

Chapter IV-3 — User-Defined Functions

The next example illustrates looping over all elements of a wave reference wave:

```
Function WaveWaveExample()
    Make/O wave0 = {0,1,2}
    Make/O/T textWave0 = {"A","B","C"}
    Make/WAVE/FREE ww = {wave0,textWave0}      // Make a wave reference wave
    for (WAVE w : ww)
        String name = NameOfWave(w)
        int dataType = WaveType(w)
        Printf "Name=%s, dataType=%d\r", name, dataType
    endfor
End
```

In this example, the loop variable is a wave reference named *w*. It holds a reference to a numeric wave (wave0) on the first iteration and to a text wave (textWave0) on the second.

Break Statement

A break statement terminates execution of do-while loops, for loops, and switch statements. The break statement continues execution with the statement after the *enclosing* loop's while, endfor, or endswitch statement. A nested do-while loop example demonstrates this:

```
...
Variable i=0, j

do                                // Starts outer loop.
    if (i > numOuterLoops)
        break                    // Break #1, exits from outer loop.
    endif
    j = 0
    do                            // Start inner loop.
        if (j > numInnerLoops)
            break                // Break #2, exits from inner loop only.
        endif
        j += 1
    while (1)                    // Ends inner loop.
    ...                          // Execution continues here from break #2.
    i += 1
while (1)                        // Ends outer loop.
...                              // Execution continues here from break #1.
```

Continue Statement

The continue statement can be used in do-while and for loops to short-circuit the loop and return execution back to the top of the loop. When Igor encounters a continue statement during execution, none of the code following the continue statement is executed during that iteration.

Flow Control for Aborts

Igor Pro includes a specialized flow control construct and keywords that you can use to test for and respond to abort conditions. The AbortOnRTE and AbortOnValue keywords can be used to trigger aborts, and the try-catch-entries construct can be used to control program execution when an abort occurs. You can also trigger an abort by pressing the **User Abort Key Combinations** or by clicking the Abort button in the status bar.

These are advanced techniques. If you are just starting with Igor programming, you may want to skip this section and come back to it later.