

제 4 강. Loop문과 배열이 만나면 무적이 된다

반복문(Loop)

VBA를 사용하는 가장 큰 목적 중의 하나가 바로 귀찮은 일을 VB를 사용하여 엑셀에게 수행하도록 명령을 내릴 수 있다는 것이다. 반복해서 무엇인가를 시킬 때 사용하는 명령어가 바로 Loop문이다. Loop문에도 여러 가지 종류가 있는데(일 시키는 방법도 가지가지다.), 많은 종류의 Loop문을 한꺼번에 할 필요는 없다. 이번 강의에서 딱 한가지의 Loop문을 배울 것인데, 이것만 잘 써도 앞으로 코딩을 함에 있어서 별다른 어려움이 없을 것이다! 그만큼 중요하다는 것이다.

For ... Next Loop

정말이지 말이 필요없다. 이 For Next 구문 하나면 만사 O.K. !! 먼저 정확한 문법을 아는 것이 중요하니 다음을 숙지하자.

For ... Next Loop Syntax

```
For counter = start To end [Step increment]
    Statement1
    Statement2
    ..
Next [counter]
```

[]표시가 의미하는 것은 써도 그만, 안 써도 그만을 의미한다. Step increment의 경우 안 쓰면 기본으로 1이 설정된다. For Next 구문을 이용하여 엑셀에게 숫자 쓰기를 가르쳐보자.

[Example 4.1]

```
Sub Write_fast()
```

```
    Dim i As Integer
```

```
    For i = 1 To 100
```

```
        Sheets(1).Range("A1")(i, 1) = i
```

```
    Next i
```

```
End Sub
```

위의 코드를 입력하고 F5를 누르는 순간! 우리 빈둥거리던 엑셀이 미친 듯이 숫자를 적기 시작한다!! 겁나 빠르다. 빠름~ 빠름~ 빠름~♫ 이왕 한 것 Step기능까지 한 번 써보자.

[Example 4.2]

```
Sub Write_fast()
```

```
    Dim i As Integer
```

```
    For i = 1 To 100 Step 2
```

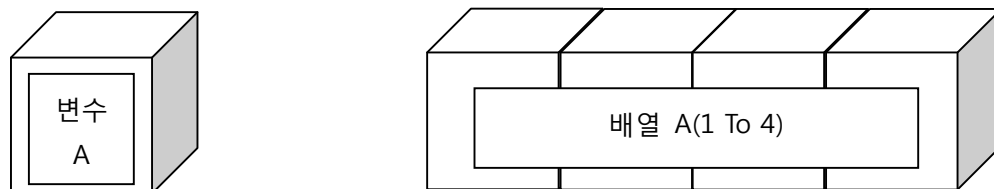
```
        Sheets(1).Range("A1")((i + 1) / 2, 1) = i
```

```
    Next i
```

```
End Sub
```

배열(Array)

지난 3강의 숙제를 제대로 해 본 사람이라면 앞으로 배열 배열의 고마움에 정말 눈물이 날 지도 모른다. 3강의 숙제를 하기 위해서는 변수를 발생하고 싶은 만큼 일일이 선언을 해줘야 했다. 아마 난수 10000개를 뽑아서 평균을 구하려 한다면 변수 선언하는 데만 반나절이 걸릴지도 모를 일이다. 변수 하나를 선언하는 것을 상자 하나를 만드는 일이라고 비유한다면, 배열은 상자가 여러 개 붙어있는 것을 한꺼번에 만들어 준다.



즉, 상자 여러 개를 한번에 선언하는 방법이 있다는 말이다. 배열 선언 방법은 다음과 같다.

Dim [배열 이름][배열 크기] as [Data Type]

수학을 즐겨 하는 독자라면 행렬을 선언하는 방법이라고 생각하여도 무리가 없다. 말로 듣는 것 보다는 한번 해보는 것이 훨씬 효과가 크다!

[Example 4.3] [Array & Loop 문 사용]

```
Sub random_sample()
```

```
    Dim X_sample(1 To 10) As Double    '<- 이렇게 배열을 선언한다!
```

```
    Dim X_Sum As Double
```

```
    Dim i As Integer
```

```
    For i = 1 To 10
```

```

    X_sample(i) = 2 * Rnd() + 3
    X_Sum = X_Sum + X_sample(i)
Next i

Dim X_bar As Double
X_bar = X_Sum / 10

Dim S_Variance As Double

For i = 1 To 10
    S_Variance = S_Variance + (X_sample(i) - X_bar) ^ 2
Next i

S_Variance = S_Variance / (10 - 1)

Sheets(1).Range("B1")(1, 1) = X_bar
Sheets(1).Range("B1")(2, 1) = S_Variance

End Sub

```

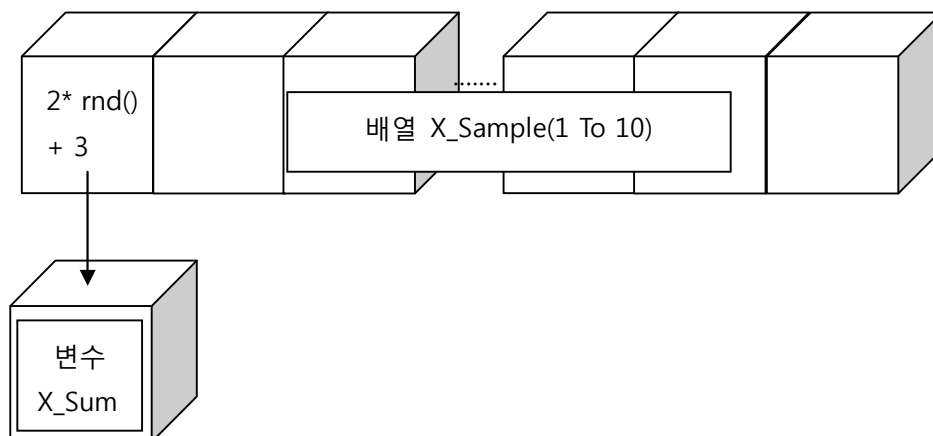
위 예제를 혼자서 잘 뜯어보면, 프로그래밍 실력이 쑥~욱 늘어날 것이다. 주의해서 봐야하는 코드는 바로 For Next Loop문이다.

```

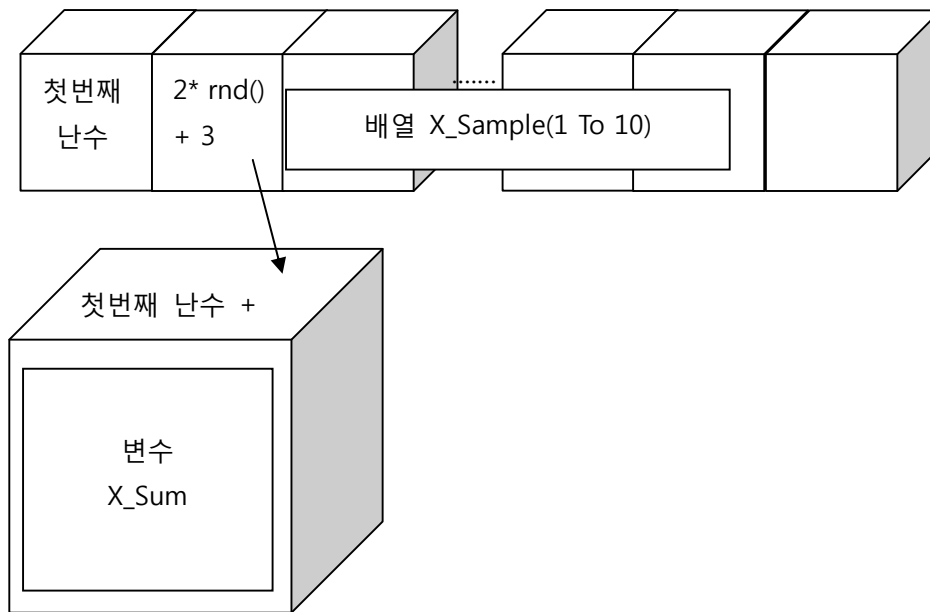
For i = 1 To 10
    X_sample(i) = 2 * Rnd() + 3
    X_Sum = X_Sum + X_sample(i)
Next i

```

첫번째 i = 1 일 때의 상황을 그림으로 그려보면 다음과 같다.



첫번째 $i = 2$ 일 때의 상황을 그림으로 그려보면 다음과 같다.



이러한 방식으로 $i = 1$ 에서 10까지 시행하면, 배열 X_Sample 열 칸에는 열 개의 난수가, 변수 X_Sum 안에는 10개 난수의 합이 들어있게 된다.

이제부터 우리가 이제까지 배운 모든 것을 총 동원하여 우리가 원하는 난수의 수를 입력하면 난수를 발생시키고, 그에 따른 표본평균과 표본분산을 계산하여 주는 엑셀 파일을 작성해보도록 하자. 우선 엑셀 시트상에 디자인은 임의대로 아래와 같이 꾸며보았다. 난수 개수가 100으로 고정된 상태에서, "C5", "C6"셀에 우리가 원하는 평균과 분산을 입력하면 지정한 평균과 분산을 따르는 정규분포에서 난수를 발생시켜주는 간단한 프로그램이다. (엑셀 시트는 첨부파일 참조)¹

	K12		fx					
	A	B	C	D	E	F	G	H
1	$X \sim Normal(\mu, \sigma^2)$							
2								
3								
4		n	100					
5		μ						
6		σ^2						
7		X_bar						
8		S ²						
9								
10								
11		발생시킨 난수						
12								
13								
14								
15								
16								

¹ 엑셀 시트에 수식 입력은 Mathtype이라는 프로그램을 이용하여 입력했다.

즉, 사용자는 "C5"와 "C6"을 입력하고, 프로그램은 "C4"와 "C7"과 "C8", 그리고 "B12"부터 아래쪽의 셀에 계산 값을 반환해준다. 위와 같은 엑셀 시트 상의 세팅이 끝나면 VBA를 열어 모듈 1에 아래의 코드를 입력해보자. 기본적인 틀은 Example 4.3과 똑같지만 사용자가 입력할 때마다 프로그램이 반응하기 위하여 살짝 수정하였다. 어떤 부분이 바뀌었는지 보고 왜 이렇게 바꾸었는지 생각해보자.

[Example 4.4]

```
Sub Random_Sample()
```

```
    Dim n As Long, mu As Double
```

```
    Dim sigma As Double
```

```
    n = Sheets(1).Range("C4")(1, 1)
```

```
    mu = Sheets(1).Range("C4")(2, 1)
```

```
    sigma = Sqr(Sheets(1).Range("C4")(3, 1))
```

```
    Dim i As Integer
```

```
    Dim X_Sample(1 To 100)
```

```
    Dim X_Sum As Double
```

```
    For i = 1 To n
```

```
        X_Sample(i) = mu + sigma * WorksheetFunction.NormSInv(Rnd())
```

```
        X_Sum = X_Sum + X_Sample(i)
```

```
    Next i
```

```
    Dim X_bar As Double
```

```
    X_bar = X_Sum / n
```

```
    Dim S_Variance As Double
```

```
    For i = 1 To n
```

```
        S_Variance = S_Variance + (X_Sample(i) - X_bar) ^ 2
```

```
    Next i
```

```
    S_Variance = S_Variance / (n - 1)
```

```
    Sheets(1).Range("C7")(1, 1) = X_bar
```

```
    Sheets(1).Range("C7")(2, 1) = S_Variance
```

```
    For i = 1 To 100
```

```
Sheets(1).Range("B12")(i, 1) = X_Sample(i)
Next i
```

End Sub

위 코드를 보면서 '아니 왜 n은 고정시켜 놓은 거야?' 하는 의문이 들지 모르겠지만, 일단은 봉인이 걸려있다고 생각하자. 몇 가지 지식을 좀 더 배운 후에 봉인을 풀어주겠다. 그때야 말로 비로소 프로그램 다운 프로그램이 되겠지만 일단 현 지식을 가지고 최대한 근접해야겠다는 마음으로 잘 따라오기 바란다.

코드를 해석해보면, 위의 코드는 다음과 같은 구문을 통하여 사용자가 원하는 정보를 변수에 담는다. 우리가 VB에서 다루던 계산한 결과값을 엑셀로 송출했듯, 이번에는 반대로 엑셀에서 VB로 송신을 해온 셈이다.

```
n = Sheets(1).Range("C4")(1, 1)
mu = Sheets(1).Range("C4")(2, 1)
sigma = Sqr(Sheets(1).Range("C4")(3, 1))
```

이렇게 입력 받은 변수에 담긴 사용자가 원하는 정보는 나중에 난수를 발생시킬 때 반영된다.

```
X_Sample(i) = mu + sigma * WorksheetFunction.NormSInv(Rnd())
```

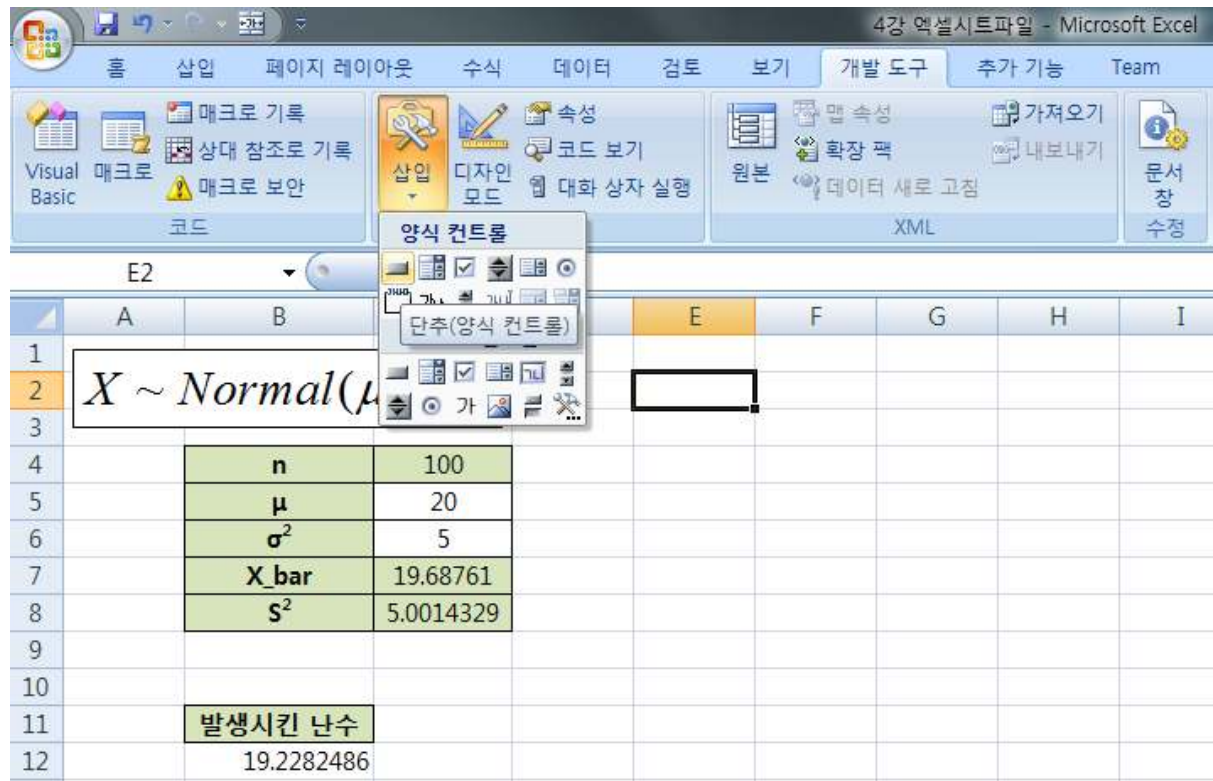
위 코드를 통하여 X_Sample() 배열의 각 칸에 받아놓은 난수들은 평균이 mu이고, 분산이 σ^2 인 정규분포를 따르게 된다. 마지막 작업으로 난수들을 엑셀시트에 뿌리는 작업이 남아 있다. 역시나 오늘 배운 것을 복습하기 위해서 For next 구문을 사용하였다.

```
For i = 1 To 100
    Sheets(1).Range("B12")(i, 1) = X_Sample(i)
Next i
```

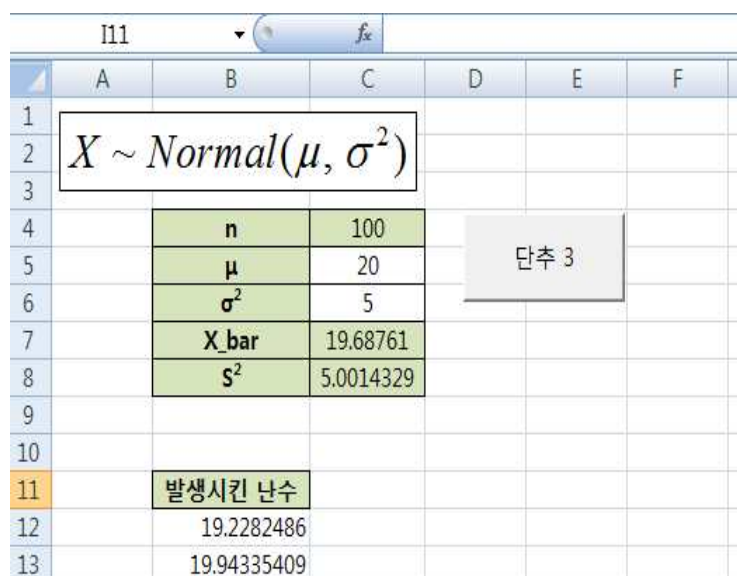
이제 엑셀 시트 상에서 원하는 평균과 분산을 입력하고, VBA의 코드를 실행시켜보자! 엑셀 시트에 우리가 뽑은 난수들이 짜악 생성이 될 것이다. 뿌듯하지 않은가? 다시 평균과 분산을 바꿔서 실행시켜봐도 잘 돌아간다! 판타스틱~ 판타스틱~♪ (가끔 위드에도 배경음악을 넣고 싶은 충동이 있다.) 하지만 아직 끝난 것이 아니다. 정규난수를 발생시키는 엑셀프로그램을 기껏 만들어 놔는데 발생할 때 마다 VBA를 열어서 실행시켜야 한다면 너무 귀찮다. 또 하나 아주 중대한 문제는 이렇게 VBA를 통하여 매번 실행을 한다면 VBA를 모르는 내 친구들에게 자랑을 할 수 없다. 매번 우리의 무지한 친구들에게 VBA의 존재를 일일이 설명해 줄 수 없는 노릇이니, 그들을 위한 어떤 조치가 필요하다. 그래서 엑셀에서는 우리가 짜놓은 코드와 연결시켜주는 여러 단추를 제공하는데, 이 단추를 써서 우리들의 무지한 친구들에게 "야, 아무것도 알 거 없고 그냥 평균, 분산 입력하고 이거 눌러." 라고 말하면서 엑셀 파일을 건네 줄 수 있는 것이다. 이렇게 하면 그들의 눈에

는 우리가 좀더 더 간지가 흘러 보이는 추가적 효과도 발생한다.

엑셀에서 [개발도구]->컨트롤 영역의 [삽입]->[양식컨트롤] 중에서 단추양식 컨트롤을 클릭하여 엑셀 시트에 예쁘게 그려보자.



단추를 그리면 짤~ 하면서 매크로 지정이라는 창이 뜨는데 일단, 취소를 누른다. (사실 여기에서 지정할 수 있지만 일단 넘겨보자.) 현재 엑셀 시트상에는 다음과 같이 단추와 테이블이 딱 위치하고 있을 것이다.



이 화면이 여러분의 친구들이 엑셀 파일을 켜고 난 후 볼 수 있는 화면이다. 단추를 더블클릭을 하면 단추 이름을 바꿀 수 있다. '난수생성'이라는 이름으로 바꿔주어 누르고 싶은 충동을 일으키게끔 하자. 아직까지는 단추를 눌러도 아무런 일이 일어나지 않는다. 단추와 우리가 짤 코드가 연결되어 있지 않기 때문이다. 이제 단추와 코드를 연결시키는 작업을 해보자. 단추에 커서를 놓고 오른쪽 클릭을 하여 [매크

로지정] 을 누르면 우리가 짜놓은 Random_Sample이라는 이름의 할 일 목록이 뜰 것이다. 즉, 난수 생성이라는 단추를 누르면, 엑셀이 자동으로 VB에게서 신호를 받아 Random_Sample Sub문 안에 할 일 들을 수행하게끔 해 준 것이다. 이제 커서를 단추 위로 가져가보면 커서가 바뀐다. 평균, 분산을 입력 했는지 확인하고 단추를 눌러보자! 다시 한번 판타스틱~ 판타스틱~♪

	A	B	C	D	E	F	G	H
1								
2		$X \sim Normal(\mu, \sigma^2)$						
3								
4		n	100					
5		μ	0					
6		σ^2	1					
7		X_bar	-0.011444					
8		S^2	0.9443556					
9								
10								
11		발생시킨 난수						
12		1.249470233						
13		1.108754915						
14		-0.294161578						
15		1.818402554						
16		0.007879043						
17		2.14798839						
18		-0.133636237						
19		0.769277499						

이제 친구들에게 자랑하기 위해서는 저장을 해야 하는데 파일 형식을 꼭 'Excel 매크로 사용 통합 문서'로 해야 한다. 만약 다른 파일 형식으로 저장을 할 경우, 단추와 코드의 연결 고리가 끊겨서 단추를 눌러도 아무 일도 일어나지 않는 경우가 발생할 수 있다.

이번 강의는 여기서 마치도록 하자. 내용이 길다 보니 다시 한번 복습하는 것을 숙제로 남긴다. 다음 강의에서는 이번 프로젝트의 봉인을 해제 해 보도록 할 터이니 오늘 강의를 꼭! 복습하기 바란다.

[Homework]

1. [Example 4.4] 복습