

HOW TO GET PSEUDO-RANDOM NUMBERS (VISUAL BASIC 6)

WHY PSEUDO?

Most computers (perhaps all computers) **don't generate truly** random values. To be truly random, the value must not be predictable. However, computers use *complex mathematical formulas* to generate values which *look* random, but really aren't. Even though the formulas are complex, they're still predictable.

Conceptually, random numbers generated by a computer can be thought of as a very, very long list of values. The computer always returns the values in the same order, but it doesn't always start at the same place in the list.

If the list contained a million values, sometimes the computer would return values starting at the first value, but sometimes it might start at the **105,768th** value. Because the starting point is unknown and the list is so long, a repeatable pattern is not discernable.

INTRO – THE RND FUNCTION

The first thing you need to know is the function that generates a **random number**. The function is called : **Rnd**
Here is how you use it:

- 1) Open a new project in **VB**.
- 2) In the load event of your form add the following code:

```
Private Sub Form_Load()  
Dim i As Integer  
For i = 1 To 10  
Debug.Print Rnd  
Next  
End Sub
```

- 3) Run the project. In the Immediate window you will notice that 10 numbers have been printed.

THE RANDOMIZE STATEMENT

After you run the program once you have the values in the Immediate window. Run the program again and compare the first 10 numbers with the last 10 numbers. Notice a similarity? The Rnd function has a predetermined set of numbers that it shows every time call the rnd function. So every time you start your app, it starts from the beginning showing the same numbers again. How do you avoid this? The Randomize statement makes the Rnd function generate truly random numbers.

TIP : Don't keep using the **randomize** statement because then it loses it's meaning... Use it only once, right before you start creating random numbers the first time.

TIP : You also have the *optional number after in the randomize statement*. That number will cause the numbers to be generated randomly but based on that number. So **Randomize 2** will always return the same numbers. This can be used in encryption algorithms...

Randomize uses number to initialize the **Rnd** function's random-number generator, giving it a new seed value. If you omit number, the value returned by the system timer is used as the new seed value.

If **Randomize** is not used, the **Rnd** function (with no arguments) uses the same number as a seed the first time it is called, and thereafter uses the last generated number as a seed value.

Note : To repeat sequences of random numbers, call **Rnd** with a negative argument immediately before using **Randomize** with a numeric argument.

Using **Randomize** with the same value for number does not repeat the previous sequence.

```
Private Sub Form_Load()  
Randomize  
Dim i As Integer  
For i = 1 To 10  
Debug.Print Rnd  
Next  
End Sub
```

Now that we added the **Randomize** statement you can see that every time you run the application, different numbers are produced. If on the other hand you do this...

```
Private Sub Form_Load()  
Randomize 8  
Dim i As Integer  
For i = 1 To 10  
Debug.Print Rnd  
Next  
End Sub
```

...you will notice that the same numbers are generated (the encryption I was saying)...