형

- 순서가 없는 컬렉션 자료형
- 각각의 요소는 key : value 형태로 저장
- index로 요소를 찾지 않고 key를 통해 value를 얻는다



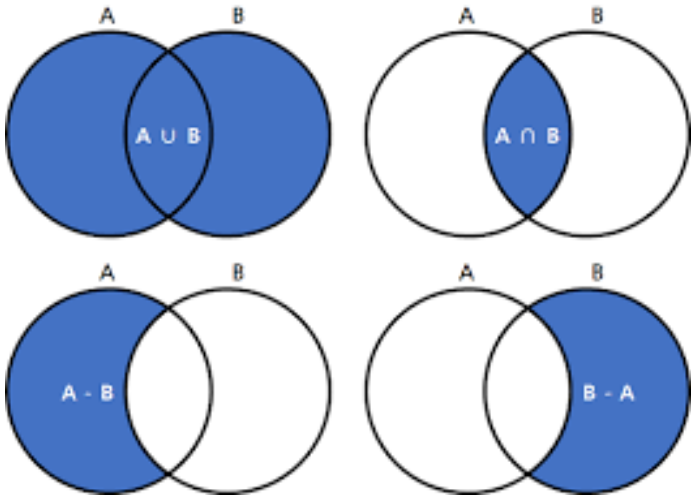
1.3 딕셔너리 자료형

- 딕셔너리 제어 함수

함수	설명	사용법
get( )	항목 접근하기	dict.get(key)
del( ) pop( )	항목 삭제하기 항목 꺼내고 삭제하기	del(dict[key]) dict.pop(key)
items( )	딕셔너리에 저장된 항목	dict.items()
keys( )	딕셔너리에 저장된 키	dict.keys()
values( )	딕셔너리에 저장된 값	dict.values()

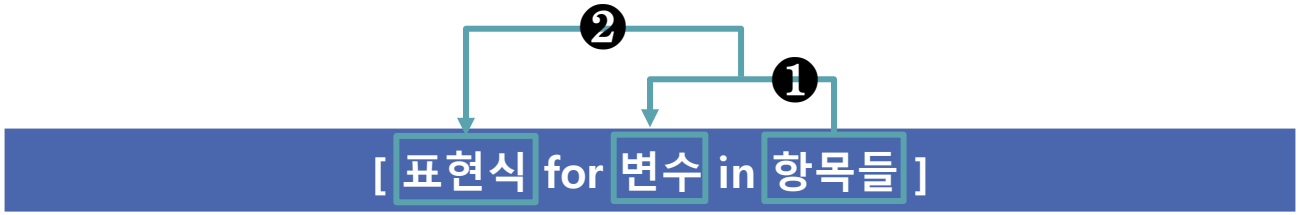
1.4 세트 자료형

- 중복을 허용하지 않는 컬렉션 자료형
- 순서가 없기 때문에 인덱싱 불가
- 수학의 집합과 같다.



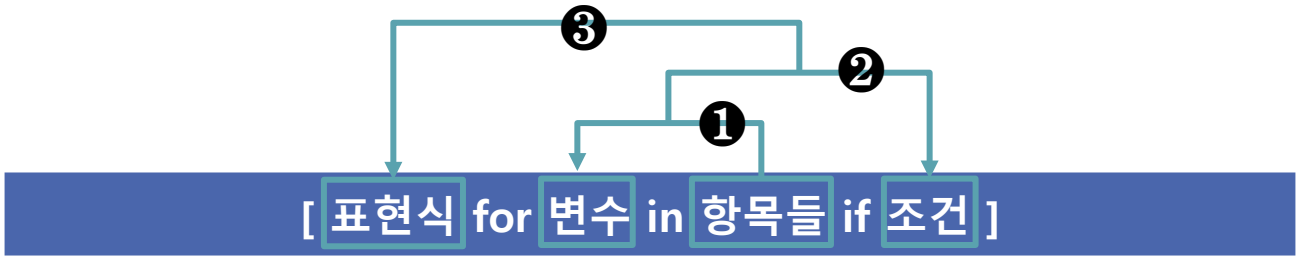
2. 리스트 내포

- 리스트 내포(List Comprehension): 원하는 자료들을 조회 또는 추출하여 리스트로 반환하는 표현식
  - 1. 항목들에서 순차적으로 하나씩 꺼내온다.
  - 2. 조건식을 적용하여 해당 조건에 맞는 항목은 추출하고, 조건에 맞지 않으면 무시한다.
  - 3. 위에서 추출된 항목은 리스트에 추가한다.



2. 리스트 내포

- 조건 있는 리스트 내포 형식



3. 정규 표현식

- 정규 표현식(Regular Expression): 패턴 매칭 기반으로 특정한 규칙을 가지는 문자열을 검색하고 분리하고 교체하는 강력한 기능을 제공하는 형식 언어

- 문자열 패턴 매칭

문자열 처음부터 매칭 문법	문자열 일부분 매칭 문법
re.match('패턴', '문자열')	re.search('패턴', '문자열')

3. 정규 표현식

대표 메타문자

메타문자	설명	사용법
^	시작 패턴 표현	^abc : abc로 시작하는 패턴
$	종료 패턴 표현	xyz$: xyz로 종료하는 패턴
[문자들]	문자들 중에 하나만 허용, 원하는 문자들의 집합 표현	[Ww]orld : 'World' 또는 'world'
[^문자들]	[문자들]을 제외한 문자들의 집합 표현	[^aeiou] : 소문자 모음이 아닌 문자들
|	두 패턴 중에 하나만 허용 (OR)	a | b : a 또는 b
?	앞 패턴이 없거나 하나만 허용	a? : a가 없거나 하나만
+	앞 패턴이 하나 이상 존재하는 표현	a+ : a가 하나 이상
*	앞 패턴이 0개 이상 존재하는 표현	a* : a가 없거나 하나 이상
패턴{n}	앞 패턴이 n번 반복하는 표현	a{2} : a가 연속 2번 나타나는 패턴
패턴{n,m}	앞 패턴이 최소 n번, 최대 m번 반복해서 나타나는 경우(n or m은 생략 가능)	a{3, 5} : a가 3번, 4번, 5번 나타나는 패턴
₩d	숫자 0~9	₩d₩d₩d : 0~9 범위의 숫자 3개 의미
₩w	문자	₩w₩w₩w : 3개 문자 의미
₩s	화이트 스페이스, [₩t₩n₩r₩f]와 동일	₩s₩s : 화이트 스페이스 문자 2개의 의미
.	줄바꿈(₩n)을 제외한 모든 문자	.{3} : 문자 3개
[0-9]	0~9 사이의 모든 숫자	[0-9]+ : ex) 0, 123, 14235
[A-Z]	A~Z 사이의 모든 대문자	[A-Z]+ : ex) ABC, AEEEE
[a-z]	a~z 사이의 모든 소문자	[a-z]+ : ex)abc, abbc

3. 정규 표현식

- 특수문자 판단

특수문자	설명
₩d	[0-9]와 같음. 모든 숫자
₩D	[^0-9]와 같음. 숫자를 제외한 모든 문자
₩w	[a-zA-Z0-9_]와 같음. 영문 대소문자, 숫자, 밑줄 문자
₩W	[^a-zA-Z0-9_]와 같음. 영문 대소문자, 숫자, 밑줄 문자를 제외한 모든 문자