

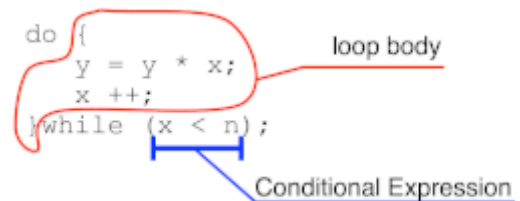
Do/While Loops

 coursera.org/learn/programming-fundamentals/supplement/mxjvk/do-while-loops

do/while Loops

Another type of loop in C is the do-while loop. Unlike a while loop, which checks its conditional expression at the top of the loop, the do-while loop checks its conditional expression at the bottom of the loop—after it has executed the body. While this distinction may seem contrived—either way the condition is checked between iterations—it is important at the start of the loop. A while loop may execute its body zero times, skipping the entire loop, if the condition is false initially. By contrast, a do-while loop is guaranteed to execute its body at least once because it executes the loop body before ever checking the condition.

The figure above shows the syntax of a do-while loop. The keyword `do` is followed by the loop body. After the loop body, the keyword `while` is followed by the conditional expression and a semicolon.



```
do {  
    y = y * x;  
    x ++;  
}while (x < n);
```

The diagram illustrates the syntax of a do-while loop. A red bracket on the right side of the code block groups the statements `y = y * x;` and `x ++;` under the label "loop body". A blue bracket at the bottom of the code block groups the `while (x < n);` statement under the label "Conditional Expression".

Execution of a do-while loop proceeds by first entering the loop body and executing all of the statements contained in it. When the execution arrow reaches the while at the end of the loop body, its conditional expression is evaluated. If the expression evaluates to true, then the execution arrow jumps back to the start of the loop body. If the expression evaluates to false, then it moves past the end of the loop and execution continues with the next statement after the loop.