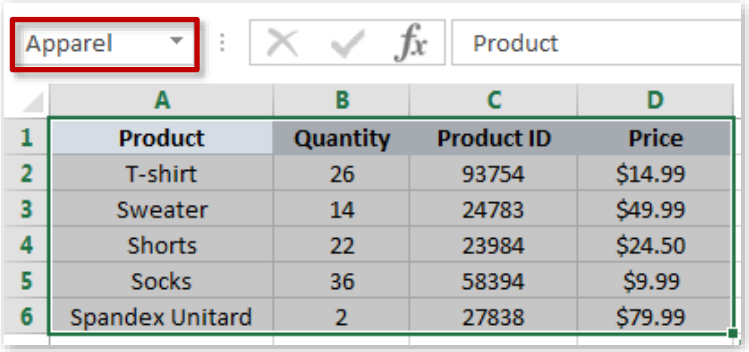
# Using Named Arrays can simplify a lookup function if you use the same data array in multiple formulas

*For example, if you name the array from A1:D6 "Apparel"…*




| | A | B | C | D |
|---|---|---|---|---|
| 1 | Product | Quantity | Product ID | Price |
| 2 | T-shirt | 26 | 93754 | $14.99 |
| 3 | Sweater | 14 | 24783 | $49.99 |
| 4 | Shorts | 22 | 23984 | $24.50 |
| 5 | Socks | 36 | 58394 | $9.99 |
| 6 | Spandex Unitard | 2 | 27838 | $79.99 |

*…you can write your vlookup formula in either of the following ways:*

**=VLOOKUP(A1,$A$1:$D$6,2)**
**=VLOOKUP(A1,Apparel,2)**

**Let's take a look at one of Excel's most common reference functions – VLOOKUP:**

**=VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])**

This is the **value** that you are trying to match in the table array

This is **where** you are looking for the lookup value

**Which column** contains the data you're looking for?

Are you trying to match the **exact** lookup value (0), or something similar (1)?

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Product | Quantity | Product ID | Price |
| 2 | T-shirt | 26 | 93754 | $14.99 |
| 3 | Sweater | 14 | 24783 | $49.99 |
| 4 | Shorts | 22 | 23984 | $24.50 |
| 5 | Socks | 36 | 58394 | $9.99 |
| 6 | Spandex Unitard | 2 | 27838 | $79.99 |

**D2=VLOOKUP(A2, $G$1:$H$5, 2, 0)**

| G | H |
|---|---|
| Product | Price |
| Shorts | $24.50 |
| Sweater | $49.99 |
| Spandex Unitard | $79.99 |
| T-shirt | $14.99 |
| Socks | $9.99 |

***To populate the Price in column D, we look up the name of the product in the data array from G1:H5 and return the value from the 2nd column over***

## Use HLOOKUP if your table array is transposed (variables headers listed in rows)

**=HLOOKUP(lookup_value, table_array, row_index_num, [range_lookup])**

This is the **value** that you are trying to match in the table array

This is **where** you are looking for the lookup value

**Which column** contains the data you're looking for?

Are you trying to match the **exact** lookup value (0), or something similar (1)?

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Product | Quantity | Product ID | Price |
| 2 | T-shirt | 26 | 93754 | $14.99 |
| 3 | Sweater | 14 | 24783 | $49.99 |
| 4 | Shorts | 22 | 23984 | $24.50 |
| 5 | Socks | 36 | 58394 | $9.99 |
| 6 | Spandex Unitard | 2 | 27838 | $79.99 |

**D2=HLOOKUP(A2, $H$1:$L$2, 2, 0)**

***With an HLOOKUP, we search for the product name in H1:L2 and return the value from the 2nd row down***

| | G | H | I | J | K | L |
|---|---|---|---|---|---|---|
| | Product | Shorts | T-shirt | Sweater | Spandex Unitard | Socks |
| | Price | $24.50 | $14.99 | $49.99 | $79.99 | $9.99 |

EXCEL MAVEN

**There are two key rules that constrain VLOOKUP and HLOOKUP formulas:**

1. **The lookup value must be in the first column of a VLOOKUP table array or the first row of a HLOOKUP table array**

2. **Excel will always return the value from the top most row or left most column of a table array when multiple instances of the lookup value are present**

**PRO TIP:**

***Avoid breaking Law #2 by identifying a "Key" that is common to both datasets and is unique for every row (NOTE: Keys often take the form of a concatenation of multiple fields)***

**The ROW function returns the row number of a given *reference*, while the ROWS function returns the number of rows in a given *array* or *array formula***

**=ROW([reference])**

**=ROWS(array)**

***This example uses an array, which is why it includes the fancy { } signs – more on that in the ARRAY functions section***

**ROW(C10) = 10**

**ROWS(A10:D15) = 6**

**ROWS({1,2,3;4,5,6}) = 2**

The **COLUMN** function returns the column number of a given *reference*, while the **COLUMNS** function returns the number of columns in a given *array* or *array formula*

=COLUMN([reference])

=COLUMNS(array)

**PRO TIP:**

***Leave the cell reference out and just write ROW() or COLUMN() to return the row or column number of the cell in which the formula is written***

COLUMN(C10) = **3**

COLUMNS(A10:D15) = **4**

COLUMNS({1,2,3;4,5,6}) = **3**

## The INDEX function returns the *value* of a specific cell within an array

**=INDEX(array, row_num, column_num)**

- array: What range of cells are you looking at?
- row_num: How many rows down is the value you want?
- column_num: How many columns over is the value you want?

| | A | B | C |
|---|---|---|---|
| 1 | Tools | Price | Inventory |
| 2 | Hammer | $5.00 | 55 |
| 3 | Screw Driver | $2.50 | 66 |
| 4 | Pliers | $3.34 | 333 |
| 5 | Wrench set | $10.00 | 234 |
| 6 | Chain Saw | $55.48 | 23 |
| 7 | Tool Box | $19.99 | 5 |
| 8 | Level | $2.25 | 7 |

**INDEX($A$1:$C$5, 5, 3) = 234**

*In this case we're telling Excel to find the value of a cell somewhere within the array of A1:C5. Starting from the upper left, we move down to the 5th row and right to the 3rd column, to return the value of 234*

# The MATCH function returns the *position* of a specific value within a column or row

**=MATCH(lookup_value, lookup_array, [match_type])**

What value are you trying to find the position of?

In which row or column are you looking? (**must be a 1-dimensional array**)

Are you looking for the exact value (0), or anything close?

***1:*** *Find largest value* **<** *or* **=** *lookup_value*

***0:*** *Find exact lookup_value*

***-1:*** *Find smallest value* **>** *or* **=** *lookup_value*

| | A | B |
|---|---|---|
| 1 | Tools | Price |
| 2 | Hammer | $5.00 |
| 3 | Screw Driver | $2.50 |
| 4 | Pliers | $3.34 |
| 5 | Wrench set | $10.00 |

**MATCH("Pliers",$A$1:$A$5, 0) = 4**

| | A | B | C |
|---|---|---|---|
| 1 | Tools | Price | Inventory |
| 2 | Hammer | $5.00 | 55 |
| 3 | Screw Driver | $2.50 | 66 |
| 4 | Pliers | $3.34 | 333 |

**MATCH(66,$A$3:$C$3, 0) = 3**

*Matching the word "**Pliers**" in column A, we find it in the **4th row**. Matching the number **66** in row 3, we find it in the **3rd column***

**INDEX and MATCH are commonly used in tandem to act like a LOOKUP function; the only difference is that INDEX/MATCH can find values in any column or row in an array**

*Example: Price Checker*

| | A | B | C | D |
|---|---|---|---|---|
| 1 | | Small | Medium | Large |
| 2 | Sweater | $10 | $12 | $15 |
| 3 | Jacket | $30 | $35 | $40 |
| 4 | Pants | $25 | $30 | $35 |
| 5 | | | | |
| 6 | *Product:* | Pants | | |
| 8 | *Size:* | Medium | | |
| 10 | **PRICE:** | ? | | |

*In this example, we want to populate the price of a given product and size in cell B10 by returning a particular value within the array B2:D4*

**B10=INDEX(B2:D4, MATCH(B6,A2:A4,0), MATCH(B8,B1:D1,0))**

*The number of rows down to index depends on what product I'm looking for, so we use a MATCH function and search for the value in cell B6 (in this case "Pants")*

*The number of columns over to index depends on what size I'm looking for, so we use a MATCH function and search for the value in cell B8 (in this case, "Medium")*

*Considering the output of each MATCH function, the formula is just a simple INDEX:*

**B10 = INDEX(B2:D4, 3, 2) = $30**

**The OFFSET function is similar to INDEX, but can return either the value of a cell within an array (like INDEX) or a specific *range* of cells**

**=OFFSET(reference, rows, columns, [height], [width])**

- **reference** — What's your starting point?
- **rows** — How many rows down should you move?
- **columns** — How many columns over should you move?
- **[height], [width]** — If you want to return a multidimensional array, how tall and wide should it be?

**An OFFSET formula where [height]=1 and [width]=1 will operate exactly like an INDEX. A more common use of OFFSET is to create dynamic arrays (like the Scroll Chart example in the appendix)**

**PRO TIP:**

***Don't use OFFSET or INDEX/MATCH when a simple VLOOKUP will do the trick***