

파이썬과 메모리

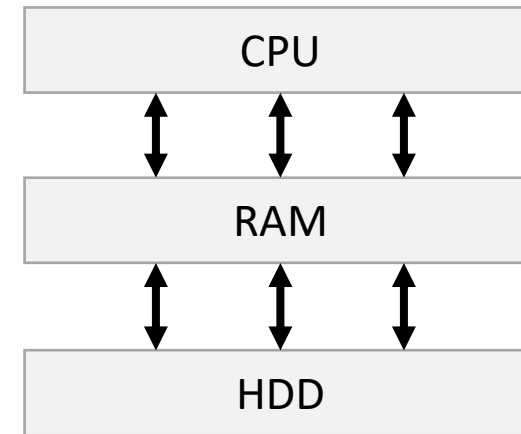
Python을 활용한 자료구조 이해하기

메모리란? (1/2)

- RAM(Random Access Memory)
 - CPU가 주로 사용하는 데이터 저장 장치
 - 메인 메모리는 전원이 차단되면 데이터는 제거



삼성전자 PC133 SDR SDRAM



메모리란? (2/2)

- 메모리의 이해를 돕기 위한 비유

- ✓ 창고

- 많은 양의 빵을 만들어 저장하는 공간

- ✓ 진열대

- 고객에게 빵을 보여주는 목적
- 예쁘게 빵을 진열하기 위한 공간

- ✓ 하드 디스크 (HDD)

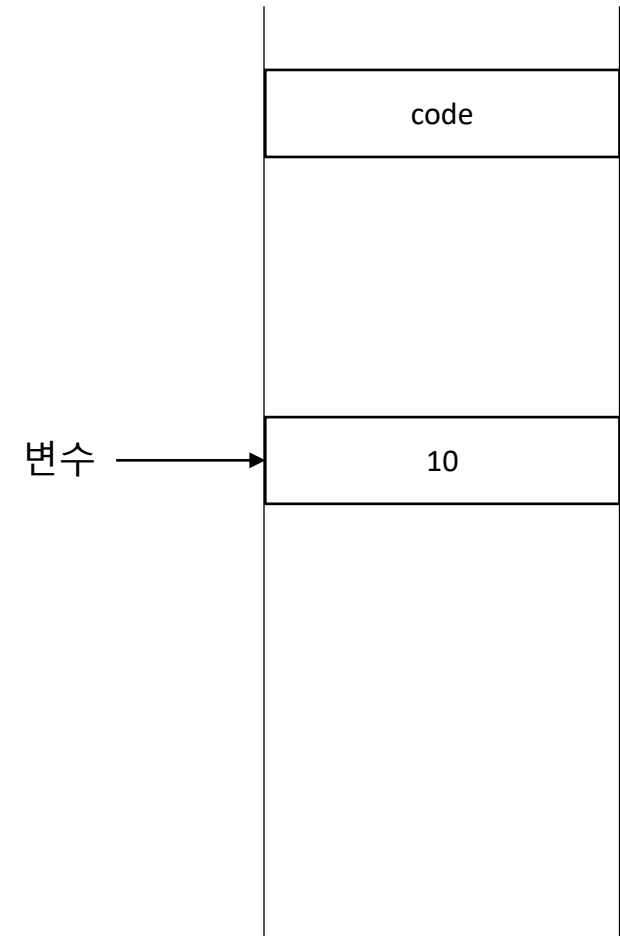
- 전원이 꺼져도 데이터를 유지
- 방대한 양의 데이터를 저장

- ✓ 메모리

- 비교적 적은 양의 데이터를 저장
- CPU가 빠르게 데이터를 가져갈 수 있는 장치

파이썬과 메모리

- 모든 데이터는 메모리에 로드 됨
 - 파이썬 인터프리터는 메모리에 데이터를 읽고 씀
 - 프로그램 실행 시 하드디스크에 저장된 코드도 메모리에 로드
 - 파이썬에서 사용하는 데이터는 메모리에 저장



데이터와 메모리

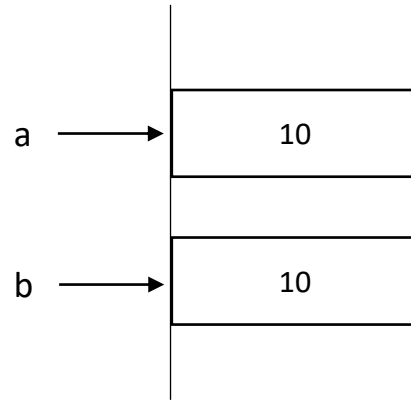
- 변수 a와 b가 저장된 메모리의 상태로 옳은 것은?

```
a = 10
```

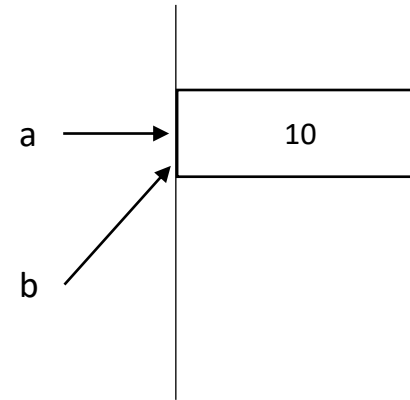
```
b = 10
```

```
print( id(a), id(b) )
```

※ id 함수는 메모리의 주소를 출력



(a)



(b)

연습 문제

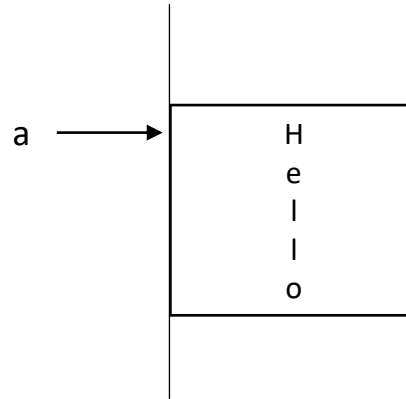
- 다음 코드를 실행했을 때의 메모리의 상태를 그리시오.

```
a = 10  
b = 10  
b = a + b + 2
```

문자열과 메모리

- 문자열이 저장된 메모리를 변수 a가 바인딩

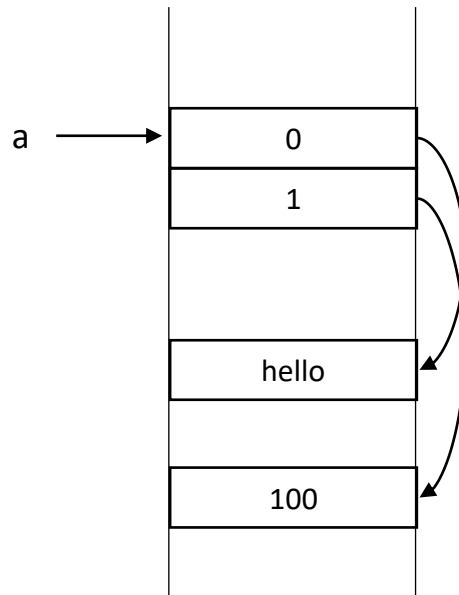
```
a = "Hello"
```



리스트와 메모리

- 다양한 데이터 타입을 저장
 - 가변 데이터를 효율적으로 저장하기 위한 구조

```
a = [ "hello", 100 ]
```

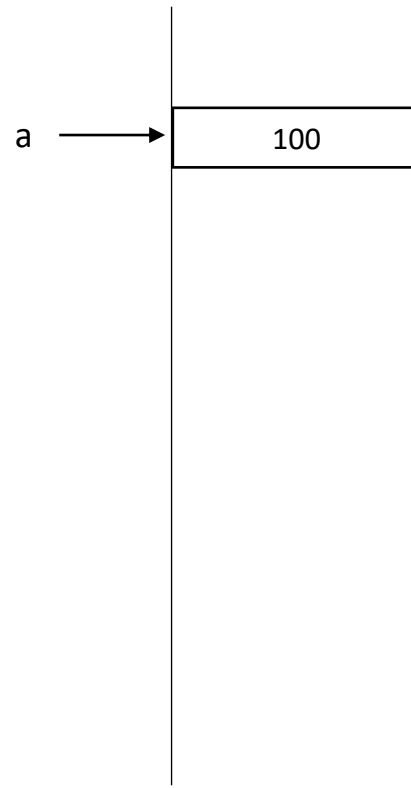


함수와 메모리 (1/3)

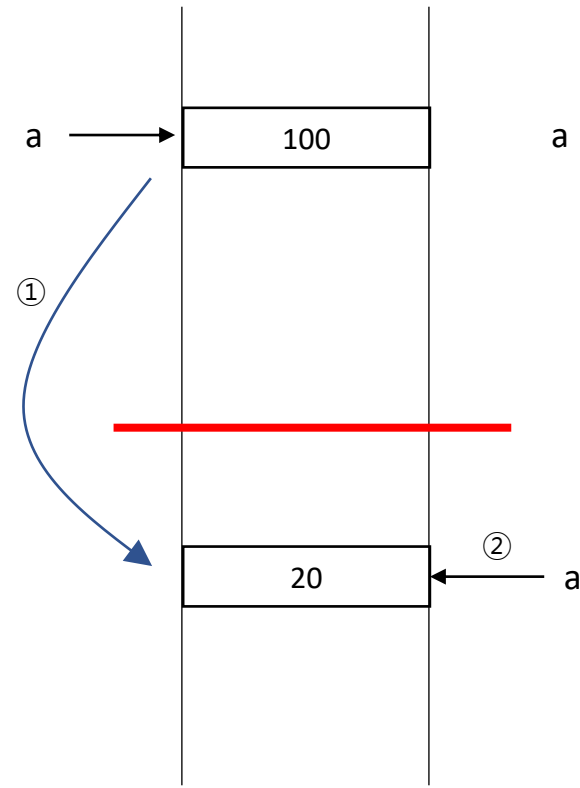
- 함수는 기본적으로 독립된 메모리 영역에서 실행
 - 실행 후 미사용 변수 제거

```
def func( ):
    a = 20
```

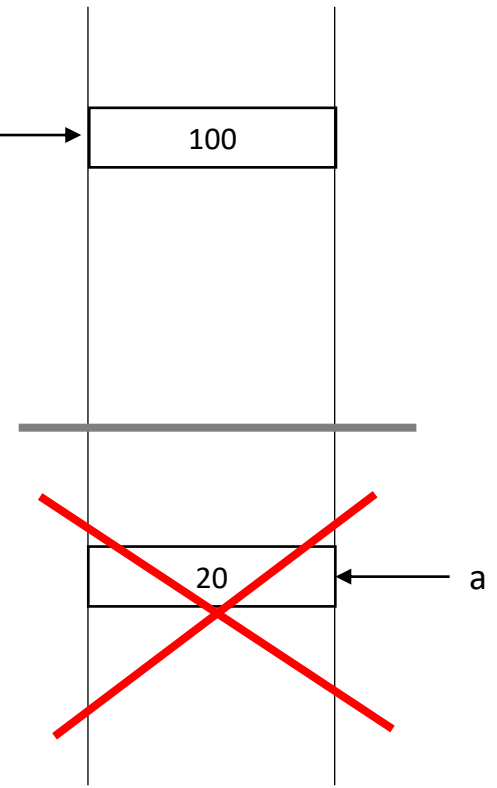
```
a = 100
func()
print(a)
```



함수 호출 전



함수 내부



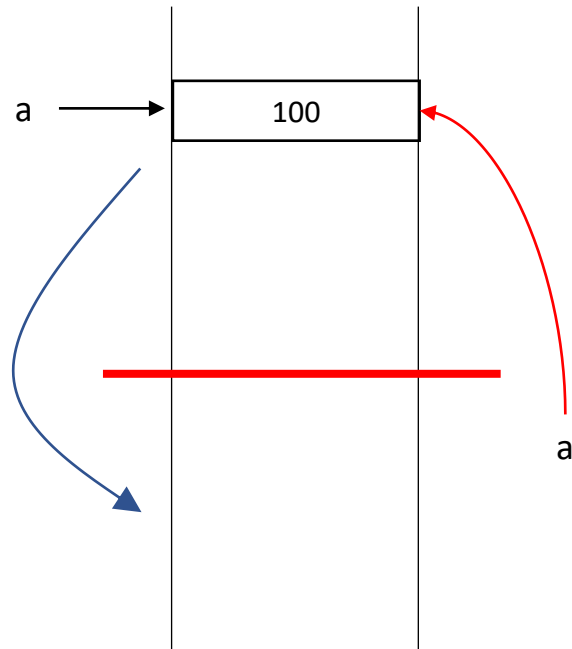
함수 호출 후

함수와 메모리 (2/3)

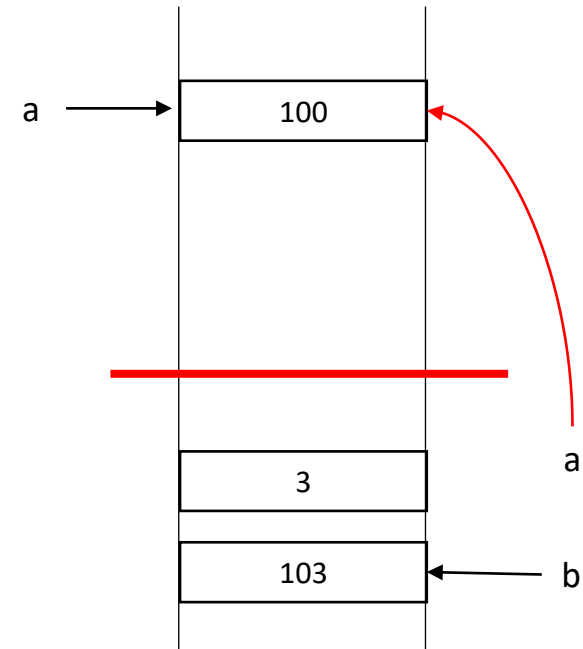
- 입력이 있는 함수와 메모리
 - 함수 호출 시 지역 변수가 전달 한 값을 바인딩

```
def func( a ) : # 1)  
    b = a + 3    # 2)  
    print( b )
```

```
a = 100  
func( a )
```



함수 내부 #1



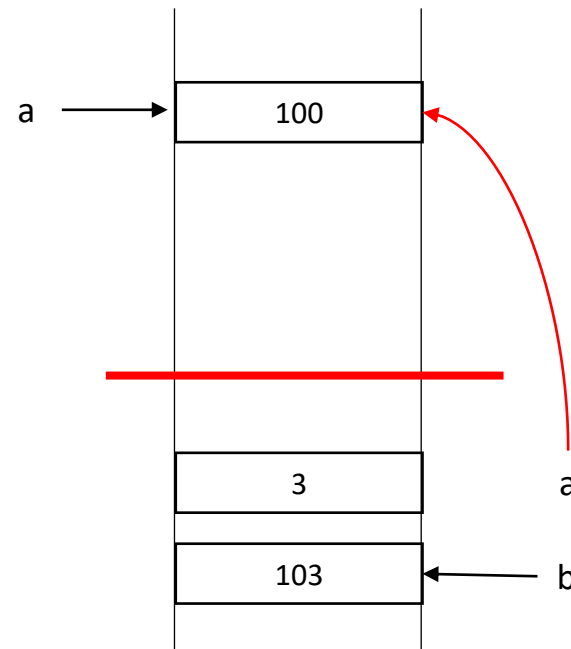
함수 내부 #2

함수와 메모리 (3/3)

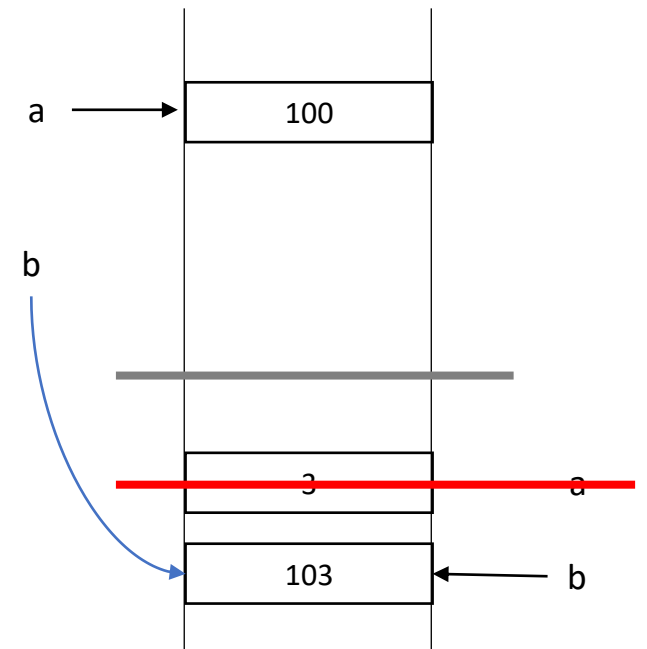
- 반환 값이 있는 함수와 메모리
 - return은 함수 외부에서 데이터를 사용한다고 알림
 - 따라서 메모리의 데이터를 제거하지 않음

```
def func( a ) : # 1)  
    b = a + 3    # 2)  
    return b
```

```
a = 100  
b = func( a )  
print( b )
```



함수 내부 #2



함수 호출 후