

```
#=====
```

```
# 함수 만들기
```

```
#=====
```

```
# 두 입력 항목을 더해 2 를 곱하는 함수 생성
```

```
my_fun <- function(i, j){
```

```
    k <- (i+j)*2
```

```
    return(k)
```

```
}
```

```
# 생성된 객체의 형식 확인
```

```
class(my_fun)
```

```
# 함수 실행: (2+3)*2 = 10
```

```
my_fun(2, 3)
```

```
#-----
```

```
# 두 입력 항목을 더해 2 를 곱하는 함수 – 객체에 값을 지정해 return 을 수행하지 않고 바로 작성
```

```
# return 구문이 없을 경우 마지막 구문의 결과를 반환
```

```
my_fun <- function(i, j){ (i+j)*2 }
```

```
# 함수 실행
```

```
my_fun(2, 3)
```

```
# 입력 항목
```

```
input <- c(1, 2, 3)
```

```
# 함수 생성
```

```
my_fun2 <- function(obj){
```

```
    obj <- obj*2
```

```
    input <- obj # 함수 안에서 계산된 값을 함수 바깥에 있는 input 객체에 지정
```

```
    return(obj)
```

```
}
```

```
# 함수 실행
```

```
my_fun2(input)
```

```
# 함수 안에서 input 변수를 변경해도 input 객체는 변경되지 않음
```

```
input
```

```
# 함수 실행 시 결과를 <-로 지정
```

```
input <- my_fun2(input)
```

```
# 함수 반환 결과 확인
```

```
input
```

```
#-----
```

```
# 입력 항목
```

```
input <- c(1,2,3)
```

# 함수 생성 - 입력받은 객체에 2 를 곱한 후 반환

```
my_fun2 <- function(obj){  
    obj <- obj*2 # 입력받은 객체에 2 를 곱한 후 자신에게 재지정  
    input <-<- obj # "<<-" 연산자로 함수 바깥의 객체에 값 지정  
    return(obj)  
}
```

# 함수 실행

```
my_fun2(input)
```

# 입력 항목 - 함수의 결과를 반환받지 않았으나 input 객체가 변경됨

```
input
```

#-----

# 함수 생성 - 80 이상이면 합격 출력

```
my_fun3 <- function(score){  
    if(score >= 80){  
        "합격"  
    }else{  
        "불합격"  
    }  
}
```

# 70 점일 때

```
my_fun3(70)
```

```
# 90 점일 때
```

```
my_fun3(90)
```

```
#-----
```

```
# ifelse
```

```
my_fun4 <- function(score){
```

```
  ifelse(score >= 80, "합격", "불합격")
```

```
}
```

```
# 70 점일 때
```

```
my_fun4(70)
```

```
# 90 점일 때
```

```
my_fun4(90)
```

```
#-----
```

```
# 함수 생성
```

```
my_fun5 <- function(score){
```

```
  if(score >= 80){
```

```
    "합격"
```

```
  } else if(score >= 75){ # 80 점 이상은 if 절에서 처리됨(75 ~ 80 점인 경우 실행)
```

```
    "대기"
```

```
  } else{
```

```
    "불합격"
```

```
    }  
}
```

```
# 70 점일 때  
my_fun5(70)
```

```
# 78 점일 때  
my_fun5(78)
```

```
# 90 점일 때  
my_fun5(90)
```

```
#-----
```

```
# 함수 생성 - 입력받은 값에 1 씩 더하면서 출력
```

```
my_fun6 <- function(num){  
  for(x in 1:10){  
    print(num + x) # x 값이 1, 2, 3, 4, 5 ... 8, 9, 10 으로 변경되면서 수행됨  
  }  
}
```

```
# 함수 실행
```

```
my_fun6(10)
```