R은 라인 명령문 왼쪽에서 오른쪽으로, 위에서 아래로 순서대로 실행된다. 이를 제어하는 문장을 제어문이라 한다.

IF 문장

if (조건) {실행문1} else {실행문2}	조건을 만족하면 실행문1을 그렇지 않으면 실행문2를 실행한다.

```
sx<-sample(c(1:10),1)
y<-readline("Guess(1-10)? ")
if(sx==y){cat("You got it")} else {cat(“Boo. wrong!”)}
```

```
> sx<-sample(c(1:10),1)
> y<-readline("Guess(1-10)? ")
Guess(1-10)? 5
> if(sx==y){cat("You got it")} else {"Boo. wrong!"}
[1] "Boo. wrong!"
```

```
sx<-sample(c(1:10),1)
if(sx%%2==0){cat("Secrete Number is even.")} else {"SN is odd."}
```

```
> sx<-sample(c(1:10),1)
> if(sx%%2==0){cat("Secrete Number is even.")} else {"SN is odd."}
Secrete Number is even.
```

FOR 문장

for(변수 in 연속) {문장}	연속에 지정된 값만큼 변수 값이 변화하면서 '문장'을 반복 실행한다.

```
x <- c("apples", "oranges", "bananas", "strawberries")
for (i in x) { print(x[i])}  # wrong, x is not 1,2,..
for (i in 1:4) {print(x[i])}
for (i in seq(x)) {print(x[i])}
for (i in 1:4) print(x[i])
```

```
> for (i in x) { print(x[i])}
[1] NA
[1] NA
[1] NA
[1] NA
> for (i in 1:4) {print(x[i])}
[1] "apples"
[1] "oranges"
[1] "bananas"
[1] "strawberries"
```

처음 문장만 in 설정에 숫자가 아니므로 출력이 되지 않고 나머지 3개는 동일한 출력 결과를 보인다.

```
for (i in 1:4) cat(x[i]) #no next line
for (i in 4:1) cat(x[i]) #from 4th to 1st
for (i in 1:4) {cat(i," obs = ",x[i], "\n")}
```

```
> for (i in 1:4) cat(x[i]) #no next line
applesorangesbananasstrawberries
> for (i in 1:4) {cat(i," obs = ",x[i], "\n")}
1  obs =  apples
2  obs =  oranges
3  obs =  bananas
4  obs =  strawberries
```

- cat() 도 화면 출력이 가능하나 빈칸이나 다음 라인 없이 연속하여 출력된다.
- 그리고 in의 앞 숫자가 크면 역순으로 출력된다.

```
m <- matrix(1:10, 2)
for (i in seq(nrow(m))) {
  for (j in seq(ncol(m))) {
    print(m[i, j]) } }
```

```
> m <- matrix(1:10, 2)
> for (i in seq(nrow(m))) {
+   for (j in seq(ncol(m))) {
+     print(m[i, j])
+   }
+ }
[1] 1
[1] 3
[1] 5
[1] 7
[1] 9
[1] 2
[1] 4
[1] 6
```

- for() 문은 반복 사용하여 Nested loop 만들 수 있음
- 이를 이용하여 12단 표를 만들어 보자

```
8  *  12  =  96
****** 9 Dan *****
9  *  1  =  9
9  *  2  =  18
9  *  3  =  27
9  *  4  =  36
9  *  5  =  45
9  *  6  =  54
9  *  7  =  63
9  *  8  =  72
9  *  9  =  81
9  *  10  =  90
9  *  11  =  99
9  *  12  =  108
****** 10 Dan *****
10  *  1  =  10
```

WHILE 문장

while(조건) {문장}	조건이 만족하는 동안 문장 반복 실행된다.

```
x<-sample(1:10,1); i<-1
while (i<x) {
  print(i)
  i <- i + 1}
```

```
> x<-sample(1:10,1); i<-1
> while (i<x) {
+   print(i)
+   i <- i + 1
+ }
[1] 1
[1] 2
[1] 3
[1] 4
```

- sample(1:10, 1) 함수는 정수 1부터 10까지 숫자 중 1개를 임의 선택함
- i=1 부터 시작하여 i<-i+1 에 의해 1씩 증가한다.
- while(i<x) : i가 임의 선택된 숫자보다 작은 동안 print() 실행된다.

```
x<-c(0)
while(x<=10){
  if(x%%3==0 & x>0){cat(x, "is a multiple of 3 \n")}
  else{cat(x, "is NOT multiple of 3 \n")}
  x<-x+1}
```

```
> x<-c(0)
> while(x<=10){
+   if(x%%3==0 & x>0){cat(x, "is a multiple of 3 \n")}
+   else{cat(x, "is NOT multiple of 3 \n")}
+   x<-x+1}
0 is NOT multiple of 3
1 is NOT multiple of 3
2 is NOT multiple of 3
3 is a multiple of 3
4 is NOT multiple of 3
```

REPEAT BREAK 문장

repeat{문장 break}	{ } 계속 반복되며 조건이 만족된 경우 break 에 의해 루프를 빠져 나간다.

```
sx<-sample(1:10,1)
i<-0
repeat{i<-i+1;
  if (i==sx) { cat("Secete number =",i)
    break }}
```

```
> sx<-sample(1:10,1)
> i<-0
> repeat{i<-i+1;
+   if (i==sx) { cat("Secete number =",i)
+     break
+   }
+ }
Secete number = 3
```

이번 실행에서 (1~10) 정수 중 무작위(랜덤)하게 추출된 값(비밀번호) 3이었으므로 if() 문에 의해 i=3이 되는 순간 출력과 함께 스크립터가 종료된다.

RETERN 문장

return	콘솔에 아무 출력없이 다음 라인을 실행한다.

```
for (i in 1:9) {
  if (i%%2 == 1) {next}
  else {cat(i, "is even number \n")}}
```

```
> for (i in 1:9) {
+   if (i%%2 == 1) {next}
+   else {cat(i, "is even number \n")}}
2 is even number
4 is even number
6 is even number
8 is even number
```

실습 1 : 하이-로우 게임

(1) 1~99 정수를 하나 랜덤하게 추출한다. sample() 함수 이용 -> SN

(2) 사용자로부터 두자리 숫자를 입력하게 한다. -> GN

-> 1~99 이외 숫자를 입력하면 오류 메세지를 출력하고 재 입력하게 한다

(3) SN과 GN 비교하여 같으면 “정답” 출력하고 프로그램 종료

(4) (SN>GN) -> “UP”, (SN<GN) -> “DOWN” 출력하고 게임을 계속 진행한다.

(5) 게임 도전 회수가 7번을 초과하면 게임을 종료한다. -> 이론적으로 최대 7번이면 숫자를 맞출 수 있음

실습 2 : 2자리 스트라이크 게임

(1) 2자리(10~99) 숫자 중 각 자리가 서로 다른 숫자를 생성한다. (예) SN=93

-> 첫자리와 두번째 자리의 숫자가 같은 경우 다시 추출한다.

(2) 게임 참여자로부터 두자리 숫자를 입력하게 한다. (예) GN=32

(3) 자리수와 숫자가 맞으면 Strike, 숫자만 맞으면 Ball 로 출력하게 한다. (예) 0 strike 1 Ball 이 출력된다.

(4) 2 Strike 가 되면 게임이 종료된다.

(5) 게임 도전 회수는 5번으로 한정한다.