The exit statement allows us to terminate a loop immediately without any condition being checked.

there is a difference in exit and continue. Where the continue statement stops processing the current loop and jumps back the top of the loop to carry on, the exit statement terminates the whole loop itself.

So once the processor meets the exit statement, the whole loop will finish and the program will continue to process a statement immediately following the closing statement of your loop, such as the end do or the end while.

Now keep in mind, because we can exit loops using this exit statement, it will only exit the current loop. So, if we have nested loops, and we hit our exit statement within, say, the second level of the nested loop only that level loop will terminate, and we'll go back to the higher-level loop. Now the exit statement can also be used outside of loops, just like the check statement, but you've got to be aware that if you use the exit statement outside of a loop it will terminate whatever type of code structure you are currently in.

So, for example, if you used the exit statement within a subroutine it will terminate the subroutine and processing will return to the calling program.



