

Continue statement

The continue statement has the form:

```
continue ;
```

It is syntactically correct only if it occurs in the body of a loop (do-loop, for-loop, while-loop, foreach loop)

Execution of a continue statement causes immediate termination of the innermost loop body in which it occurs.

Example. Execution of the following loop prints the values 0, 1, and 3. When k is 2 or 4, the if-condition is true, so the continue statement is executed, which terminates execution of the loop body, so the println statement is not executed.

```
for (int k= 0; k < 5; k= k+1) {  
    if (k == 2 || k == 4) continue;  
    System.out.println(k);  
}
```

Example. The example below illustrates that execution of the continue statement terminates execution of the *inner* loop body only. Execution prints the values 0 and 1. But when k is 2, the inner loop doesn't terminate because execution of the continue statement terminates the body of the inner loop, so j is never incremented.

```
for (int k= 0; k < 5; k= k+1) {  
    int j= 0;  
    while (j < 3) {  
        if (k == 2 || k == 4) continue;  
        j= j+1;  
    }  
    System.out.println(k);  
}
```

Continue statement with a label

One can write a continue statement

```
continue lab;
```

where lab is the name of a labeled statement within the method in which this continue statement appears. Execution causes control to pass to that labeled statement. We don't full describe execution because we rarely see the used for this statement and would rather not see it used.