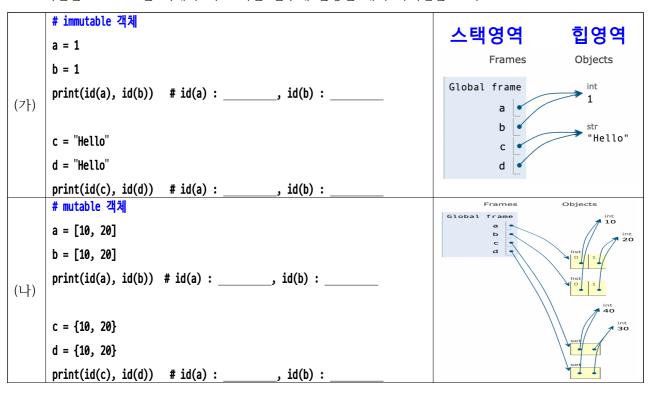
파이썬의 (immutable/mutable)객체에 대한 요약 보고서

- * "https://pythontutor.com/visualize.html#mode=edit" : 코드 진행상황 및 메모리 상태를 시각화
- * id()는 객체가 메모리(주기억장치)의 어느 위치(주소)에 저장되는지를 뜻함.
- * id(a)와 id(b)가 서로 같다면 a와 b는 서로 같은 객체를 참조하고 있다고 할 수 있다.
- 1. 아래 질문에 응답하시오.
 - 가. 아래 코드의 실행 결과 출력되는 값을 쓰시오.
 - 나. 왼쪽 코드가 실행 후 메모리 상태를 도식화 한 그림을 보고, 똑같은 immutable한 객체가 서로 다른 변수에 할당될 때와, 똑같은 mutable한 객체가 서로 다른 변수에 할당될 때의 차이점을 쓰시오.



2. immutable한 객체와 mutable한 객체의 값 변경 전과 변경 후의 id()값을 비교하시오.

```
# immutable 객체 변경
a = 10
print(a, id(a)) # a : _____, id(a) : _____

a = a + 1 # 변경
print(a, id(a)) # a : _____, id(a) : _____

immutable한 객체를 변경했을 때, id(a)가 변경되었나?( )

# mutable 객체 변경
L = [10]
print(L, id(L)) # L : [10] ___, id(L) : _____

(나) L.append(20) # 변경
print(L, id(L)) # L : _____, id(L) : _____

mutable한 객체를 변경했을 때, id(a)가 변경되었나?( )
```

3. mutable한 리스트 객체를 아래와 같이, 변수 a와 b에 할당할 때, 차이점은? wutable한 객체가 얕은 봤거나 되었을 때, 발생될 수 있는 문제점에 대해 이해하다.

```
# mutable 객체
   a = [10, 20]
   b = [10, 20]
   (가)
   a[0] = 100
   print(a, id(a)) # a : [100, 20], id(a) : ______
print(b, id(b)) # b : [_____], id(b) : ______
   # a와 b는 같은 객체를 참조하는가? 근거는?
   # mutable 객체
   a = [10, 20]
            # 얕은 복사
   b = a
   (나)
   a[0] = 100
   # a와 b는 같은 객체를 참조하는가? 근거는?
```

4. 2차원 리스트를 생성하는 (가), (나) 코드의 실행 후 메모리 상태의 차이점을 그림으로 그려보시오.

(가) 2차원 리스트 만들기1 a = [[0] * 3] * 2	# (나) 2차원 리스트 만들기2 a = [[0 for j in range(3)] for i in range(2)]

5. immutable/mutable한 전달인자를 함수안(local)에서 변경시켰을 때, Global영역의 값도 변경되는지 비교해 보자.

(가) immutable한 전달인자 받기	(나) mutable한 전달인자 받기
<pre>def Immutable_Func(i) :</pre>	<pre>def Mutable_Func(L) :</pre>
i = i + 1	L.append(20)
print(i) # i :	print(L) # L :
i = 1 # immutable한 정수 객체 Immutable_Func(i)	L = [10] # mutable한 리스트 객체 Mutable_Func(L)
print(i) # i :	print(L) # L :