**The Range Operator (..)**

The range operator is used as a shorthand way to set up arrays.

When used with arrays, the range operator simplifies the process of creating arrays with contiguous sequences of numbers and letters. We'll start with an array of the numbers one through ten.

For example tp Create an array with ten elements that include 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10 simply write:

@array = (1..10);

You can also create an array of contiguous letters, for ecample: an array with ten elements that include A, B, C, D, E, F, G, H, I , and J. is written

@array = ("A".."J");

And, of course, you can have other things in the array definition besides the range operator.

You can create an array that includes AAA, 1, 2, 3, 4, 5, A, B, C, D, and ZZZ by the following

@array = ("AAA", 1..5, "A".."D", "ZZZ");

You can use the range operator to create a list with zero-filled numbers.

To create an array with ten elements that include the strings 01, 02, 03, 04, 05, 06, 07, 08, 09, and 10 do:

@array = ("01".."10");

And you can use variables as operands for the range operator.

To assign a string literal to $firstVar. Create an array with ten elements that include the strings 01, 02, 03, 04, 05, 06, 07, 08, 09, and 10:

$firstVar = "10";

@array = ("01"..$firstVar);

If you use strings of more than one character as operands, the range operator will increment the rightmost character by one and perform the appropriate carry operation when the number 9 or letter z is reached. You'll probably need to see some examples before this makes sense.

You've already seen "A".."Z," which is pretty simple to understand. Perl counts down the alphabet until Z is reached.

**Caution should be heeded however:** The two ranges "A".."Z" and "a".."Z" are not identical. And the second range does not contain all lowercase letters and all uppercase letters. Instead, Perl creates an array that contains just the lowercase letters. Apparently, when Perl reaches the end of the alphabet-whether lowercase or uppercase-the incrementing stops.

What happens when a two-character string is used as an operand for the range operator?

To create an array that includes the strings aa, ab, ac, ad, ae, and af. write:

@array = ("aa" .. "af");

This behaves as you'd expect, incrementing along the alphabet until the f letter is reached. However, if you change the first character of one of the operands, watch what happens.

If we create an array that includes the strings ay, az, ba, bb, bc, bd, be, and bf. we must write:

@array = ("ay" .. "bf");

When the second character is incremented to z, then the first character is incremented to b and the second character is set to a.

**Note** If the right side of the range operator is greater than the left side, an empty array is created.