

BUCLES FOR (Loops)

El lenguaje de programación R nos permite crear estructuras repetitivas llamadas bucles.

EJEMPLO PARA ENTENDER EL CONCEPTO DE BUCLE

Imaginemos una carrera de coches de Fórmula 1. Los pilotos se encuentran en un circuito y tienen que dar un número determinado de vueltas al mismo para llegar a la meta. Supongamos una carrera de 40 vueltas.

El bucle for en este caso significa que, por cada vuelta, el piloto tiene que hacer el recorrido del circuito. Es decir, en la primera vuelta, el piloto hace el recorrido entero; en la segunda vuelta, el piloto hace otra vez el recorrido; y así hasta la última vuelta, en nuestro caso la número cuarenta, donde el piloto vuelve a realizar el recorrido. Una vez se ha completado este último, el piloto para porque ha llegado a la meta (se ha finalizado el bucle).



ESCRITURA GENÉRICA EN R

```
for(i in vnic:vfin){  
    tareas o comandos a realizar  
}
```

En el ejemplo anterior de la F1, el bucle se escribiría de la siguiente manera

```
for(vuelta in 1:40){  
    hacer el recorrido del circuito  
}
```

CONSEJO ACERCA DE LA ESCRITURA

¡Es importante la forma de colocar las llaves!

Existen muchas maneras de hacerlo. La más sencilla aparentemente es la siguiente:

```
for (i in 1:n){tareas o comandos a realizar}
```

Sin embargo, esta forma puede ser algo más laboriosa y difícil de entender cuando los comandos a realizar de la estructura repetitiva no son sencillos o cuando nos encontramos frente a bucles anidados.

Por ello, el profesor recomendó otra manera de escribir el comando:

```
for (i in 1:n){  
    tareas o comandos a realizar  
}
```

¿Como?

Se escribe for, se abre un paréntesis donde se va a escribir la variable de control, que puede ser la i o cualquier otra letra o nombre, seguido de in y luego el valor inicial, dos puntos, y el valor final que va

a adquirir la variable de control. Seguido del paréntesis colocamos la primera llave. La segunda llave, el que cierra el bucle, se coloca dos líneas por debajo, es decir, dándole al botón enter dos veces. En la línea que se ha quedado vacía entre las llaves, le damos una vez al botón de tabulación y, una vez hecho esto, escribimos los comandos a realizar.

Siguiendo estas instrucciones, los bucles y los comandos dentro de ellos se identificarán a simple vista.

BUCLES ANIDADOS (Nested loops)

Los bucles anidados consisten en dos o más bucles dentro uno dentro de otro.

EJEMPLO VISUAL

Imaginemos la situación de ir a ver una película al cine. Cuando se entregan las entradas, se especifica en ellas el número de la fila en la que se debe sentar el espectador y, dentro de esa fila, la butaca correspondiente. Supongamos que 5 amigos van al cine y quieren saber cuál es su asiento.



ESCRITURA GENÉRICA EN R

```
for (i in vnic1:vfin1){
    for (j in vnic2:vfin2){
        for (k in vnic3:vfin3){
            tareas o comandos a realizar
        }
    }
}
```

En el ejemplo visual, el bucle se programaría de la siguiente forma, siendo n el número total de filas en la sala y m el número de butacas en cada fila:

```
for(amigo in 1:5){
    for(fila in 1:n){
        for(butaca in 1:m){
            Comprobar si es el número asignado (se haría con un bucle if o condicional)
        }
    }
}
```

Una vez que se termina el bucle, ya todos los amigos sabrán cuál es su sitio y se podrán sentar en él. No obstante, para programar esta situación, sería recomendable utilizar también una estructura condicional para finalizar el bucle de las filas y las butacas en cuanto se diesen con las adecuadas y así seguir con el siguiente amigo en vez de que, como hemos escrito sin el bucle if, por cada amigo, se revisaran todas las butacas posibles en todas las filas de la sala.

CONSEJO 2

También se pueden poner comentarios detrás de la llave como #fin bucle i para ubicarse más fácilmente. Esto es de gran ayuda en los bucles anidados, pero también se puede hacer en los bucles for simples.

En el ejemplo genérico se podrían poner de la siguiente manera:

```
for (i in v1:vf1){
    for (j in v2:vf2){
        for (k in v3:vf3){
            tareas o comandos a realizar
        } #fin bucle k
    } #fin bucle j
} #fin bucle i
```