Python Looping Notes: Which Way?

A common question among beginning programmers involves which way to code a loop. Generally, a given loop can be coded in a variety of ways; it's all a matter of which form is the most convenient, efficient and clear. The `while` loop is the most general form but is not always the most efficient. The `while` form is almost always used when a certain unique set of conditions need to be met to continue or terminate the loop. In these instances, the precise number of loop iterations is not necessarily known at the start of the loop. In contrast, there are loops that need to be executed a specific number of times or for a specific set of values. These can be coded using either the `for` or the `while` form. The following are guidelines with examples:

If you need to do something a certain number of times, a `for` loop with the default `range()` iterator is the logical choice. For example, if you need to do something 5 times, consider the following:

```
for x in range(5):
      # code to loop goes here
```

Note that you can do this with a `while` loop but it takes more work:

```
x=0
while x<5:
      # code to loop goes here
      x=x+1
```

If you need to cycle a variable between an initial and an ending value, changing it by a constant offset each time, a `for` loop with the `range()` iterator is probably the best bet, this time using all three arguments. For example, if you need to start a variable at 10 and increase it by 5 each time until it gets to (but not including) 80, consider the following:

```
for x in range(10, 80, 5):
      # code to loop goes here
```

If you need to run through a bunch of values that are more or less randomly arranged, a `for` loop with an item list will generally prove most efficient. For example, if you need to run through a list of resistor values such as 3300, 1200, 6800, 5600; consider the following:

```
for x in 3300, 1200, 6800, 5600:
      # code to loop goes here
```

Finally, if you need to cycle through a variable in a regular manner but which is not a simple increment, the `while` loop is the choice to make. For example, suppose you need to cycle through a set of voltages that begins at 5 volts and doubles until it reaches or exceeds 1000 volts. The following would do nicely:

```
v=5
while v<1000:
      # code to loop goes here
      v=v*2
```