



## Looping in Dynamic Text

Zephyr Associates, Inc.  
P.O. Box 12368  
Zephyr Cove, NV 89448

775-588-0654  
800-789-5323  
Fax 775-588-8423

[www.styleadvisor.com](http://www.styleadvisor.com)

## **Let's get loopy**

Now that you're an accomplished DT nerd and you're looking for more Kung Fu to add to your repertoire, let's talk about making the program loop.

A loop is a set of instructions that you want to execute repeatedly, either for a specific number of times, or while a specific condition is true, or until a specific condition becomes true. I worded that definition that way so that I could introduce you to the three kinds of loops that DT understands: For loops, While loops, and Repeat Until loops.

## **For Loops**

Use a For loop to run a set of instructions a specific number of times. The syntax is:

```
for counter in Range(start value, end value, step size) do
  instructions
end
```

Such a loop will set a variable (I called it counter, but you can call it whatever you like) to the starting value, execute the instructions you supply, add 1 to your counter variable (or add whatever you set for your optional step size), then do all the instructions again, and stop when your counter gets past the ending value you supplied.

## **While Loops**

Use a While loop to run a set of instructions while a condition is true. The syntax is:

```
while condition is true do
  instructions
end
```

Such a loop will check to see if the condition is true and if it is execute the set of instructions (which must be set up to change the condition if you ever want the loop to finish), then check the condition again and either end or run the instructions again.

*A loop that never passes the test is called an infinite loop. StyleADVISOR would rather not wait forever to see if a loop will end, so what it does is count DT instructions and when it gets to a very large number it stops and asks you if you think you might have made an infinite loop. If you don't believe you have (say you're not using any loops at all) there is a way to increase the number of DT instructions StyleADVISOR will allow before it gives up and accuses you of being a bad programmer. That'll be covered in another lesson.*

## ***Repeat Until Loops***

Use Repeat Until loops to run a set of instructions until a condition becomes true. The syntax is:

```
repeat
  instructions
until condition is met
```

Such a loop will run the set of instructions (again, the instructions will need to affect the condition you're testing if the loop is ever to complete) then check to see if the condition is true. If it is not, the instructions get run again, then the condition is checked again, and so on until the condition is true and the loop stops.

The big difference between the last two types of loops is at what point the conditions are tested. With While loops you check the condition first then only do the instructions if the condition is true. With Repeat Until loops you always do the instructions at least once before you test to see if the condition is true.

## Examples

### For Loops

Suppose you want to display a manager's Morningstar star rating with stars instead of just as a number. Here's an opportunity to use a For loop.

```
stars = Value(1,"Star Rating Inception", "Page 2|Scan #1")

for counter in Range(1, stars) do
  Display("**")
end
```

This example gets the star rating for the first manager and assigns it to a variable named "stars". Then it runs a For loop that sets the variable called "counter" to 1, checks to see if 1 is less than or equal to "stars", and if it is, displays an asterisk. Then it automatically adds 1 to "counter" (if you don't specify a step size you get a step of 1), goes back to the top and does it again, if 2 is less than or equal to "stars" it displays another asterisk, and so on until there are as many asterisks as the manager's star rating.

*In this snippet of code I'm not bothering to check the validity of the Star Rating Inception field from Morningstar. I'm assuming that it'll be a number from 1 to 5. This is not a smart assumption, since I don't have any control over what is in the Morningstar database, but I didn't want to clutter up the example. Always check for bad input data if the data is outside your control.*

### While Loops

To illustrate the difference, we'll use a While loop to count stars, too.

```
stars = Value(1,"Star Rating Inception", "Page 2|Scan #1")
counter = 1

while counter <= stars do
  Display("**")
  counter = counter + 1
end
```

You'll notice that this loop took a lot more code than the For loop. First we set some variables, including the initial value for our counter. Then the loop checks to see if the value of the counter variable is less than or equal to the number of stars to draw. If it is, we display an asterisk and we add 1 to the counter variable, then we go back to the top and check to see if we're less than or equal to the number of stars again.

Reasonable people probably wouldn't use a While loop for this application, since the For loop was simpler, but there are times when this type of loop makes more sense. Use this type of loop when you want to do a set of calculations while some complex conditions are true, rather than just counting how many times you've been through the loop.

### ***Repeat Until Loops***

Now we'll count stars with a Repeat Until loop just to drive the differences home:

```
stars = Value(1,"Star Rating Inception", "Page 2|Scan #1")
counter = 1

repeat
  Display("*")
  counter = counter + 1
until counter > stars
```

Again we have to initialize our counter variable to the start value and increment it by adding 1 each time through the loop. So this block of code does the following: First we set "stars" to the star rating, then we initialize "counter" to 1, then we go into the loop. Since there's no test at the beginning of the loop we're always going to go through this loop at least once. Once inside we display an asterisk, then we increment "counter" by adding 1 to whatever its current value is. Only then do we test to see if we need to go back to the top. Is "counter" greater than "stars"? If so then we're done. If not we have to go through the loop again.

Because of where the test is done, this time we had to check to see if our counter variable was bigger than the number of stars instead of checking to see if it's less than or equal.

Again, a reasonable person would not use Repeat Until when a For loop would be simpler, but we're not reasonable people ... we're learning DT. Use this type of loop when you want to do a calculation over and over until some complex conditions are met...conditions that can't easily be expressed by a counter variable hitting a specified maximum.