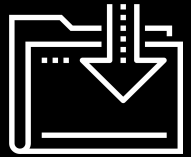


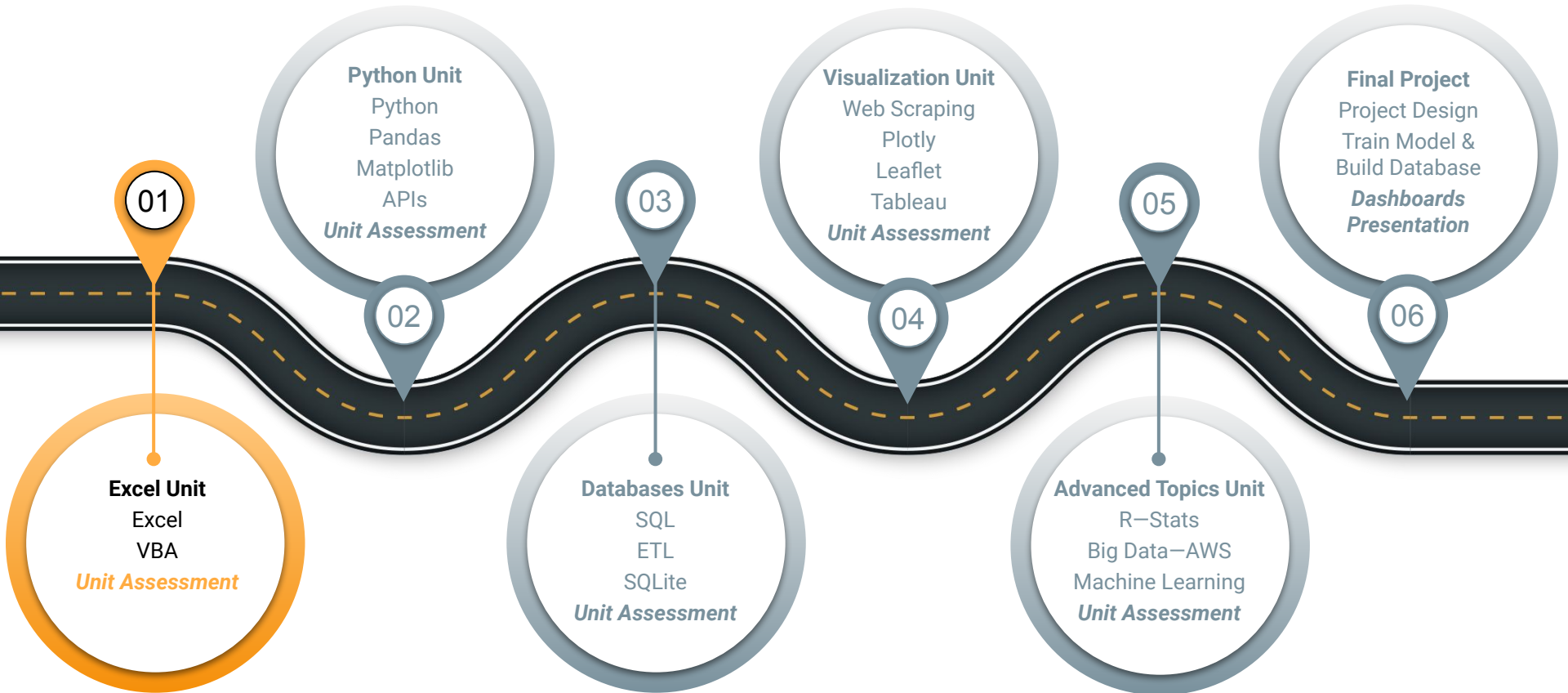


Data Boot Camp

Lesson 2.1



The Big Picture





Quick Tip for Success:

Take full advantage of office hours and your support network. Refactoring this Challenge code might be tricky! Don't be worried if you also need help with GitHub.

Module 2

This Week: VBA

This Week: VBA

By the end of this week, you'll know how to:



Create a macro that can trigger pop-ups and inputs, read and change cell values, and format cells



Use for loops and conditionals to direct logic flow



Use nested for loops



Apply coding skills such as syntax recollection, pattern recognition, problem decomposition, and debugging



This Week's Challenge

Using the skills learned throughout the week, refactor existing code to make a VBA macro run more efficiently.



Career Connection

How will you use this module's content in your career?

Module 2

How to Succeed This Week



Quick Tip for Success:

Take full advantage of office hours and your support network.
Refactoring this Challenge code might be tricky!
Don't be worried if you also need help with GitHub.

Module 2

Today's Agenda

Today's Agenda

By completing today's activities, you'll learn the following skills:

01

VBA Macros

02

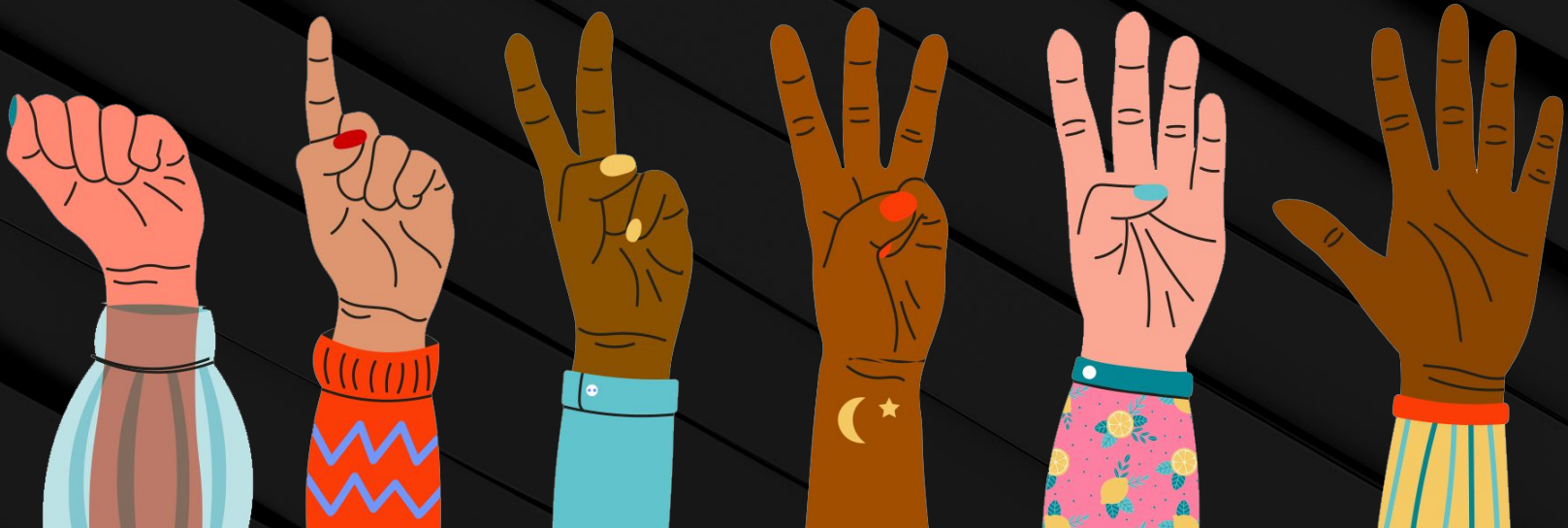
Conditionals



Make sure you've downloaded
any relevant class files!

FIST TO FIVE:

How comfortable do you feel with this topic?



Cells and Ranges

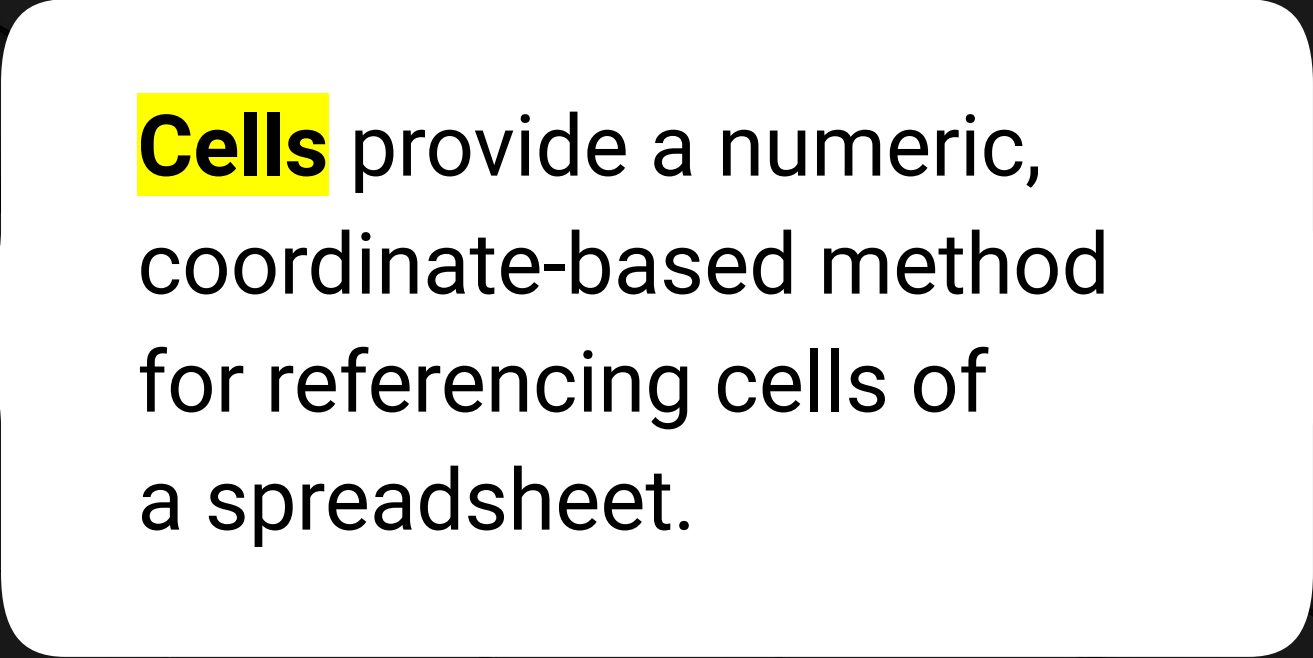


Instructor Demonstration

Cells And Ranges



VBA provides two primary ways to modify the contents of spreadsheet: cells and ranges.

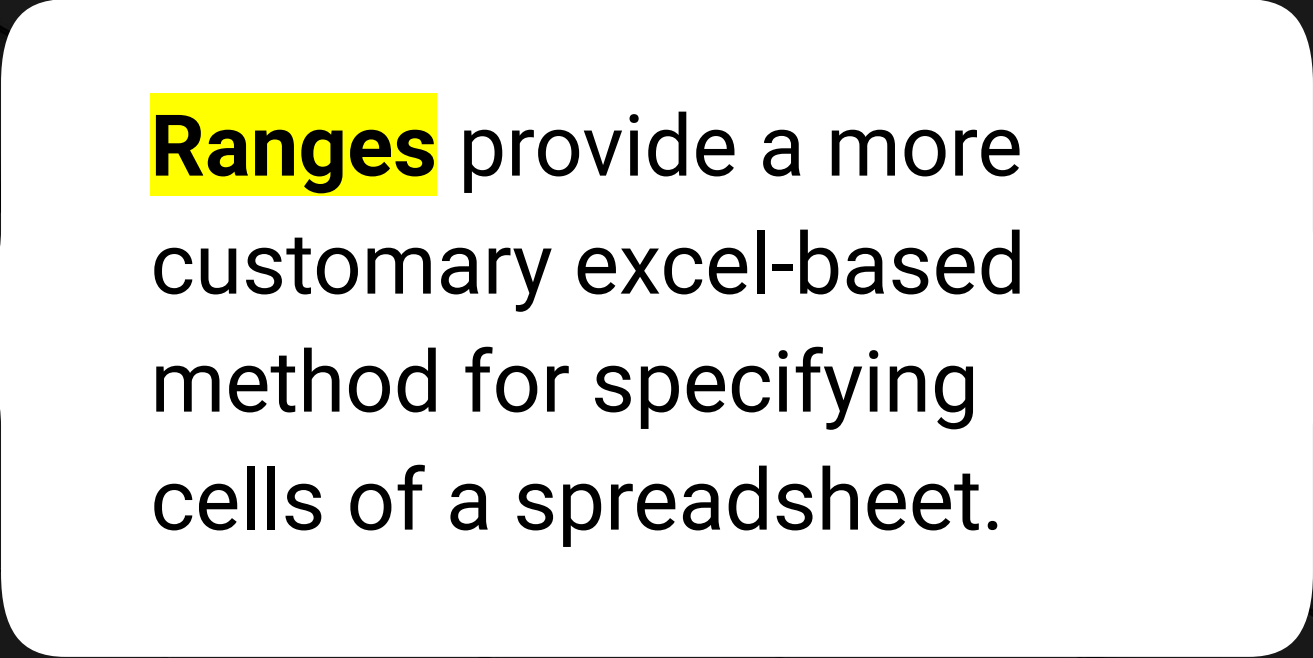


Cells provide a numeric, coordinate-based method for referencing cells of a spreadsheet.

Cells

Cells are organized in a (Row, Column) format where integers 1, 2, 3 denote columns A, B, C.

	A	B	C
1		Successful	Failed
2	Mean Goal	\$5,049	\$10,554
3	Median Goal	\$3,000	\$5,000
4			
5	Mean Pledged	\$5,602	\$559
6	Median Pledged	\$3,168	\$103

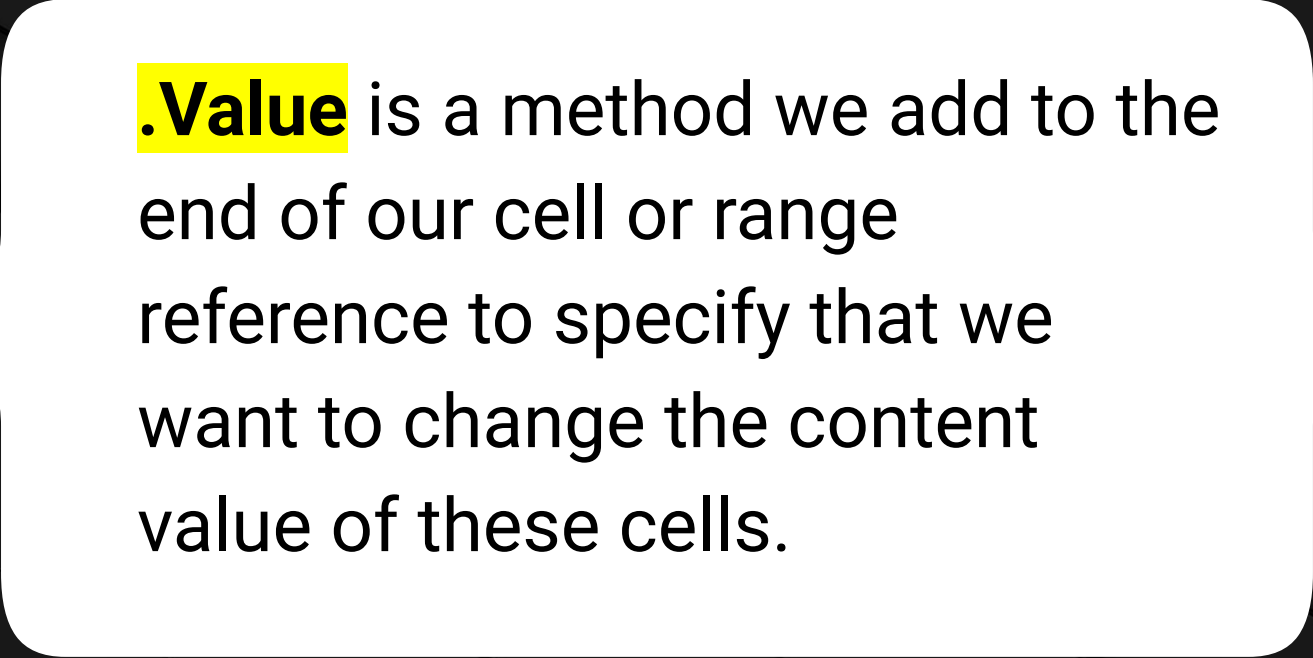


Ranges provide a more customary excel-based method for specifying cells of a spreadsheet.

Ranges

Ranges can be contiguous (e.g. "F5:F7") or non-contiguous (e.g. "R2,D2").

=MAX(C42:C57)-MIN(C42:C57)			
	A	B	C
1		Successful	Failed
2	Mean Goal	\$5,049	\$10,554
3	Median Goal	\$3,000	\$5,000
4			
5	Mean Pledged	\$5,602	\$559
6	Median Pledged	\$3,168	\$103

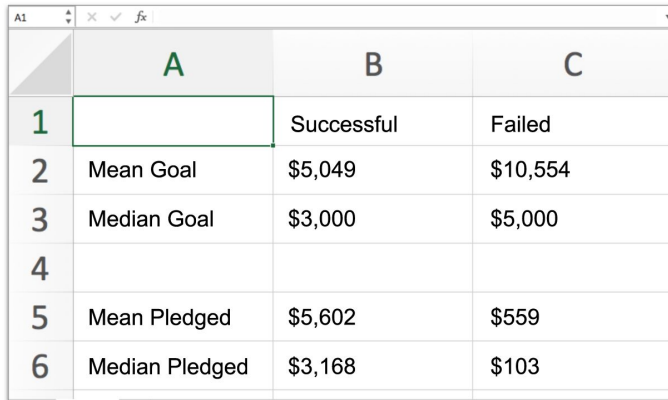


.Value is a method we add to the end of our cell or range reference to specify that we want to change the content value of these cells.

Cells vs Ranges

Cells

Allow a developer to capture a single cell at a time.



	A	B	C
1		Successful	Failed
2	Mean Goal	\$5,049	\$10,554
3	Median Goal	\$3,000	\$5,000
4			
5	Mean Pledged	\$5,602	\$559
6	Median Pledged	\$3,168	\$103

Ranges

Allow a developer to capture multiple cells at a time.

For this reason, ranges are used more often.

=MAX(C42:C57)-MIN(C42:C57)



Instructor Demonstration

Variables

Activity Workbook: Cells and Ranges

As your review the file, think about the following questions:



Where have we used this before?



How does this activity equip us for the Challenge?



What can we do if we don't completely understand this?

VBA Syntax



Variables are named
items in programming.

VBA Syntax

Variables can be **physical things** (like a name) or **abstractions** (like an age).

Variable Declaration

```
Dim name As String  
Dim age As Integer
```

VBA Syntax

In VBA, items can be **declared** as variables by using **Dim** followed by the type. We can then utilize these variables using their names by **assigning** them a value.

Variable Declaration

```
Dim name As String  
Dim title As String
```

Variable Assignment

```
name = "Gandalf"  
title = "The Great"
```

VBA Syntax

We can "concatenate" strings by combining them.

```
Dim fullname As String  
fullname = name + " " + title
```

VBA Syntax

And we can perform mathematical functions by combining numeric variables with operators.

```
Dim price As Double
Dim tax As Double
Dim total As Double
price = 19.99
tax = 0.05
total = price * (1 + tax)
```

VBA Syntax

We can also use these variables to set the value of our cells.

```
Cells(1,1).Value = price * (1 + tax)
```

VBA Syntax

We can combine numerics and strings by first "casting" our numerics into string format using the `Str()` method. And, we can cast strings into integers using the `Int()` method.

```
Dim my_age As Integer
```

```
my_age = 30
```

```
MsgBox("I am " + Str(my_age) + "years old.")
```

Questions?



Activity Workbook: Variables

As your review the file, think about the following questions:



Where have we used this before?



How does this activity equip us for the Challenge?



What can we do if we don't completely understand this?



Activity: TypeRighter

In this activity, you will need to change the data types of variables so that the code runs without errors.

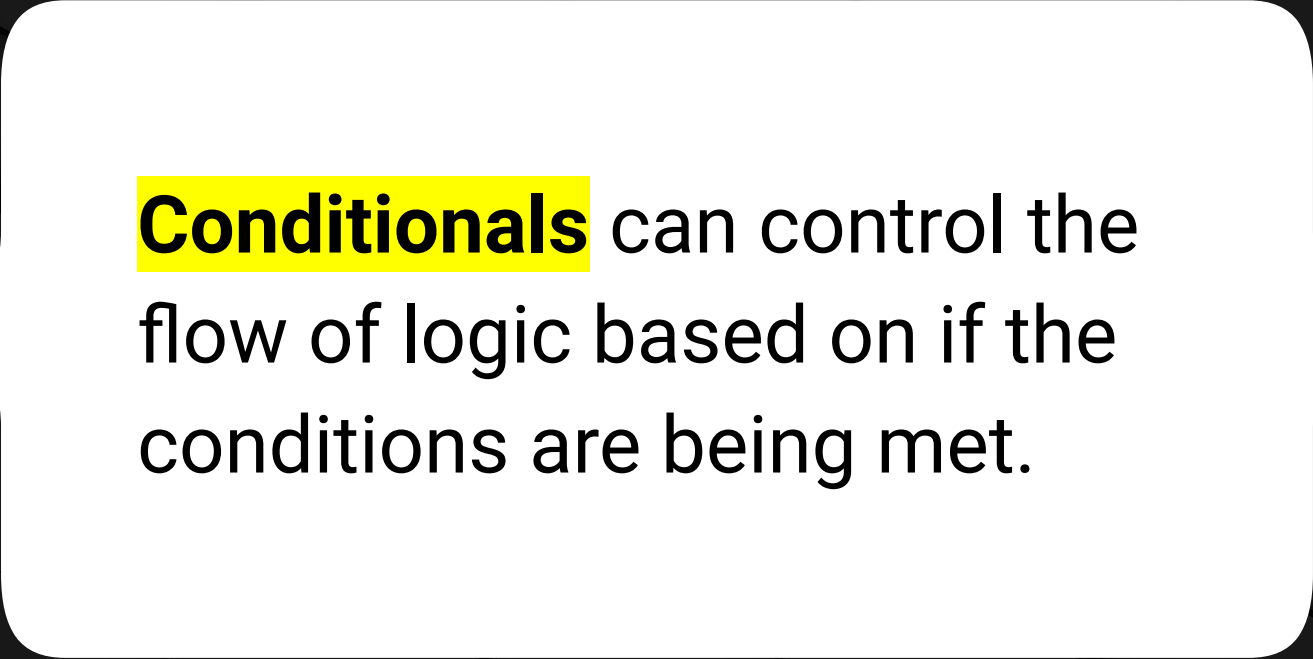
Suggested Time:

15 minutes



Time's Up! Let's Review.

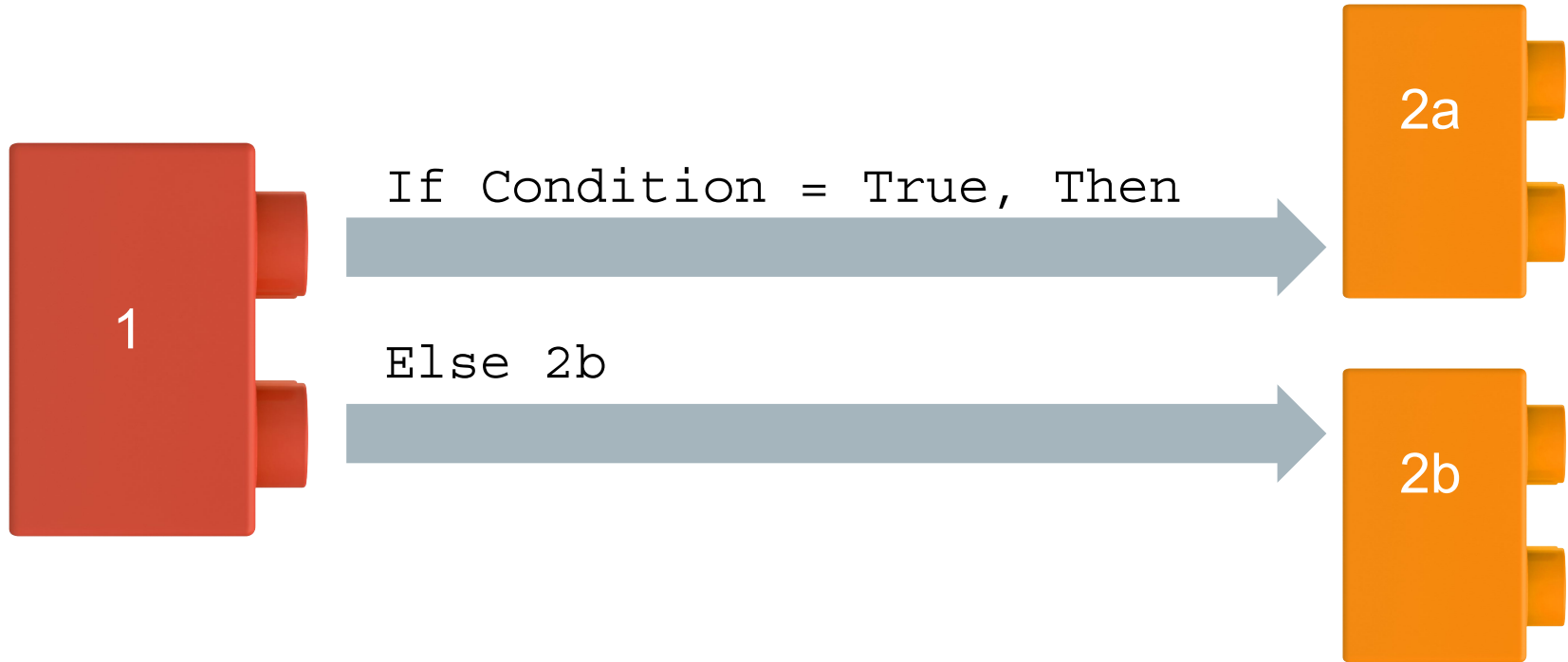
Conditionals



Conditionals can control the flow of logic based on if the conditions are being met.

Conditionals: If This, Then That

In most languages, you use if/else code for this purpose.



Simple Conditional Example

Simple Conditional Example

```
If Range("A2").Value > Range("B2").Value Then  
    MsgBox ("Num 1 is greater than Num 2")  
End If
```

If, Else, and Elseif

```
If Range("A5").Value > Range("B5").Value Then  
    MsgBox ("Num 3 is greater than Num 4")
```

```
ElseIf Range("A5").Value < Range("B5").Value Then  
    MsgBox("Num 4 is greater than Num 3")
```

```
Else  
    MsgBox("Num 3 and Num 4 are equal")
```

```
End If
```




Instructor Demonstration

Conditionals

Activity Workbook: Conditionals

As your review the file, think about the following questions:



Where have we used this before?



How does this activity equip us for the Challenge?



What can we do if we don't completely understand this?

Questions?





Activity: Choose Your Story

In this activity, work in groups to create a simple game that outputs a message box based on the user's input number.

Suggested Time:

15 minutes



Time's Up! Let's Review.

Activity Workbook: Choose Your Story

As we review, think about the following questions:

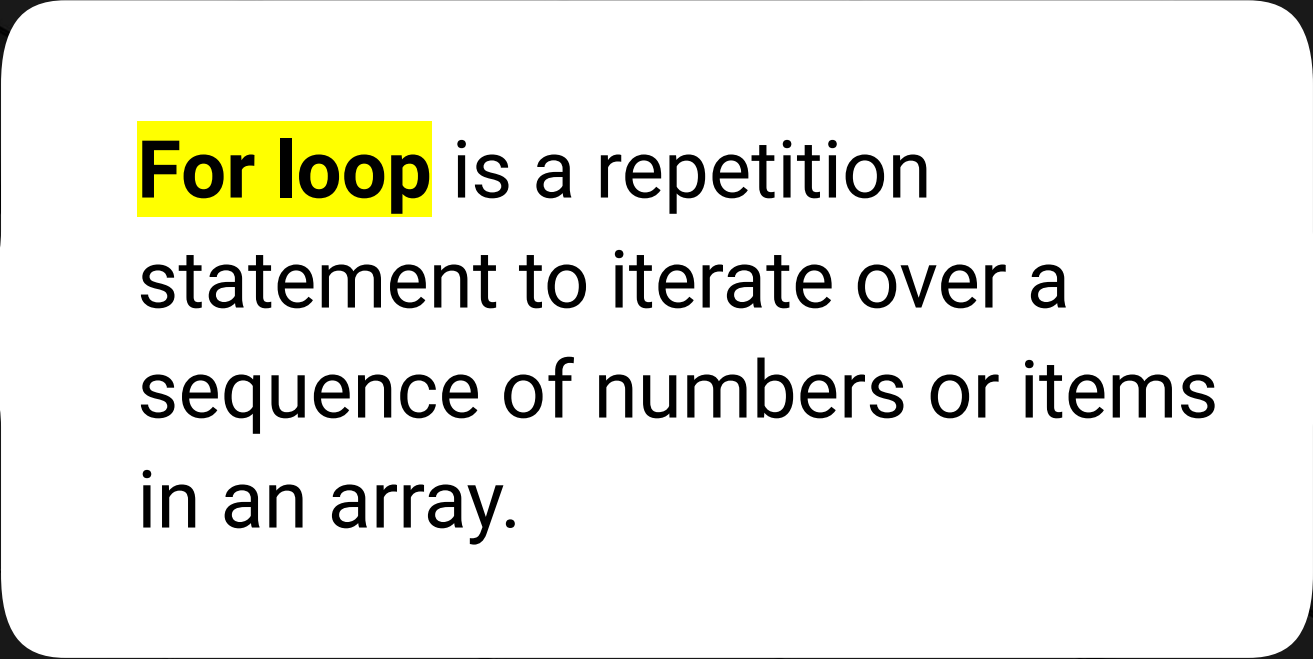


Will the program run correctly if you don't add the `.Value` at the end of the `Range()` method? Why or why not?



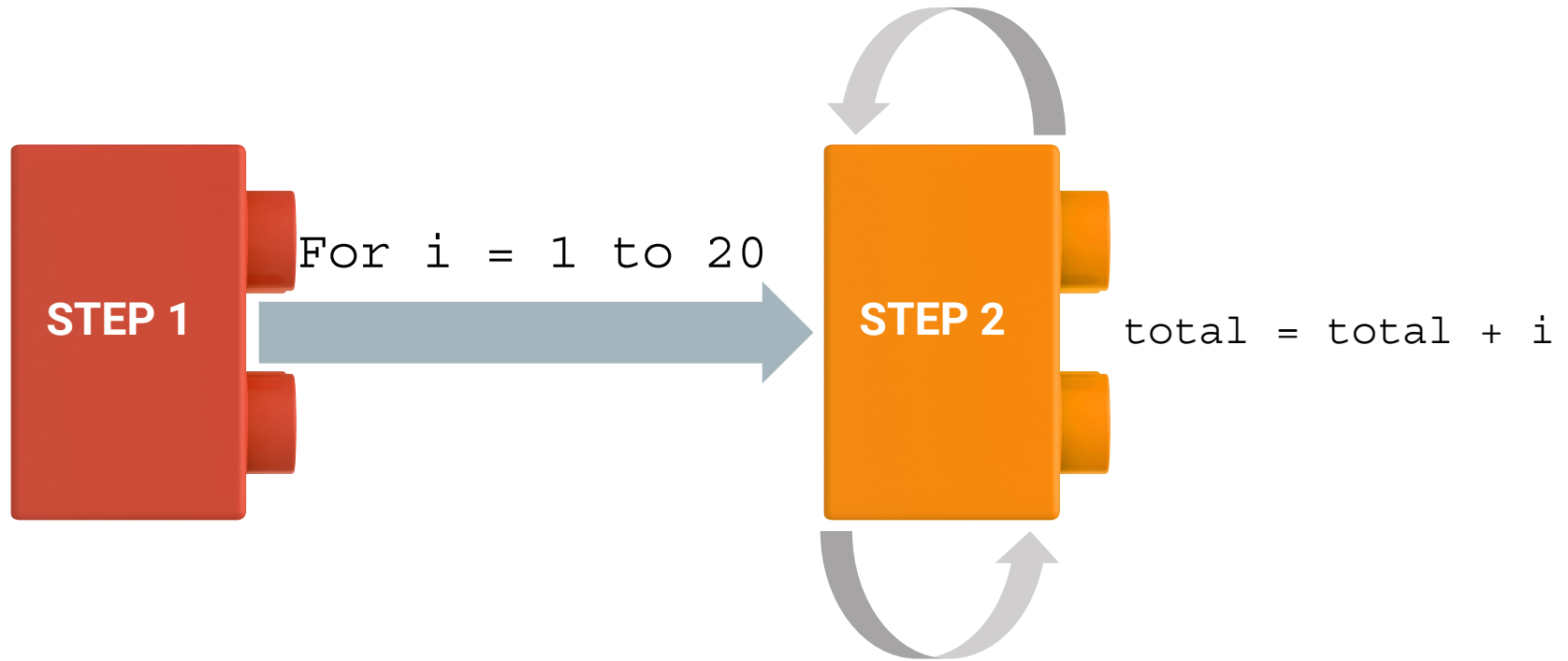
What can we do if we don't completely understand this?

For Loop



For loop is a repetition statement to iterate over a sequence of numbers or items in an array.

For Loop





Instructor Demonstration

For Loop

Activity Workbook: For Loop

As your review the file, think about the following questions:



Where have we used this before?



How does this activity equip us for the Challenge?



What can we do if we don't completely understand this?



Activity: Chicken Nugget Loop

In this activity, you will create a VBA script with a for loop that prints "I will eat "i" Chicken Nuggets," where the value of "i" changes within the for loop.

Suggested Time:

20 minutes

Questions?

