

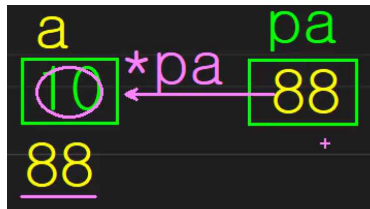
go는 포인터를 쓸 수 있는 몇 안되는 언어중 하나이다.(자바,썬은 지원x)

포인터는 잘 사용하면 굉장히 좋지만 잘못 사용하면 안쓰는 것만도 못하다.(메모리 문제유발)

포인터란 : 메모리 주소를 값으로 가지는 타입. &주소연산자, *포인터 기호

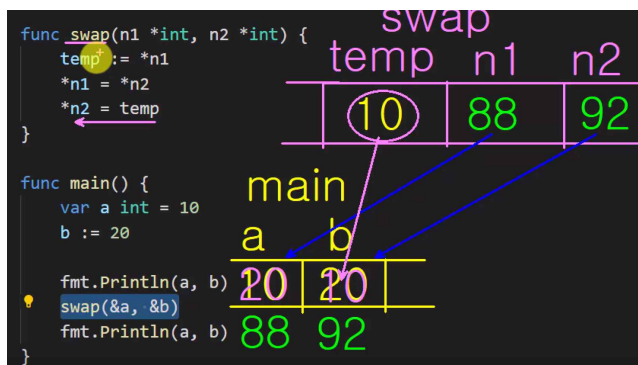
Println(&a ,a)를 하면 앞에는 a의 주소 뒤에는 a의 값이 Print된다.

포인터 변수 pa에 a의 주소를 지정했을 때 Println(pa, *pa) 하면 위에 것과 값이 같음.



pa는 포인터변수 타입 *int a는 값변수 int타입

함수에는 call by value와 call by reference가 있는데 다음은 call by reference의 예이다.



a와 b의 주소를 함수 swap에서 받아

a,b의 주소를 이용해서 a와b의 값을 바꿔준다 이것을 call by value로 호출하면

```
func swap(n1 int, n2 int) {
    temp := n1
    n1 = n2
    n2 = temp
    fmt.Printf("함수속 a = %d, b = %d \n", n1, n2)
}

func main() {
    a := 10
    b := 20

    fmt.Println(a, b)
    swap(a, b)
    fmt.Println(a, b)
}
```

이런식으로 호출을 하게 되고 결과는

```
PS C:\Users\spa89\gowork\syntax01> go run .\syntax01.go
10 20
함수속 a = 20, b = 10
10 20
```

a와 b의 값이 함수의 지역변수로 들어가서 함수 속에서만(n1,n2로) 값이 바뀌고 함수가 끝나고 나와서는 다시 원래 a와b의 값이 10 20 그대로인 것을 확인할 수 있다. 이것이 call by value 와 call by reference의 차이이다