

# **{ 사용자 정의 함수 }**

**( R Study group 자료 )**

# 1 ) 사용자 정의 함수 (function( ))

사용 이유 ? 편리하다.

# 익숙해지면 그렇다고 한다..

반복작업을 줄여준다.

# 여러 번 작업할 것을 한번에

다른 함수와 같이 사용할 때 강력하다.

기본 구조

```
function( 값 ) {  
    출력하려는 값 or 적용할 함수 식  
}
```

# 값 : 문자, 숫자 상관 x

# 값L : 숫자(정수)만 받아 드림

ex) function(xL) { ... }

( R Study group 자료 )

# 1-1 ) 조건 함수 (if else 구문, ifelse( ))

특정 조건에 부합하는 값을 출력할 때 사용 한다.

기본 구조

```
if ( 조건 ) {  
    참일때 출력하려는 값 or 적용할 함수 식  
} else {  
    거짓일때 출력하려는 값 or 적용할 함수 식  
}
```

**# 조건은 반드시 참 거짓으로 가능한 조건만 !!**

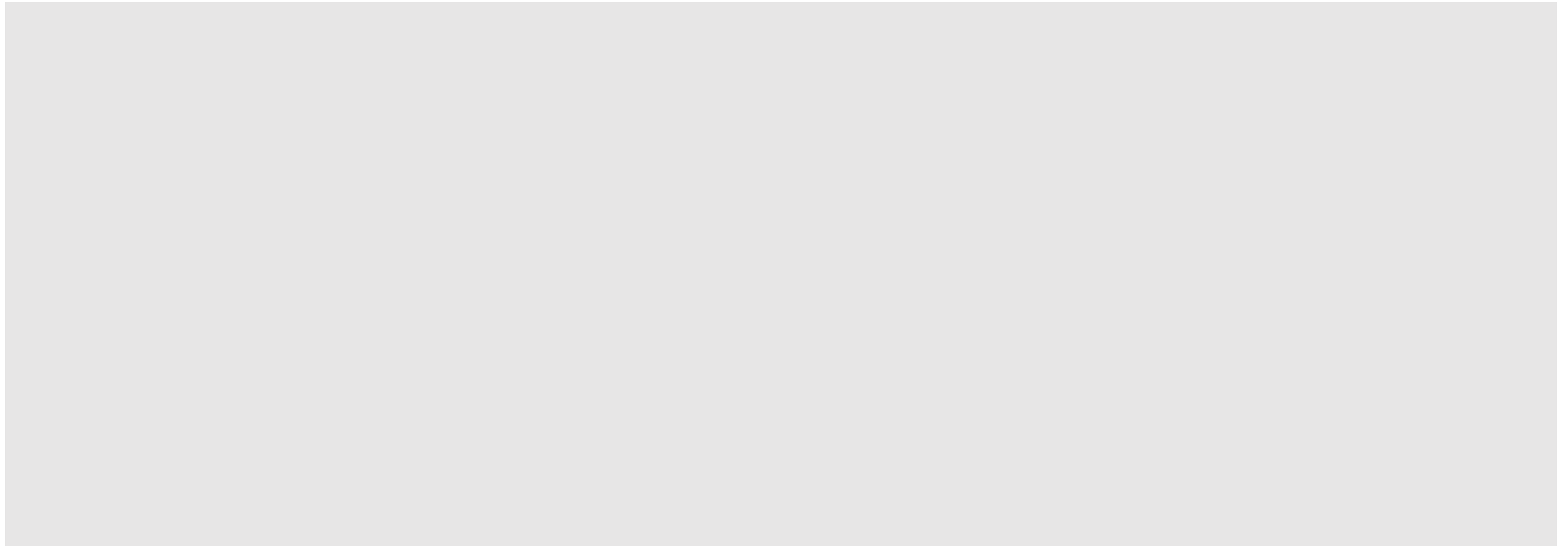
**ex )  $a == b$  (o),  $a > b$  (o),  $a + b$  (x)**

**ifelse(조건, 참일 때 나태 낼 값, 거짓일때 값)**

( R Study group 자료 )

# 1 ) 사용자 정의 함수 (function( ))

**연습문제 : 특정한 숫자가 짝수일때는 짝수, 홀수일때는 홀수 라고 나오는 함수를 만드시오.**



# 1 ) 사용자 정의 함수 (function( ))

**연습문제 : 특정한 숫자가 짝수일때는 짝수, 홀수일때는 홀수 라고 나오는 함수를 만드시오.**

```
my_fun <- function(x) {  
    if(x %% 2 == 0 ) {  
        print(paste("숫자 :", x,"는 짝수입니다"))  
    } else {  
        print(paste("숫자 :", x, "는 홀수입니다"))  
    }  
}  
  
my_fun(9)  
> [1] "숫자 : 9 는 홀수입니다 "
```

# 1 ) 사용자 정의 함수 (function( ))

**다른 함수와 같이 사용할 때 강력하다 !!**

ex ) “ , ” 같은 구분 기호때문에 숫자가 문자로 바뀔경우 “ , ” 를 제거해도 여전히 “201907” 같이 문자로 남아 있다.  
100개 혹은 더 많은 변수를 일일이 변경하기 힘들다. 그래서 쉽게 변경해 주기 위해서 function()을 사용한다.

```
df %>%  
  map_dfc(function(x) {  
    if(str_detect(x[[1]], "[0-9]")) { # str_detect() 조건과 일치  
      as.numeric(x)                  하는 값을 참으로 출력  
    } else {                          # [0-9] : 숫자전체 정규표현식  
      as.character(x)  
    }  
  }) # df에서 모든 변수들에서 첫번째 값이 숫자면 숫자 속성으로  
    # 문자면 문자 속성으로 변경
```

( R Study group 자료 )