Function	More Math Functions
Math.Abs()	Returns the absolute value. `Math.Abs(-10)` returns `10`.
Math.Ceiling()	Returns an integer that is greater than or equal to a number. `Math.Ceiling(5.333)` returns `6`.
Fix()	Returns the integer portion of a number. `Fix(5.3333)` returns `5`.
Math.Floor()	Returns an integer that is less than or equal to a number. `Fix(5.3333)` returns `5`.
Int()	Returns the integer portion of a number. `Int(5.3333)` returns `5`.
Math.Max()	Returns the larger of two numbers. `Math.Max(5,7)` returns `7`.
Math.Min()	Returns the smaller of two numbers. `Math.Min(5,7)` returns `5`.
Math.Pow()	Returns a number raised to a power. `Math.Pow(12,2)` returns `144`.
Rnd()	Returns a random number between 0 and 1. Used in conjunction with `Randomize` statement to initialize the random number generator.
Math.Round()	Rounds a number to a specified number of decimal places. Rounds up on `.5`. `Math.Round(1.1234567,5)` returns `1.12346`.
Math.Sign()	Returns the sign of a number. Returns `-1` if negative and `1` if positive. `Math.Sign(-5)` returns `-1`.
Math.Sqrt()	Returns the square root of a positive number. `Math.Sqrt(144)` returns `12`.

Alternative Exponent Notation

Using the ^ will give you the same functionality as the Pow() function.

```
Dim y As Integer
Dim x As Integer
Dim answer As Double

y = Val(InputBox("Enter Base"))
x = Val(InputBox("Enter Exponent"))

answer = y ^ x

MsgBox(answer)
```

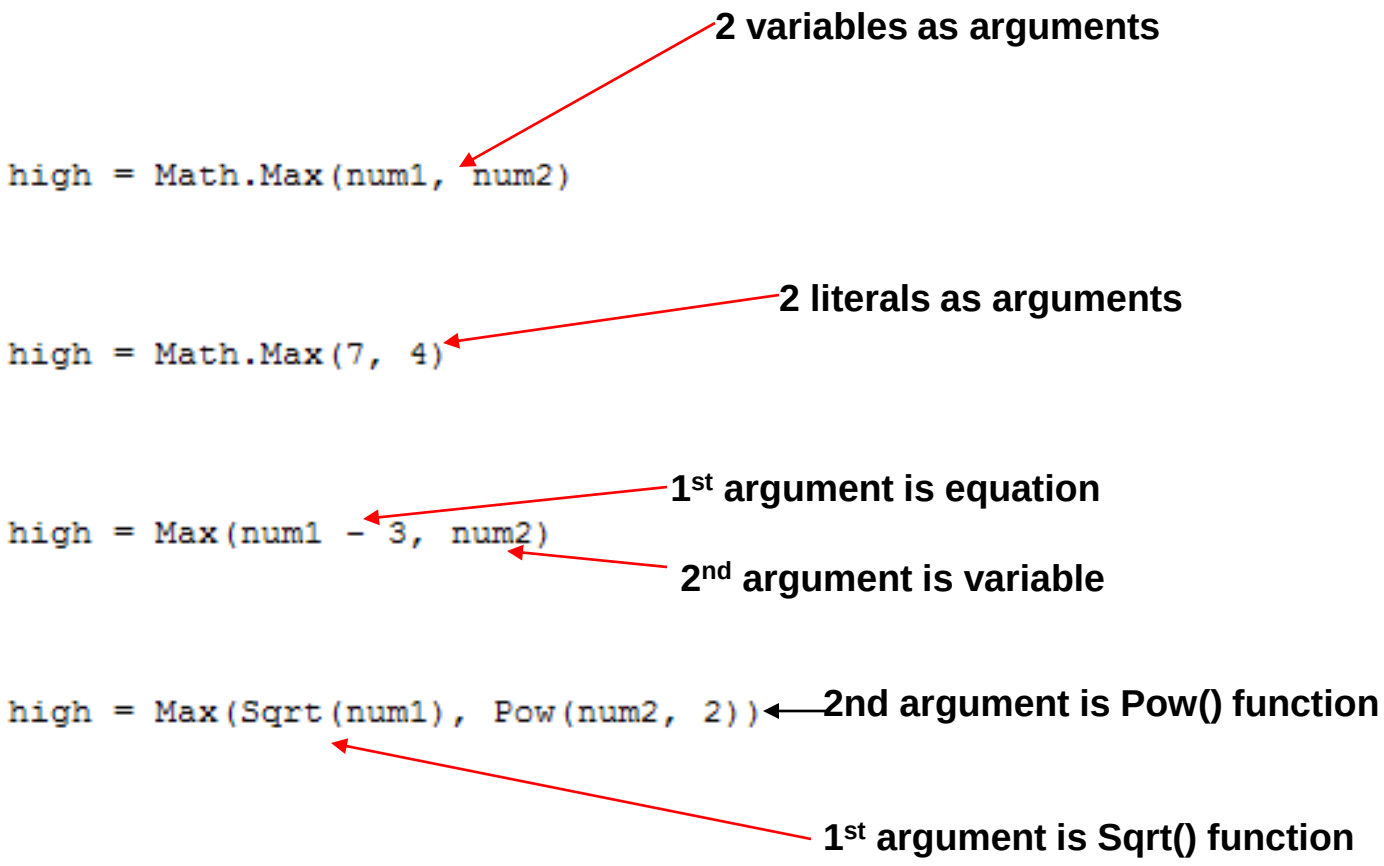
Max() Function

The Max() function takes 2 numeric values as arguments an returns the larger of the two.

```
Dim num1 As Integer
Dim num2 As Integer
Dim high As Integer

num1 = Val(InputBox("Enter 1st number"))
num2 = Val(InputBox("Enter 2nd number"))

high = Math.Max(num1, num2)

MsgBox(high)
```

The Max() Function Arguments

What makes the Max() and many other math functions so powerful is the flexibility the language permits with how the arguments are supplied. Note arguments are always separated by commas.



Breaking Down Complex Arguments

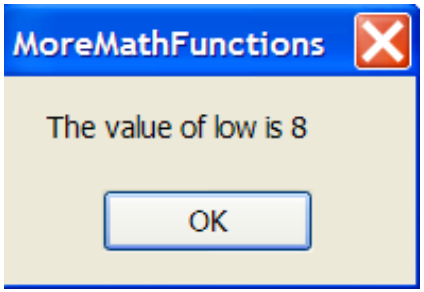
```
low = Min(Pow(2, 3), Sqrt(81))
```

value of low evaluates to 8

evaluates to 8

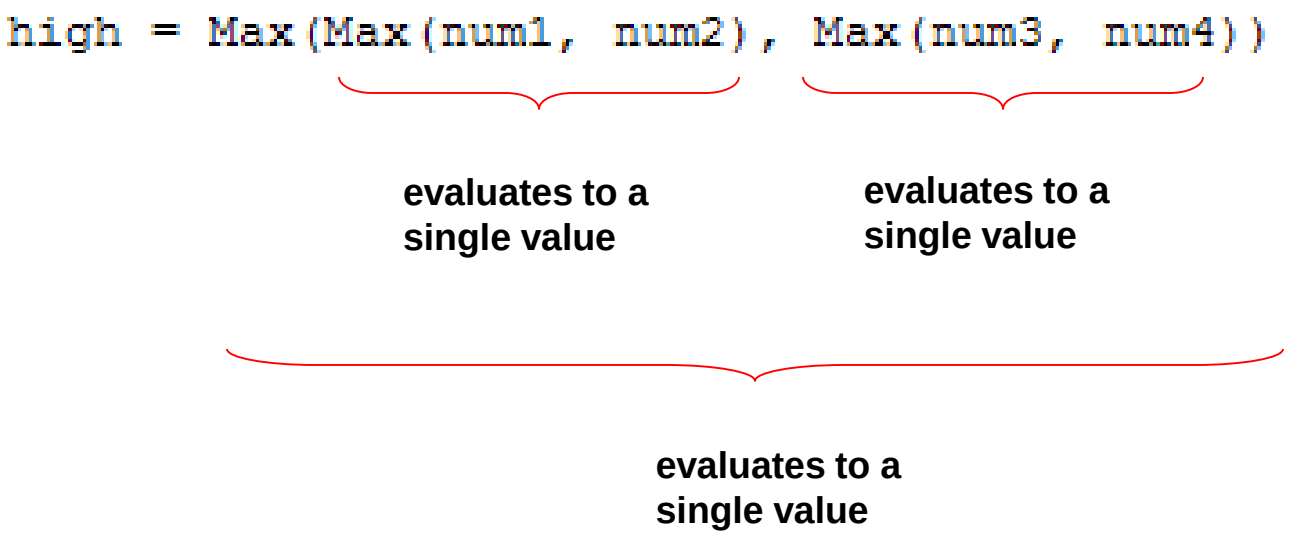
evaluates to 9

```
Dim low As Integer
low = Min(Pow(2, 3), Sqrt(81))
MsgBox("The value of low is " & low)
```



Comparing 4 Values in 1 Function

By writing functions within arguments as demonstrated in the previous slide, it is possible to determine the Max() or Min() of more than 2 values within a single line of code.



Example

```
Dim num1 As Integer = 76
Dim num2 As Integer = 156
Dim num3 As Integer = 324
Dim num4 As Integer = 257
Dim high As Integer
```

high = Max(Max(num1, num2), Max(num3, num4))

evaluates to 156

evaluates to 324

high = Max(156, 324)

high gets assigned a value of 324

So...You Think You Are Smart?

```
Dim answer As Double

answer = Max(Sqrt(Max(3.4 ^ 7, (Sqrt(2.345674)) ^ 4)), Pow(3.4, 6))

MsgBox("The answer to this ridiculous mis-use of functions is  " & answer)
```