```
# --- COMPARISON OPERATOR ---

== # partial equality check for value but not type
!= # partial inequality check for value but not type
> # comparison operator
< # comparison operator
>= # comparison operator
<= # comparison operator

# --- LOGICAL OPERATOR ---

& # logical and
| # logical or
! # logical not
```

Control structures

```
# ----- CONTROL STRUCTURE -----

# --- CONDITIONALS ---

# IF ELSE IF ELSE

x <- 5
if (x > 0) {
    print("x is positive number")
} else if (x < 0) {
    print("x is non-positive number")
} else {
    print("this is just for edge-guarding but should logically never run")
}

# SWITCH()
    # the switch() construct allows for a degree of pattern-matching in R, the equivalent of switch case and match case statements in other programming languages
    # each comma-delimited predicate case condition listed within the switch() construct has its relationship specified with =
    # first argument in switch() is the value to be checked
    # final argument in switch() is the default fall-through value returned if all other specified predicate case conditions are unmet
    # the result of switch() constructs can be directly assigned to a variable, reminiscent of other functional languages

x <- 3
result <- switch(x,
                 "1" = "one",
                 "2" = "two",
                 "3" = "three",
                 "4" = "four",
                 "invalid number")

# --- LOOPS
```