

while statement`while (expression) statement`

The expression inside the parentheses is the controlling expression; the statement after the parentheses is the loop body. Here's an example:

```
while (i < n)    /* controlling expression */
    i = i * 2;    /* loop body */
```

Note that the parentheses are mandatory and that nothing goes between the right parenthesis and the loop body. (Some languages require the word `do`.)

When a `while` statement is executed, the controlling expression is evaluated first. If its value is nonzero (true), the loop body is executed and the expression is tested again. The process continues in this fashion—first testing the controlling expression, then executing the loop body—until the controlling expression eventually has the value zero.

The following example uses a `while` statement to compute the smallest power of 2 that is greater than or equal to a number `n`:

```
i = 1;
while (i < n)
    i = i * 2;
```

Suppose that `n` has the value 10. The following trace shows what happens when the `while` statement is executed:

```
i = 1;      i is now 1.
Is i < n?   Yes; continue.
i = i * 2;  i is now 2.
Is i < n?   Yes; continue.
i = i * 2;  i is now 4.
Is i < n?   Yes; continue.
i = i * 2;  i is now 8.
Is i < n?   Yes; continue.
i = i * 2;  i is now 16.
Is i < n?   No; exit from loop.
```

Notice how the loop keeps going as long as the controlling expression (`i < n`) is true. When the expression is false, the loop terminates, and `i` is greater than or equal to `n`, as desired.

Although the loop body must be a single statement, that's merely a technicality. If we want more than one statement, we can just use braces to create a single compound statement:

compound statements ► 5.2

```
while (i > 0) {
    printf("T minus %d and counting\n", i);
    i--;
}
```