

Microsoft Excel
IF & AND
Functions

By Stephen Groulx

Excel IF & AND Functions

Table of Contents

	Page
Introduction	3
IF Function, Simplified	6
IF Function, Nested.....	8
AND Function.....	16

Assumptions...

This paper assumes that you are familiar with Microsoft Excel, its environment and how to set up a simple worksheet.

Admittedly, the hardest part about constructing formulas is translating what you want to do, what's in your head, into the exact formula that, in this case, Excel can actually recognize.

You can pour over books, read help files until your eyes water, and try constructing formulas from trial and error. However, I'll give you the basics on some popular formulas within this document and teach you the technique for getting your ideas out of your head and into Excel.

For those who have taken any kind of programming class, you'll recognize this method. For those who haven't, this technique will be new to you. But rest-assured that it does work! All programmers (and that's what you essentially become when constructing formulas in Excel) use this technique. Master this technique, and you'll be able to construct any type of formula either perfectly the first time or with fewer errors.

In a nutshell, here's the technique:

- 1- Think of what you want to do with the worksheet, with the problem you're faced with. Use your regular thought process. For example, *"I need to test to see if this individual is a minor or an adult."*
- 2- Translate your thought into what is called pseudo-code. Pseudo-code is a mix of human language and programming language. To look at pseudo-code, you would definitely agree that it is not perfect English. Be aware, though, simply trying to use pseudo-code as the actual formula would result in numerous errors in Excel.
- 3- Write the actual formula from your pseudo-code.

Step 1: Your Regular Thought Process...

Let's assume you're building a worksheet where another person will enter information into it. One of the cells will contain a person's age. Using a formula, it is your job to determine whether or not this person is a minor (we'll assume anyone less than 18 is a minor).

As a result, the "human" thought would be something like: *"If someone enters a person's age and it's less than 18, then that person is a minor"*

That's as hard as this step gets.

Step 2: Pseudo-Code...

The second step is what is referred to as pseudo-code: Translating your human thought into an interim step to help you better understand the concept of what you will need to do. For example, to put the above into pseudo-code.

"If age is less than 18 then this person is a minor, otherwise this person is an adult."

Actual Code...

To construct the actual code, or formula, take your pseudo-code and start replacing key items with Excel's language. Here's the key items (bolded) that we'll focus on in our pseudo-code:

*"If age is less than 18 then this person is a **minor**, otherwise this person is an **adult**."*

You might also find it helpful to take your pseudo-code and further refine it. Like this:

“If age less than 18 then **minor**, otherwise **adult**.”

Now take your pseudo-code and convert it into Excel’s language.

“If age less than 18 then **minor**, otherwise **adult**.”

Will look like this in Excel:

=IF(C4<18,"minor","adult")

The information I didn’t give you before showing you the Excel example is that the place where someone would type a person’s age into is cell C4. When referencing a cell in a formula, this is referred to as a *cell reference*.

If you understood the above Excel example, try this one:

=IF(C4>18,"adult","minor")

How is this one different in construction than the first one? Does it do the same thing? *

Excel is to Numbers as Word is to Letters...

Excel is number driven, just as Word is letter driven. However, it is possible to place letters in Excel, just as it is possible to place numbers in Word. How you do this in a formula is not very obvious.

To place words (letters) in any cell in Excel, you just start typing them into a cell. To place words or letters into a formula, you must enclose them with quotation marks. However, to reference a cell in a formula (the C4 in the example above), do not use quotation marks (even though the “C” is a letter). Cell references are never (ever!) placed within quotation marks!

Hard Coding...

Try to stay away from hard coding your formulas. Hard coding is where you actually type a value (constant) into your formula. For example,

=IF(C4>18,"adult","minor")

There are three pieces of hard code above: the number 18, *adult* and *minor*.

Instead, use cell references for these pieces of information. You can do this by setting up an area of your worksheet for entering your constants.** When you then need to change a constant, you simply change your item in the constants area and all your formulas that use that item will be updated. Look at the reverse side of the coin: If you hard code your formulas, and later have to change an item, you then have to find every formula that has that item in it and change each one!

* No, it does not accomplish the same thing! If we reverse our original formula (the =IF(C4<18,"minor","adult") function, to test for an adult, we have to include the >= comparison operator. If we don’t, then anyone 18 **and** under will be considered a minor!

** Ideally, in Excel, you want to set up an entire sheet for your constants instead of using a portion of the worksheet where all your data reside.

To rewrite the previous formula using cell references, it would look like this:

`=IF(C4>B50,B52,B51)`

All the data would reside in the cells B50, B51 and B52. On your worksheet, you'd set up your constants something like:

	A	B
49	- Constants -	
50	Minor Minimum	18
51	Minor Words	Minor
52	Adult Words	Adult
53		

Again, when you need to change your criteria or other information, you simply change the data in the constants area and your formulas will automatically use the new information.

A Note About Case...

Excel allows any case letters in formulas. You can enter your formulas as all upper case letters, all lower case letters or a combination of both. It doesn't matter. Do whatever you are comfortable with.

=IF(Condition,True,False)

Use the =IF function when you need to test for a condition. For example, if a cell on a worksheet asks for a person's age, and you want to determine if he/she is a minor, use the =IF function.

Condition format is:

Item1 *Comparison_Operator* Item2

Item1 can be a cell reference or hard code

Where *Comparison_Operator* is one of the following:

=	[equals]
<	[less than]
>	[greater than]
<>	[not equal to]
<=	[less than or equals]
=>	[equals or greater than]

Item2 can be a cell reference or hard code

True

If *Condition* is true, then whatever is here in the IF function will be executed.

False

If Condition is not true (false), then whatever is here in the IF function will be executed.

Suggestions...

Read this formula like this: If condition is true then (type your first comma when you say the word "then"), type the true portion of this formula as you say "do this" otherwise (type your second comma when you say the word "otherwise") do this (and type your false portion of this formula). Remember to enter your parentheses when entering your formula!

Example:

Using our example for creating a formula to test for an adult or minor...

What you'll say to yourself

What you'll type as you say it

If this condition is True.....	=IF(C4<18
Then	,
Do this	"minor"
Otherwise (Else).....	,
Do this	"adult")

And don't forget the closing parentheses!

More Examples:

=IF(A4<=0,"Insufficient Data",A4/2)

If what is in cell A4 is zero or less, then the text “Insufficient Data” will appear in the cell where this IF function was created.

If what is in cell A4 is any number greater than zero, then that cell (A4) is divided by 2.

The following IF function is the same as the one above, but without the hard code (the zero, “Insufficient Data” and the 2):

=IF(A4<=Z100,Z101,A4/Z102)

Using cell references, instead of hard coding, is better and more flexible. It’s easier to change a constant (for example, those defined in cells Z100 through Z102) than changing multiple formulas.

When is it practical to include hard code in your formulas?

When ever it is easier!

For example, if you have a worksheet that has one simple formula and it would take ten minutes to set up a constants area to work from, it’s obviously more cost effective time-wise to simply hard code your formula.

Let’s say you have one percentage formula on a worksheet where one value (C10) is divided by .5.

=C10/.5

If this is all there is to your worksheet, hard coding your formula is fine. The greatest advantage, in this case, of soft coding (using a cell reference) would be the ability of changing your percentage figure to whatever you want very easily (as opposed to editing your formula).

=C10/E2

If our .5 resides in cell E2, simply changing the contents of E2 will update our formula.

Let's assume we want to create a worksheet that allows us to enter the type of day (Sunny, Cloudy, etc.) and the temperature outside (two pieces of information). Our formula will then look at these two cells to determine if we should go to the beach. Of course, we only want to go to the beach if it is cloudless and hot outside (our criteria), so we'll need an IF function to determine this...and a nested IF function at that!

What is a nested IF function?

It's an IF function where you have one or more IF function within it. For example:

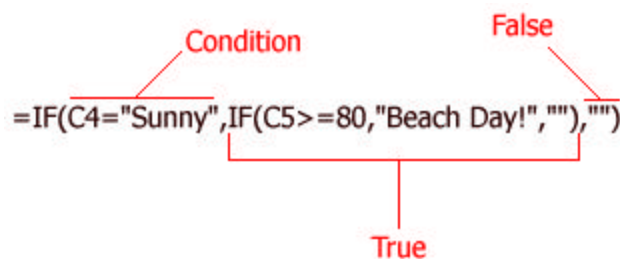
```
=IF(C4="Sunny",IF(C5>=80,"Beach Day!","", ""))
```

(see note below)

And though the above formula may look intimidating, it really isn't when you remember the basic syntax of the IF function:

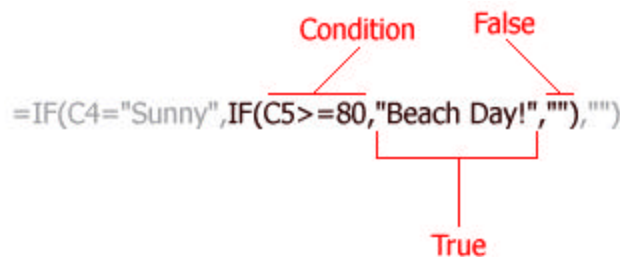
=IF(Condition,True,False)

We can break down our example as such:



The *True* portion of this nested IF function is the second IF function. But remember, even though this is another IF function, it basically fulfills the *True* portion of the starting IF function: If cell C4 contains the word "Sunny," then the True portion (the second IF function) will be executed.

When this second IF function is executed, it tests cell C5 to see if the temperature (a number) is greater than or equal to 80. If it is, then "Beach Day!" is displayed in the cell where this formula resides. Remember, even though this second IF function occupies the True portion of the original IF function, it still conforms to our syntax rule for an IF function:



Note: the closed quotation marks (`""`) represent "null" or nothing. Whenever you use two quotation marks like this, with nothing (not even a space) between them, Excel interprets them as "nothing." In other words, in the case of displaying something, if you want Excel to display nothing, you would use two quotation marks.

To put the previous formula in human terms, one would read it as such:

If the day is Sunny, then check the temperature and if this is 80 or higher, then it's a beach day, otherwise it's not a beach day.

Before we rewrite this in pseudo-code, let's identify the important pieces of our task:

If the day is Sunny, then check the **temperature** and **if this is 80 or higher**, then it's a **beach day**, otherwise **it's not** a beach day.

Rewriting this in pseudo-code:

If day is Sunny, then if temperature is equal to or greater than 80, then it's a beach day, otherwise do nothing (stay home).

And let's identify the key components of our pseudo-code:

If day is Sunny, then **if temperature is equal to or greater than 80**, then it's a **beach day**, otherwise **do nothing**.

Let's substitute certain words with Excel's language (it will still be pseudo-code, but it will be closer to what we'll need to actually type into Excel):

IF C4 = Sunny, then IF C5 >= 80, then display "Beach Day!" otherwise display nothing.

To construct this into a formula, we would enter it as:

=IF(C4="Sunny",IF(C5>=80,"Beach Day!",""), "")

You will notice that we have two nulls in use in this formula. This makes our IF statement conform to our syntax usage:

=IF(Condition,True,False)

To help you deal with nested IF functions, you may find it helpful and very useful to use what is called a *decision tree*. This is just another tool that you can use to help determine what it is you are trying to do. The more complex the problem, the greater the need for using a decision tree. I encourage you to create a decision tree so that you become comfortable working with this as an aid to creating complex formulas.

You create a decision tree simply by sketching your ideas onto a blank piece of paper (an example of a decision tree appears on the next page). When you get good at making decision trees, you'll do the majority of them in your head. The more complex ones (especially if you're going to use six nested IF functions*) will require a sketch of some sort.

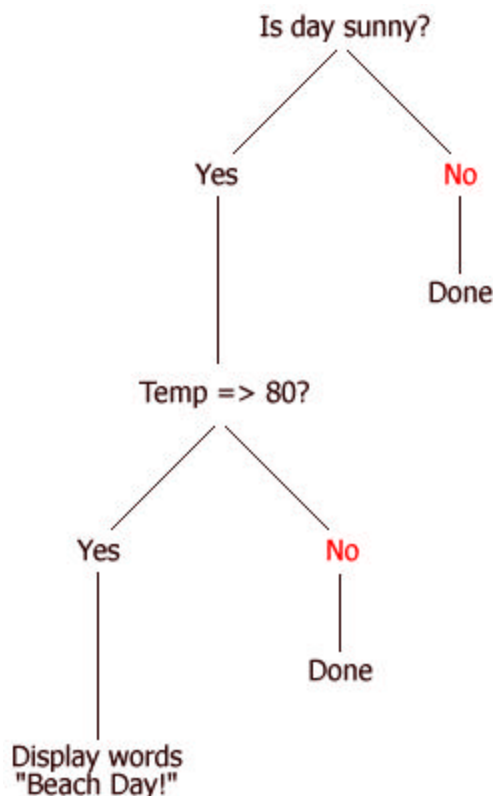
Keep your words simple. You don't have to be elaborate in creating the tree. The purpose is to get your idea down to determine if it makes sense, it flows, and it's logical. The result of doing this correctly will help you construct the actual formula.

* There is a limit of seven IF functions within a formula. A nested IF function cannot have more than seven IFs!

Here's our original task:

We want to create a worksheet that allows us to enter the type of day (Sunny, Cloudy, etc.) and the temperature outside (two pieces of information). Our formula will then look at these two cells to determine if we should go to the beach. Of course, we only want to go to the beach if it is sunny and hot outside (our criteria), so we'll need an IF function to determine this.

And here's the decision tree:



Creating a decision tree...

Start with a question (so that the answer can only be Yes or No). Here, I chose to start with “is the day sunny or not” (and I shortened it to keep it brief). If the answer is No (it's *False*), then nothing else happens; we're done.

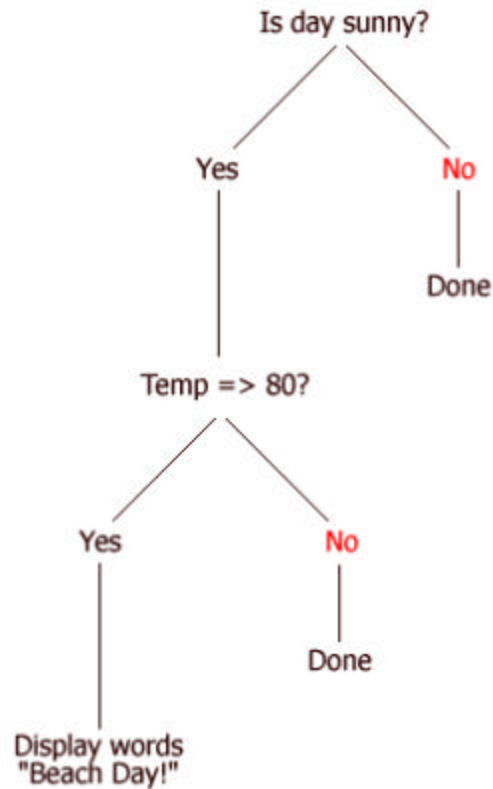
If the answer is Yes, then we follow the tree down that path to the next item. Here, another question appears: Is the temperature equal to or greater than 80 degrees? Again, if the answer is No, then nothing else happens. If it is Yes, we follow that path down to the next item which, in this case, is just to display the words “Beach Day!” (but in other complex formulas, it could lead to another IF function).

Translating the Tree into a Formula...

Now look at your decision tree you've created. Each question in the decision tree will be the word IF in our formula. We have two questions above, so we'll have two IF's within our formula.

Every “Yes” you encounter in the decision tree represents the True portion of our IF function. Every “No” you encounter will represent the False portion of our function.

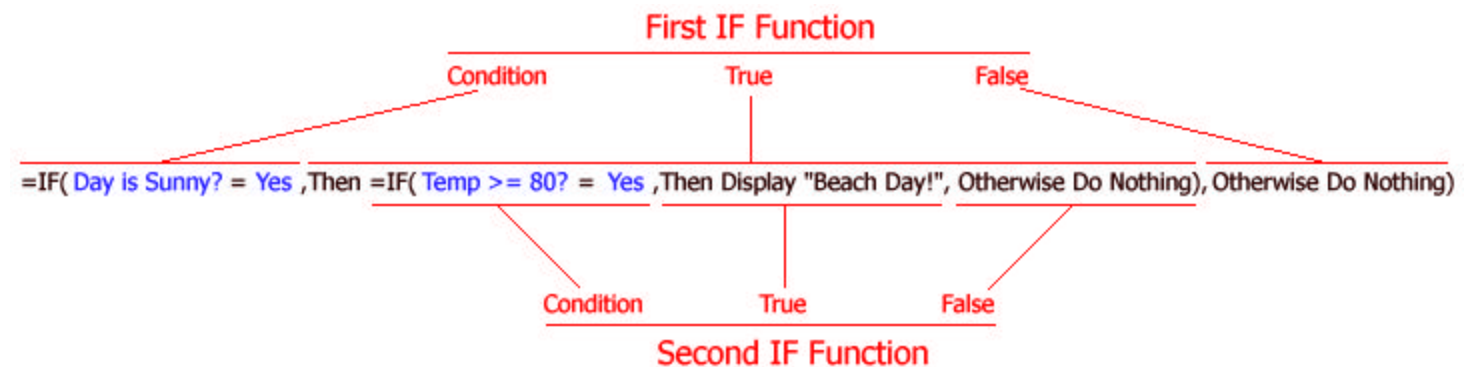
We can then take our decision tree (which is basically pseudo-code, right?) and transpose it into a sentence (which will still be pseudo-code):



=IF(Day is Sunny? = Yes ,Then =IF(Temp >= 80? = Yes ,Then Display "Beach Day!", Otherwise Do Nothing), Otherwise Do Nothing)

To ensure you have the correct format for your pseudo-formula, check it against the structure of the IF function:

=IF(Condition, True, False)



Further translate this pseudo-formula to refine its language closer to Excel's:

=IF(Day is Sunny? = Yes ,Then =IF(Temp >= 80? = Yes ,Then Display "Beach Day!", Otherwise Do Nothing), Otherwise Do Nothing)

=IF(Day = Sunny , IF(Temp >= 80 , "Beach Day!", Nothing), Nothing)

Now we're ready to translate this pseudo-formula one more time. Before we do, we'll need information on the cells where we'll be getting our data from (type of day, temperature, etc.). If our worksheet looks like the following:

	A	B	C	D	E	F	G
1							
2							
3							
4		Type of Day	Sunny				
5		Temperature	80				
6							
7							

Then our formula should look like this:

=IF(Day = Sunny , IF(Temp >= 80 , "Beach Day!", Nothing), Nothing)

 =IF(C4="Sunny", IF(C5>=80 , "Beach Day!", ""), "")

You should now be able to enter this formula into Excel (say, in cell C8), enter the word *Sunny* into cell C4, the number 80 into cell C5, and your formula should display *Beach Day!* in cell C8.

	A	B	C	D	E	F	G
1							
2							
3							
4		Type of Day	Sunny				
5		Temperature	80				
6							
7							
8			=IF(C4="Sunny",IF(C5>=80,"Beach Day!",""), "")				
9							

Now, having mentioned the value of using cell references instead of hard coding your worksheet formulas, let's set up our worksheet to use constants (albeit on the same worksheet). When we do, it should look like this:

	A	B	C	D
1				
2				
3				
4		Type of Day	Sunny	
5		Temperature	80	
6				
7				
8			Beach Day!	
9				
10				
11				
12	- Constants -			
13	Type of Day	Sunny		
14	Temperature Minimum	80		
15	Display Words (if True)	Beach Day!		
16	Display if False			
17				

And our formula in cell C8 has now been transformed from:

```
=IF(C4="Sunny",IF(C5>=80,"Beach Day!","", ""))
```

To...

```
=IF(C4=B13,IF(C5>=B14,B15,B16),B16)
```

When you need to change the criteria of your worksheet, do so in the Constants area and your formulas, all that reference the constants area, will be updated. No more formula editing!

Cosmetics...

If you change your Temperature Minimum constant to 81 on the above worksheet, you'll notice that a zero is displayed in cell C8 instead of simply nothing (null). The reason for this is if you do not put anything in a cell, Excel will interpret the cell's content as having a zero value (as opposed to having a null value).

To change this cosmetic flaw in our worksheet, simply press the space bar (enter a space) into cell B16. That will make the zero disappear.

Ideally, though, you would want to display *something* regardless what a user types into our two cells (C4 and C5). To get our formula to do this, simply enter the text you want into cell B16. For example, if it's sunny but the temperature is less than 80, we don't want to go to the beach. Our formula would not change if we are using cell references (to the Constants area of our worksheet):

```
=IF(C4=B13,IF(C5>=B14,B15,B16),B16)
```

What changes is our worksheet though, specifically our Constants area:

	A	B	C	D	E
1					
2					
3					
4		Type of Day	Sunny		
5		Temperature	79		
6					
7					
8			Not good enough, stay home		
9					
10					
11					
12	- Constants -				
13	Type of Day	Sunny			
14	Temperature Minimum	80			
15	Display Words (if True)	Beach Day!			
16	Display if False	Not good enough, stay home			
17					

But what if it's not sunny outside?

If you type anything other than sunny into cell C4, the results will be the same (the “*Not good enough, stay home*” from cell B16).

Ideally, again, we would want to plan for this possibility. To do this, we need another Constants entry.

	A	B	C	D	E	F
1						
2						
3						
4		Type of Day	Cloudy			
5		Temperature	90			
6						
7						
8			Not sunny, don't even consider the beach			
9						
10						
11						
12	- Constants -					
13	Type of Day	Sunny				
14	Temperature Minimum	80				
15	Display Words (if True)	Beach Day!				
16	Display if False	Not good enough, stay home				
17	Not Sunny Text	Not sunny, don't even consider the beach				
18						
19						

As a result, our formula will change to reflect this:

`=IF(C4=B13,IF(C5>=B14,B15,B16),B17)`

Here's a couple questions for you: Why did I alter my formula and put the reference to B17 at the end of the formula and not in place of the first B16? Why not replace both B16's with B17?

Remember how the formula was created (using pseudo-code):

```
=IF(Day = Sunny, IF(Temp >= 80, "Beach Day!", Nothing), Nothing)
```

The first test of our formula is cell C4, whether or not it's sunny. If cell C4 contains anything other than "Sunny," then the True portion (the black text above) does not get executed, only the False portion (the blue *Nothing*). In other words, the blue *Nothing* applies to the condition that is blue (*=IF(Day = Sunny)*).

Because of this, we replace our last B16 with our new reference to B17:

```
=IF(C4=B13,IF(C5>=B14,B15,B16),B16)
```

Changes to...

```
=IF(C4=B13,IF(C5>=B14,B15,B16),B17)
```

You might find it easier to revise the text that appears in cells A16 and A17 so that it is easier to read and less confusing. You can also edit the text in cell B16 to read "hot" instead of just "good." Again, this will be less confusing in the end.

11				
12	- Constants -			
13	Type of Day	Sunny		
14	Temperature Minimum	80		
15	Display Words (if True)	Beach Day!		
16	For Temp: Display if < B14	Not hot enough, stay home		
17	For Type of Day: Not Sunny Text	Not sunny, don't even consider the beach		
18				

This worksheet is basically done. There are, however, some flaws in it. For example, if you delete the text (or number) in cells C4 and C5, the worksheet will still display the text from cell B17. To get our formula to display nothing (null), we'd actually have to add a third IF function to our formula. (We won't bother doing that here, though, but you're certainly encouraged to try this on your own!)

Having read this paper to this point, try recreating these examples in Excel. You'll get a better sense of how a single change can affect one or more areas of a worksheet. And though creating cell references (rather than hard code) within formulas can be confusing, you'll see how easy it is when you simply update your Constants area with changed data.

Also, remember to sketch out a decision tree for your more complex tasks. Remember to use pseudo-code to translate your decision tree into a formula Excel can work with.

Good luck!

Let's complicate things a little more by introducing a new word: AND.

AND(*Condition1*,*Condition2*)

Use the AND function when you need to test one or more cells against two different conditions. This function, as a whole, will return True if both *Condition1* and *Condition2* are true. If *Condition1* and *Condition2* are not both True, the result for the AND function will always be False.

Let's simplify it this way:

AND(18=18,20=20)

Though that's all that is required by the AND function (two conditions), you would read it as such:

If 18 equals 18 AND 20 equals 20 then this whole thing (function) equals true.

If you typed the above function into a cell in Excel (remember to start with entering the equals sign) and pressed the Enter key, the result in that cell would read TRUE.

Try it!

=AND(18=18,20=20)

What's nice about this function is its use within an IF function (you knew that was coming, right?).

For example,

=IF(AND(18=18,20=20),"They're True","They're False")

Because 18 equals 18 and 20 equals 20 (the AND function is *True*), using this function within the IF function will allow you to do whatever you want within the true portion of the IF function. In the above example, "*They're True*" would be displayed.

Now edit your formula and change one of the 18's to 19.

=IF(AND(18=19,20=20),"They're True","They're False")

Because of your change, the AND function is no longer true and the *True* portion of the IF function will not be executed. As a result, the False portion of the IF function, the "*They're False*" entry, will be displayed.

Try both of the above examples and see what happens.

Here's a simplified example of using AND within an IF function.

Let's say Ben has two jobs, maintenance at a local school (his payroll job) and a paper route. He has made an agreement with his parents that he will save \$50 each week if his paychecks from both jobs exceed \$75. He's not worried about how much will be left for his own use, so he wants to do it this way rather than say "everything I make above \$100 will go in my savings account."

At first glance, one would tend to just look at the total that he brings in and if this total is greater than \$150 (150 = 75 + 75 from each job) take \$50 off of it for the savings account. But that would be wrong. He could make \$130 from his payroll job and \$70 from his second job (total = \$200).

But his agreement is for both jobs to pull in more than \$75 each.

This is the perfect time to create a formula using the AND function.

Here's the worksheet:

	A	B	C	D
1				
2		Payroll Income	\$ 76.00	
3		2nd Job Income	\$ 76.00	
4		Total	\$ 152.00	
5				
6		Amount to Savings	\$ 50.00	
7				
8		Balance	\$ 102.00	
9				
10				
11	- Constants -			
12		Payroll Minimum Income	75	
13		2nd Job Minimum Income	75	
14		Amount to Savings	50	
15				

And the formula would look like:

`=IF(AND(C2>C12,C3>C13),C14,0)`

We place this formula in cell C6. If the AND portion is true, then the *True* portion of this IF function goes down to C14 for the amount to be put into his savings account. If the AND portion is false, the IF function's False portion is executed, and in this case it displays the value of zero in cell C6.

The Total entry (C4) on the worksheet above is addition:

`=C2+C3`

The Balance entry is a simple math expression:

`=C4-C6`

And because we're using AND within the IF function and displaying zero if the conditions are not met, the Balance formula will always be correct:

=C4-C6

If the AND function is *True*, cell C6 will contain the value set from our constants area (50). The Balance takes the Total income (cell C4) and minuses the value in cell C6.

If the AND function is *False*, cell C6 will contain a zero. The Balance still takes the Total income (cell C4) and minuses the value in cell C6, which now happens to be zero.

Try duplicating this worksheet in Excel to see how AND is used. Remember to set up a constants area and use cell references in your formulas (don't hard code except for the zero in your formula!).

Good luck!

Glossary

Closed Quotation Marks

See Null

Data

Data are the smallest or rudimentary elements of an entity that are, by themselves, quite meaningless (compare *Information* below). For example, what do the letters E, V, S, E, T mean to you?

Nothing. They're just letters that are meaningless.

Arrange them in a certain order and put them together and you have STEVE.

Data by themselves mean nothing. They are a collection of elements that, until you do something with them, are worthless.

Decision Tree

A Decision Tree is a tool to help aid you in constructing a formula, typically an IF function (but it can have other uses).

Information

Information is a collection of data that is meaningful to someone; information tells the reader something. For example, 4TB-83D is a license plate number. But the elements (data) by themselves, the 4, T, B, -, 8, 3 and D, are meaningless until they are organized in a certain way.

Another example would be the number 112455.

That number is meaningless by itself. It could refer to miles, altitude, the last six digits of a social security number...anything!

But if I use that number, format it a certain way and place a label next to it, it then becomes meaningful: Birthday 11-24-55.

The 112455 by itself is data.

The "Birthday 11-24-55" is information.

Null ""

Null, or nothing, is specified in Excel by using closed quotation marks. Typically, you will use Null inside formulas.

Cell Reference

Instead of hard coding a cell or portions of a formula, you can reference the contents of another cell by using what is called a cell reference. For example, this is hard coded: =IF(C4>=200,C4/2.5,C4/1.5). If we had our constants (200, 2.5 and 1.5) placed in a cell elsewhere in our workbook in Excel, we could reference them by "pointing" to their cell names.

Assume 200 resides in cell B50, 2.5 resides in B51 and 1.5 resides in B52. Our formula would then look like:

=IF(C4>=B50,C4/B51,C4/B52)