

제 2 강. 변수 정의 하기

Visual Basic이라는 리모컨 다루기

1강에서 엑셀을 원격조종 할 수 있게끔 만들어주는 Visual Basic이라는 리모컨이 있다는 사실을 배웠다. 하지만 아직까지 여러분은 이 VB라는 리모컨을 다루는 방법을 모른다. 우리가 말을 할 때에 문법에 맞게 말을 해야 사람들이 잘 알아들을 수 있듯, 이 리모컨을 다루는 방법을 알아야 엑셀이 잘 알아먹고 작동을 하는 것이다. 일단 문법에 맞지 않게 리모컨을 다룰 때 일어날 수 있는 경우를 알아보자.

1강에서 배웠던 코드를 응용해서 엑셀에게 덧셈 한번 시켜보자.

[Example 1]

```
Sub test()
```

```
    A = 1
```

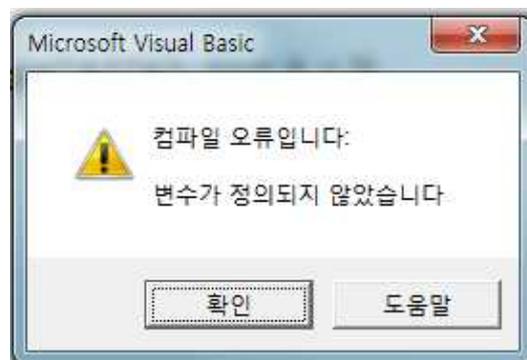
```
    B = 2
```

```
    Sheets(1).Range("A1")(1, 1) = A + B
```

```
End Sub
```

위 코드는 필자가 리모컨에게 초등학생들에게 덧셈을 설명해주듯 알려주는 상황을 떠올리면 쉽다. '자 리모컨(VB)아, A라는 애가 과일 1개를 가지고 있어. 그리고 B라는 애는 과일 두 개를 가지고 있어요. 이제 A랑 B가 가지고 있는 과일 전체 개수를 엑셀한테 알려주고 와~'

하지만 위의 코드를 입력한 후 재생버튼을 눌러보면 1강에서와 같이 엑셀에 딱~ 하니 3이라는 숫자가 써 있어야 할 것 같지만, 웬걸 리모컨이 작동 않는다는 오류 화면을 보여준다. (이 오류 화면은 우리가 조종하고 있는 Visual Basic에서 오류가 난 것이다. 엑셀은 여전히 놀고 있는 중이다.)



친절하게도 리모컨에서는 무엇 때문에 작동하지 않는지 이유를 우리들에게 알려주었다.

컴파일 오류입니다.
“변수가 정의되지 않았습니다.”

컴파일 이라는 것은 VB에서 우리의 코드를 시험으로 한번 테스트 하는 과정을 의미하는데, 이 과정 중에서 오류가 생겼다는 것이다. 리모컨이 스스로 일일이 우리 코드가 잘 쓰여졌는지 체크해주는 것이다. 실로 대단한 리모컨이 아닐 수 없다. 이러한 체크 작업 중 ‘**변수라는 것이 정의 되지 않아** 리모컨 작동하지 않는 거예요~’ 하고 알려준 것이다.

Variable(변수)

‘아, 이거는 뭐 수학도 아니고 변수는 또 뭐고 정의까지 하라니 미취~버리겠네!’ 하고 생각하는 독자가 있을 지 모르겠지만, 수학을 싫어하는 독자들에게는 애석하게도 수학에서 쓰이는 변수 맞다. 충격적인 사실을 애써 뒤로 한 채 확인 버튼을 눌러 이 사실을 외면해보면, 아까 입력한 코드에 A 부분에 노란색으로 하이라이트가 되어 있을 것이다. 이 노란색은 Visual Basic에서 우리들에게 ‘이 부분이 이상해요.’ 하고 알려주는 힌트이다. ‘A가 어찌라고?’ 생각하는 독자들을 위해서 아까 상황을 예로 설명해보겠다. 문제가 되는 대목은 바로 첫 문장이다.

‘자, 리모컨아 A라는 애가 과일을 가지고 있어~’

이런 말을 들었을 때, 우리의 똑똑한 Visual Basic은 다음의 두 가지 질문을 한다.

1. A라는 아이는 이름이 뭐야? 과일 좋아해? 힘은 세?
2. 과일이려면 어떤 과일이야? 사과? 배? 아니면 수박?

일단, Visual Basic은 아이가 어떤 아이인지를 궁금해한다. 처음 만났는데 통성명은 당연한 것 아닌가? 또한 과일을 싫어하는 아이에게 과일을 주었을 때는 안 받는다고 고개를 절레절레 흔드는 최악의 사태가 벌어지는 것을 막기 위해서 과일을 좋아하는지에 대한 여부를 알아야 한다. 마지막으로 아이의 힘이 어느 정도 인지에 따라서 각자가 들 수 있는 과일이 정해진다.

어떤 아이는 힘이 약해 사과 정도 밖에 들 수 없는 반면에 다른 힘센 아이에게는 수박을 주어야 한다. 행여 힘센 아이에게 사과를 들라 하면 시시하다면 버려버릴 지도 모른다. 따라서 **우리는 Visual Basic에게 아이의 이름이 무엇인지, 어떤 과일을 좋아하는지, 어느 정도 힘이 센지, 주고 싶은 과일이 어떤 과일인지 알려줘야 하는 것이다.**

변수의 정의

위의 예에서는 변수는 VB에게 소개하는 아이이다. 따라서 변수의 정의라는 것은 VB에게 어떤 아

이인지 설명해 주는 것을 의미한다. VB에게 아이들을 소개시켜 줄 때에는 다음과 같은 방법으로 소개시켜 주어야 한다.

Dim [아이이름] as [힘 센 정도 & 성향]

조금은 이상하게 느껴질 수 도 있겠지만 어찌겠는가? 저렇게 해야 알아먹는 것을! 자, 이제 앞의 방법을 이용하여 다시 Example 1을 수정하여보자.

[Example 2]

```
Sub test()
```

```
    Dim1 A as integer
```

```
    Dim B as integer
```

```
    A = 1
```

```
    B = 2
```

```
    Sheets(1).Range("A1")(1, 1) = A + B
```

```
End Sub
```

이제 재생 버튼(F5)을 눌러보자! 사사삭 하면서 리모컨이 우리가 쓴 코드의 검토도 성공적으로 끝나서 놓고 있는 엑셀에게 명령을 내려봤다. 첫번째 시트의 "A1"셀에 딱 하니 3이라는 숫자가 써있는 것을 확인 할 수 있다. 조금은 감이 오는가? 우리가 수학시간에 배운 변수의 개념은 바로 값이 변할 수 있는 어떤 가상의 수였다. 예제에서 보면 A라는 아이에게 1을 들고 있게 만들 수도 있고, 맘에 안 들면 2를 들고 있게 만들 수도 있는 것이다. 지금까지의 내용을 프로그래밍 용어로 정리해보면 다음과 같다.

변수는 숫자를 보관할 수 있는 공간을 의미하는 개념이다. 변수를 정의하고 난 후에야 우리는 변수에 정보를 저장할 수 있다.

변수를 정의하는 방법은 다음과 같다.

Dim [Variable Name] as [Variable Data Type]

Data Type

¹ Dimension의 줄임 말이다. 책에서는 변수를 정의한다는 용어를 썼지만, 변수를 선언한다 (Declare)는 말로도 많이 쓰인다.

여러분이 앞으로 친하게 지내야 되는 아이들의 힘센 정도와 성향을 나타내 보았다. 처음부터 다 외우려 들지 말아라. 자연스럽게 외워진다. 멘붕금지다!

	Size(힘센 정도)	Description(성향)
Boolean	2	논리적인 값 True of False
Integer	2	큰 정수
Long	4	훨씬 큰 정수
Single	4	자세한 실수
Double	8	엄청 자세한 실수
Currency	8	소수점이 4자리로 고정된 수 실수
String	Length of string	긴 문자

위와 같이 각 아이들의 근력과 성향(변수 타입)에 맞게 과일(데이터)을 건네주어야 Visual Basic이 화를 내지 않는다. 이 말을 좀 더 프로그래밍 공부에 맞게 바꿔서 말하면 다음과 같다.

정의한 변수의 타입에 맞는 데이터를 넣어야 문제가 발생하지 않는다.

이해를 돕기 위해 다음의 예제를 살펴보자. 일단, 문제를 입력하기 전에 아래 코드를 입력하면 어떤 일이 발생할 지 예측해보자.

[Example 3]

```
Sub test()

    Dim A as integer
    Dim B as double

    A = 3.2
    B = 2

    Sheets(1).Range("A1")(1, 1) = A + B

End Sub
```

입력하였다면 F5키를 눌러 확인해보자. 만약 오류가 날 것이라고 생각한 학생이 있다면 의아해 할 것이다. 아무런 오류도 없이 잘 돌아가기 때문이다. 오류가 나지 않는다는 것은 Visual Basic 이 컴파일을 제대로 하였다는 의미인데, 그럼 저 코드에는 오류가 없다는 말이 아닌가? 이거 너무 대충 가르치는 거 아니야? 라고 필자를 원망하기 전에! 엑셀이 우리에게 알려준 결과를 확인해보자.

3.2 + 2 = 5 (!?)

이런 황당한 결과는 **정의한 변수의 타입에 맞는 데이터를 넣어야 한다는 원칙을 지키지 않아서** 발생한 결과이다. 즉, A라는 변수는 정수만을 담을 수 있는 변수이므로 3.2의 정수부분인 3만을 가지고 있게 된다. 사과 무게밖에 들지 못하는 아이에게 수박을 건네주면, 아이는 사과 정도 무게만 잘라서 들고 있고 나머지는 버려버린다. 나중에 수박을 왜 버렸냐고 화를 낼 것인가? 사과 밖에 들지 못하는 아이에게 수박을 건네준 사람이 더욱 잘못이 크다.

‘아니, 그래 내가 잘못했다고 쳐도, 그리 똑똑한 Visual Basic은 왜 가만히 있는 거야?’ 하고 따질 수 있겠다. 똑똑한 VB가 오류가 없다고 판단한 이유는 코드를 짜는 사람이 너무 똑똑하다고 생각해서 이다. 무슨 말이냐고? 우리가 코드를 짜는 경우는 다양한 상황이 있을 것인데 때에 따라서 올림이나 내림, 반올림을 해야 하는 상황이 있을 수 있다. **즉, 계산한 결과를 반올림한 정수 값으로 얻고 싶을 때, 그 계산 값을 받는 변수의 타입을 정수로 정의해 놓으면 자연스럽게 반올림의 효과를 누릴 수 있다.** 똑똑한 이러한 상황을 염두하고 오류가 아니라는 판단 하에 엑셀에게 값을 도출하라고 명령을 내린 것이다. 실로 Visual Basic의 기대를 한참 못 미친 사용자의 우매함이 빛을 발하는 순간이다.

앞으로 프로그래밍을 할 때에 다른 사람들과 협력을 하여 코딩을 하는 상황이 빈번히 일어난다. 만약 [Example 3]의 코드를 다른 3자가 보았다면 명백히 오류가 있는 코드라고 판단, 수정하고 싶어할 것이다. 하지만 만약 정말 코딩을 하는 사람이 의도를 가지고 반올림을 하고 싶을 때에는 어떻게 해야 할까? 해결책은 ‘내가 정말 의도해서 한 거예요~!’ 하고 표시를 하면 된다.

[Example 4]

```
Sub test()

    Dim A as double
    Dim B as double
    Dim Result as integer

    A = 3.2
    B = 2.4
    Result = Cint(A+B)

    Sheets(1).Range("A1")(1, 1) = Result

End Sub
```

위의 예제를 실행시켜보면 A와 B에 들어있는 숫자들의 합을 반올림하여 Result라는 변수에 담아

놓았다. 따라서 우리의 엑셀은 5.6의 반올림 값인 6을 반환한다. [Example 4]의 코드는 Cint()²라는 구문을 통하여 Visual Basic이나 코딩을 같이 하는 다른 사람들에게도 데이터의 형식을 일부러 변경하였다는 명백한 의도를 보여주고 있다.

오늘의 강의는 변수의 의미와 정의하는 방법, 형식 등에 대하여 알아보았다. 가장 핵심적인 내용은 다루는 데이터와 변수의 형식이 일치하지 않았을 때에 의도치 않은 결과가 나올 수 있다는 것이다. 앞으로 우리는 보험에서 다루는 여러 상황을 VBA를 통하여 모델링 하는 일이 하게 될 것인데, 이러한 사실을 생각지 않고 코딩을 하다가는 회사의 이익과 직결되는 중대한 실수를 하게 될 수 있음을 명심하자.

[Homework]

1. 도움말을 이용하여 형식 변환 함수(Conversion function)에 대하여 알아보자.
2. 아래 Data type 표를 참고하여 각 타입의 범위를 벗어나는 정보를 넣게 되면 어떻게 되는지 확인해보자.
 <예> Integer로 정의된 변수 A에 40000을 대입하면 어떻게 될까?

[Data Type]

	Size(힘센 정도)	Description(성향)
Boolean	2	논리적인 값 True of False
Integer	2	-32768 ~ 32767 사이의 정수
Long	4	-2147483 ~ 2147483647 사이의 정수
Single	4	-3.402823E38 ~ -1.401298E-45 1.401298E-45 ~ 3.402823E38 사이의 실수
Double	8	-1.79769313486231E308 ~ -4.94065645841247E-324 4.94065645841247E-324 ~ 1.79769313486231E308 사이의 실수
Currency	8	-922337203685477.5808 ~ 922337203685477.5807
String	Length of string	약 65400자 정도의 알파벳

² Conversion to Integer를 의미한다고 생각하자. ()안의 데이터의 형식을 정수 형식으로 바꿔주는 역할을 한다.