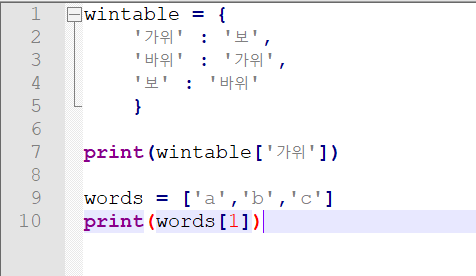
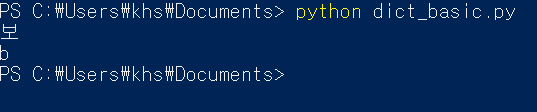
6월 넷째 주 보고서

python dictionary, tuple

고희수

<python dictionary>



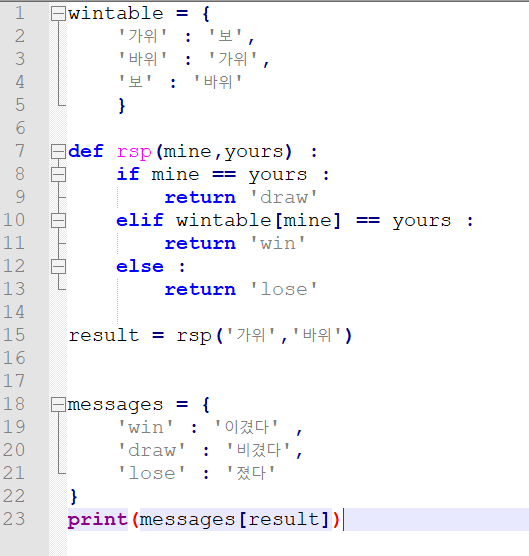
* 딕셔너리

1. 여러 값을 저장해 두고 필요한 값을 꺼내 쓰는 기능
2. 이름표를 이용하여 값을 꺼내 사용
3. 사용할 때는 리스트와 비슷한 방식

Ex\_) wintable > 각 가위바위보 패의 승리조건을 가지고 있음.

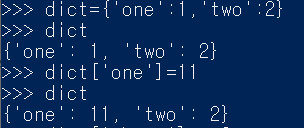
Print(wintable[‘가위’]) : ‘가위’라는 이름표를 가진 값을 가져오라는 의미

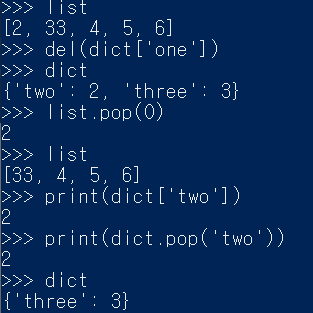
: 이름표와 값은 : 으로 구분, 여러 개의 이름표와 값 세트는 콤마로 구분



* Messages 라는 딕셔너리를 만들어서 사용자가 알기 더 쉽게 출력

Dict 수정과 삭제

 >> dict의 값 수정

 >> dict, list 의 값 삭제 방법 비교

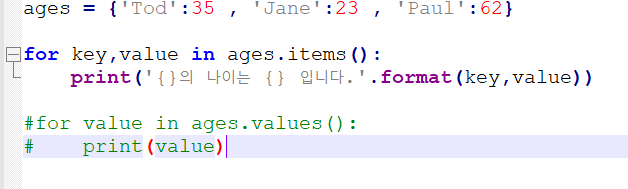
<Dict의 반복>

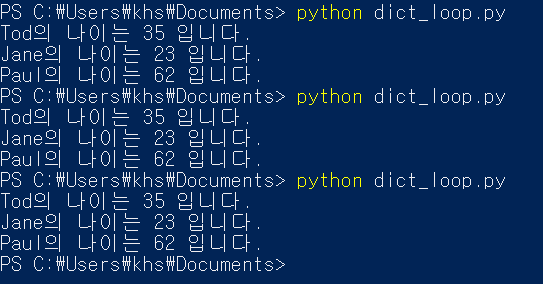
Dict 반복에는 여러가지 방법이 있음

1. key 를 가져오는 법 >> dict의 이름표를 가져오는 것

2. value를 가져오는 법 >> dict의 값을 가져오는 것

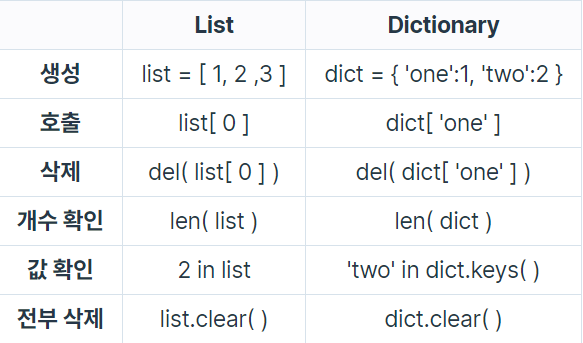
3. 둘 다 가져오는 법 >> items 활용 ( for in 구문에서 keys()는 생략 가능함)





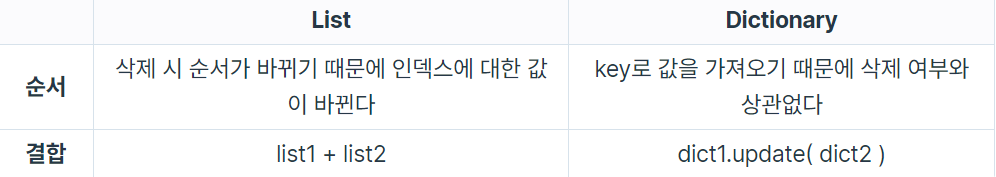
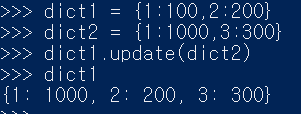
>> items 를 ages[key]로 써도 실행 가능

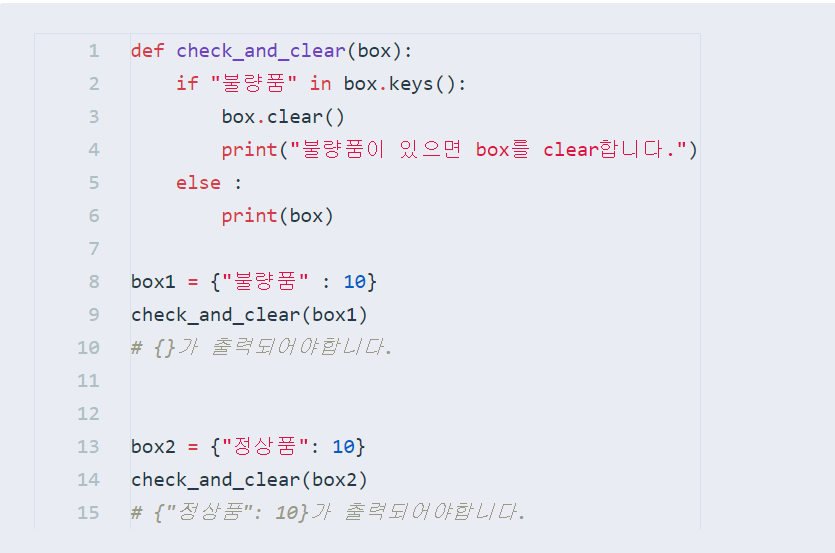
\*\* List와 Dict 의 공통점, 차이점



! 2 in dict.keys or 2 in dict.values 는 dict안 이름표 혹은 값이 있는 지 확인 해 줌

! dict.clear() 은 dict 안을 모두 비워줌

 >> dict.update 예시

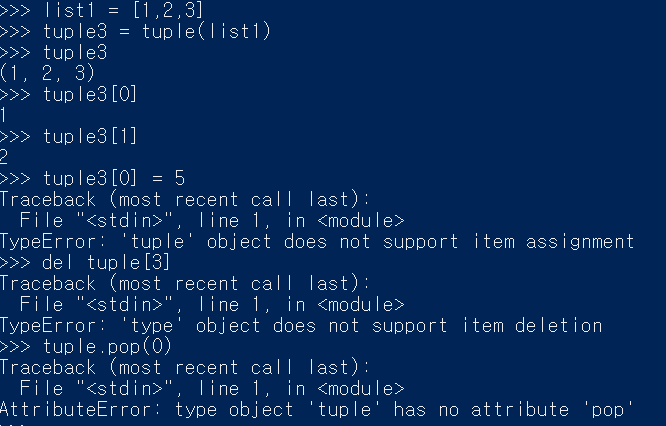


>> “불량품”이 있으면 모두 비우고 없다면 그대로 출력하기 위한 코드

< tuple 생성 >

순서가 한번 정해지면 바꿀 수 없음

>> 순서가 정해져 있기 때문에 인덱스 값 수정 불가



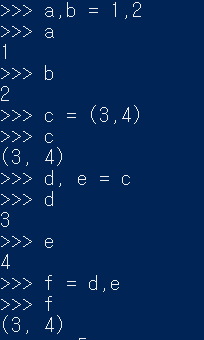
>> list를 tuple 에 넣어서 사용 할 수 있음

>> list 처럼 del, pop 함수는 사용 불가

>> tuple = (1,2,3) , tuple= 1,2,3 모두 사용 가능

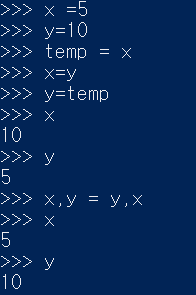
<tuple을 이용한 packing, unpacking>

1. Packing : 하나의 변수에 여러개의 값을 넣는 것
2. Unpacking : 패킹된 변수에서 여러개의 값을 꺼내오는 것

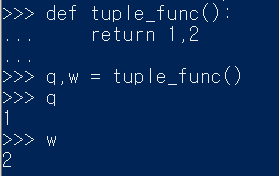


>> d, e = c >> unpacking

>> f = d, e >> packing

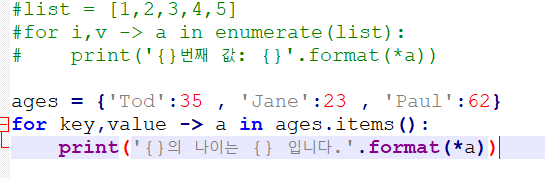


>> temp( 임시 변수 ) 없이도 값 교환 가능



>> 함수의 리턴 값으로 여러 값을 전달할 수 있다.

<tuple을 이용한 함수의 리턴값>



>>(i, v) , (key, value)를 a로 바꿔도 결과는 같음

>>format 뒤 \*a 는 for in 구문 확인