

BUCLE WHILE

Función “while” genérica:

```
while(condición) {  
    #Función que se ejecuta mientras la condición es verdadera  
    #Se actualiza el valor de la condición  
}
```

Este bucle se repite siempre que la condición se cumpla, esta condición puede ser por ejemplo $a < 4$. Hay que tener cuidado de que el bucle no sea infinito (es infinito si la condición siempre se cumple), ya que dará error al intentar el ordenador un cálculo infinito.

Se realiza la función. Puede ser un cálculo, unir un valor a un vector o cualquier otro código.

Se debe actualizar el valor de la condición, ya que, si no sería un bucle infinito, generalmente sumando 1 a la condición, esto es muy útil ya que nos permite realizar un número determinado de “vueltas” al bucle.

En el momento en el que ya no se cumpla la condición el bucle se corta y continúa el resto del programa.

EJEMPLO DE EJERCICIO: CÁLCULO DEL ÁREA DE UN TRAPECIO

```
# inicio
```

```
B <- as.numeric(readline(prompt = "Ingresa el valor de la base grande: "))
```

```
b <- as.numeric(readline(prompt = "Ingresa el valor de la base pequeña: "))
```

```
h <- as.numeric(readline(prompt = "Ingresa el valor de la altura: "))
```

```
while (B<=b) {
```

```
  B <- as.numeric(readline(prompt = "Ingresa el valor de la base grande: "))
```

```
}
```

Si la base pequeña, b , es asignada un valor más grande o igual que a la base grande, B , entrarás en el bucle ya que cumple la condición, y se te preguntará de nuevo el valor de la base grande, B , hasta que no se cumpla la condición, es decir, hasta que B tenga un valor más alto que b .

A continuación, sigue el programa.

```
# Calcular el área
```

```
AT= (B + b)*h/2
```

```
# Mostrar el resultado
```

```
print(paste("El área del trapecio es: ", AT))
```

```
#fin
```

1. EJEMPLO EN EL QUE NO SE CUMPLE LA CONDICIÓN DEL BUCLE WHILE

```
1 B <- as.numeric(readline(prompt = "Ingresa el valor de la base grande: "))
2
3 b <- as.numeric(readline(prompt = "Ingresa el valor de la base pequeña: "))
4
5 h <- as.numeric(readline(prompt = "Ingresa el valor de la altura: "))
6
7 while(B<=b){
8   B <- as.numeric(readline(prompt = "Ingresa el valor de la base grande: "))
9 }
10 AT=(B + b)*h/2
11 print(paste("El área del trapecio es:", AT))
```

1:1 (Top Level) ↕

Console Terminal × Background Jobs ×

R 4.4.1 · ~/ ↗

```
Ingresa el valor de la base grande: 4
Ingresa el valor de la base pequeña: 3
Ingresa el valor de la altura: 2
[1] "El área del trapecio es: 7"
```

En este caso como no se cumple la condición (el valor de la base pequeña, b , es menor que el de la base grande, B), no entra en el bucle y pasa directamente a realizar la operación establecida.

2. EJEMPLO EN EL QUE SE CUMPLE LA CONDICIÓN DEL BUCLE WHILE

```
1 B <- as.numeric(readline(prompt = "Ingresa el valor de la base grande: "))
2
3 b <- as.numeric(readline(prompt = "Ingresa el valor de la base pequeña: "))
4
5 h <- as.numeric(readline(prompt = "Ingresa el valor de la altura: "))
6
7 while(B<=b){
8   B <- as.numeric(readline(prompt = "Ingresa el valor de la base grande: "))
9 }
10 AT=(B + b)*h/2
11 print(paste("El área del trapecio es:", AT))
```

1:1 (Top Level) ↓

Console Terminal × Background Jobs ×

R - R 4.4.1 · ~/ →

```
> source("~/Programación R/area_trapecio_4bueno.R")
Ingresa el valor de la base grande: 3
Ingresa el valor de la base pequeña: 4
Ingresa el valor de la altura: 2
Ingresa el valor de la base grande: 2
Ingresa el valor de la base grande: 1
Ingresa el valor de la base grande: ...|
```

En este otro caso como se cumple la condición (el valor de la base pequeña, b , es mayor que el de la base grande, B ,) entra en el bucle y no saldrá hasta que demos un valor a la base grande B mayor que al de la pequeña b . Por tanto, no podemos preestablecer el número de vueltas que se realiza, sino que puede llegar a ser infinito.