



# Fundamentals of Programming with VBA

Data Boot Camp

Lesson 2.1



# Intro to Programming Logic

# Coding (Sort Of)

---

Using Excel has introduced you to a sort of proto-programming. When writing scripts in Visual Basic for Applications (VBA), you will rely on **functions** (methods) that do something to or with **arguments**.

=

SUM(

1, 2, 3

)

Function

Arguments

Function

# Fundamental Tools of Programming

---

These structures are found in nearly all programming languages:



Conditionals



Iterations



Functions



Variables and arrays

# How a Computer Thinks (Procedurally)

Every problem in software development begins with a complex and abstract real-world need.

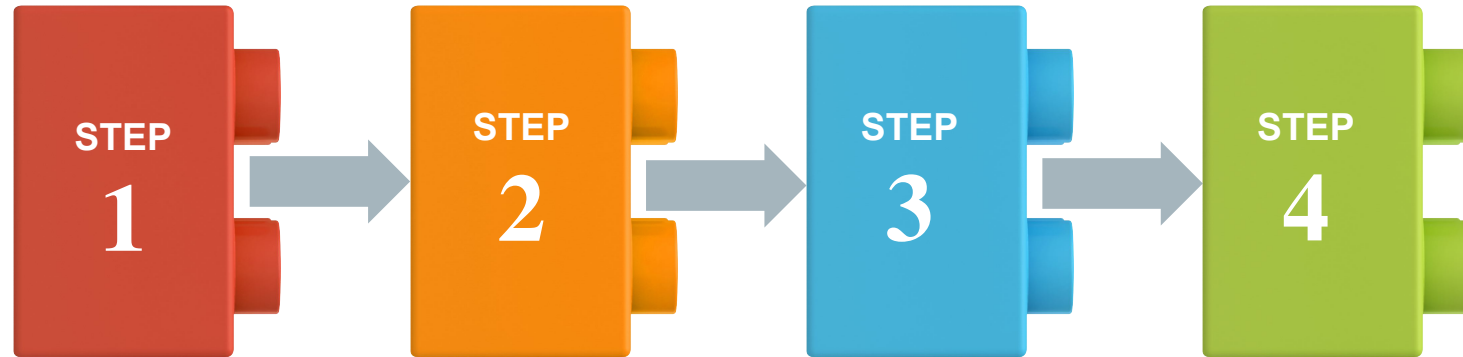


# How a Computer Thinks (Procedurally)

---

For a computer to interpret it, the real-world problem must be broken down into a set of procedural steps.

**Complex Real-World Problem**



# How Code Is Written (Procedurally)

---

## Code (Python)

```
# STEP 1
# -----
thingamagig = 500
doodad = 200

# STEP 2
# -----
combinedThing = thingamagig + doodad

# STEP 3
# -----
runContraption(combinedThing)

# STEP 4
# -----
resetContraption()
```





# When Procedures Aren't Enough, We Need More Tools

## Code (Python)

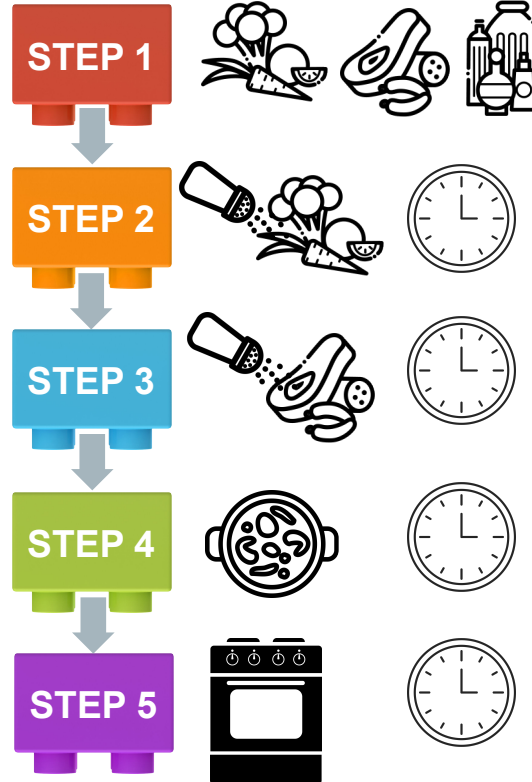
```
# STEP 1
# -----
ingredient1 = vegetables
ingredient2 = meats
ingredient3 = spices

# STEP 2
# -----
season(vegetables)

# STEP 3
# -----
season(meats)

# STEP 4
# -----
stirfry(vegetables)

# STEP 5
# -----
roast(meats)
```





# To Make Vegemite on Toast

---



# To Make Vegemite on Toast

---

Logical Procedure:

01 Get bread from toaster, and get butter and vegemite from pantry.

02 Lay out bread on table.

03 Open butter and vegemite.

04 Get spreading knife.

05 Use knife to spread butter.

06 Use knife to spread vegemite.

07 Combine bread to create sandwich.

# Fundamental Tools Can Help Make our Vegemite on Toast

---

We use these tools as building blocks to make an ideal sandwich procedure:

<b>Conditionals</b>	If the butter is salted, use less.
<b>Iterations</b>	While there is more butter, add more vegemite.
<b>Functions</b>	Spread the condiment by using a knife.
<b>Variables and Arrays</b>	The ingredients are bread, butter and vegemite.

# VBA Building Blocks



# Variables and Arrays

# Variables: The Nouns of Code



**Variables** are effectively the items in a procedure.



They can be **physical things** (like an ingredient) or **abstractions** (like a counter).



In VBA, items can be **declared** as variables by using **dim** followed by the type. They can then be **assigned** a value.

## Variable Declarations

```
dim ing1 as String
dim ing2 as String
dim budget as Double
```

## Variable Assignments

```
ing1 = "Butter"
ing2 = "Vegemite"
budget = 5.00
```

# Arrays: Collections of Items

---

An array is effectively a **group** of related items. It presents another way to store and reference similar pieces of information.

Item 0

Item 1

Item 2

["Butter",	"Vegemite",	"Bread"]
------------	-------------	----------

```
dim ingredients(0 to 2) as String
```

```
ingredients(0) = "Butter"
```

```
ingredients(1) = "Vegemite"
```

```
ingredients(2) = "Bread"
```



# Conditionals

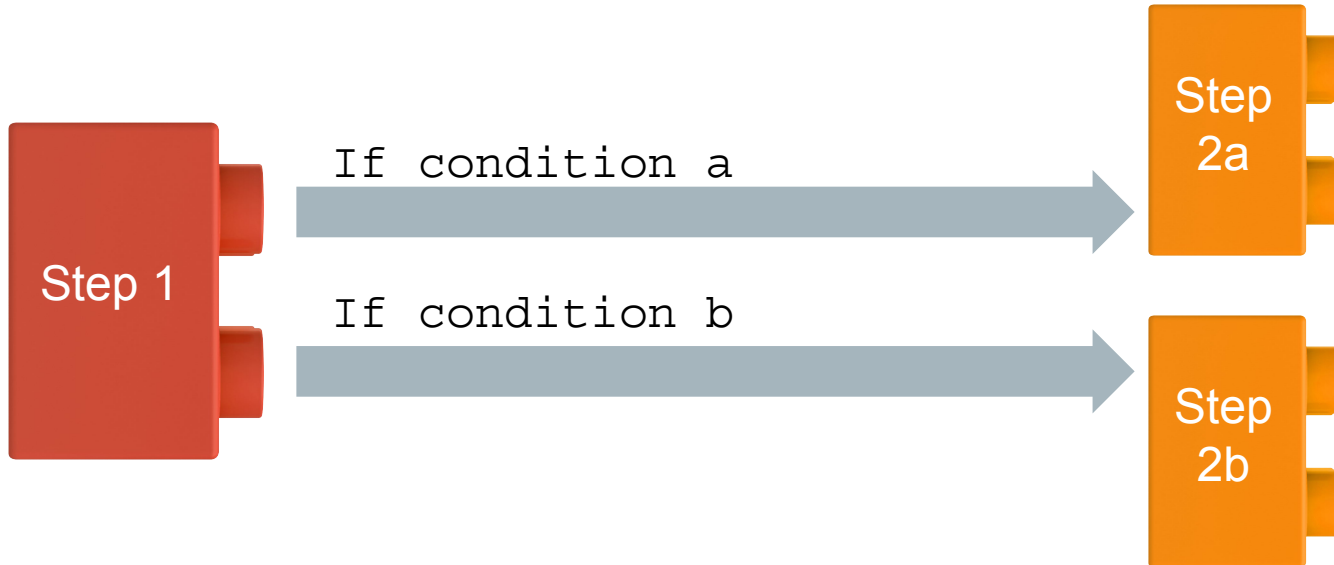
# Conditionals: If This, Then That



**Conditionals** can control the flow of logic based on certain conditions being met.



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



# Conditionals: If This, Then That

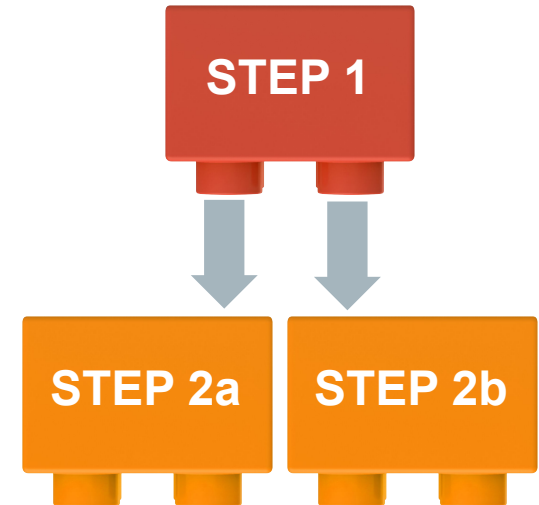


In VBA, conditionals are declared by using the keywords **If**, **Then**, **Elseif**, **Else**, and **End if**.



VBA lets us create far more sophisticated conditional logic than we can with Excel formulas alone.

```
If (pbThickness > 1.0) Then
    stopSpreading()
Else
    spreadMore()
End if
```



# Iteration (Looping)

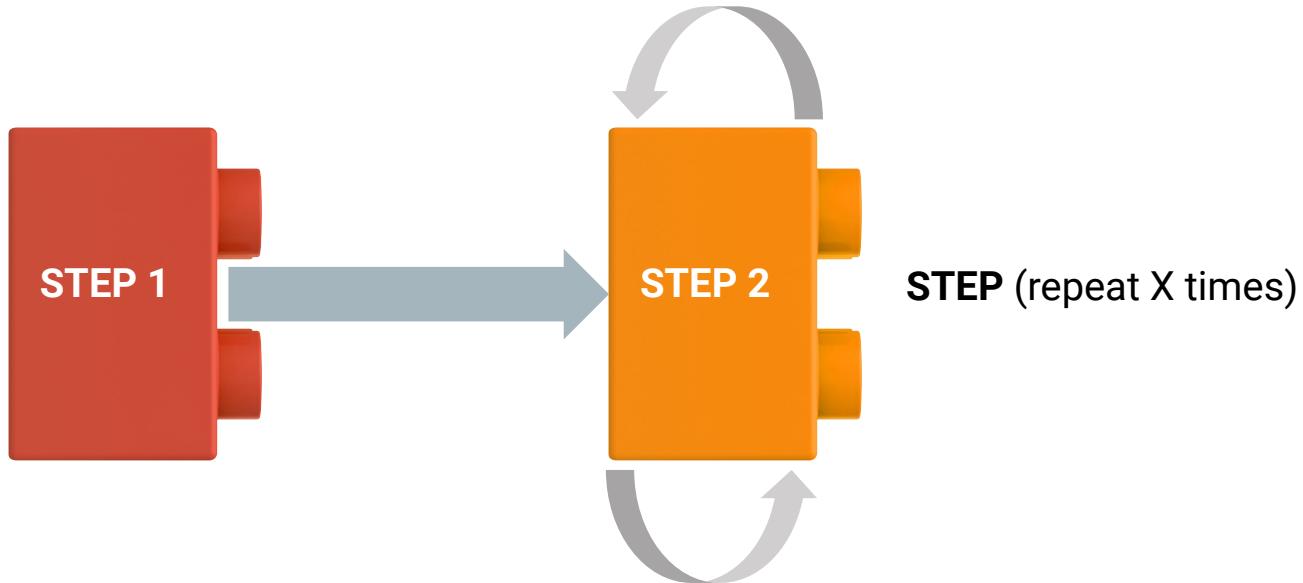
# Iteration



**Iteration** is the concept of using a loop to repeatedly perform a group of tasks a number of times.



Almost all programming languages use **for loops** and **while loops** for iteration.



# Iteration

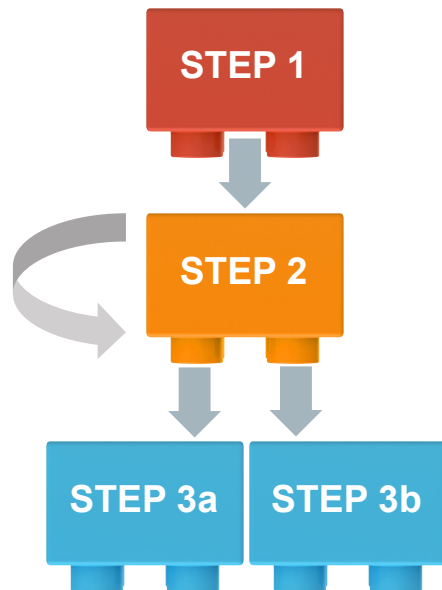
---

This code will make more sense later. Basically, it's the VBA way of repeating the same block multiple times.

```
' Repeat the same step until i becomes 20  
For i = 0 to 20  
  
    ' Each time spread more  
    spreadMore()  
  
    ' Add one to the value of i each time  
Next i
```

# Build the Program

```
1  ' Get ingredients
2  dim ing1, ing2, ing3 as String
3  ing1 = "Butter"
4  ing2 = "Vegemite"
5  ing3 = "Bread"
6
7  ' Repeat the spreading process a max of 5 times
8  for i = 1 to 5
9
10     'Each time, check that you haven't spread too much
11     if bThickness >= 1.0 then
12
13         ' If you have spread too much
14         stopSpreading()
15
16     ' Otherwise...
17     else:
18
19         ' Keep spreading.
20         spreadMore()
21     end if
22
23 next i
```



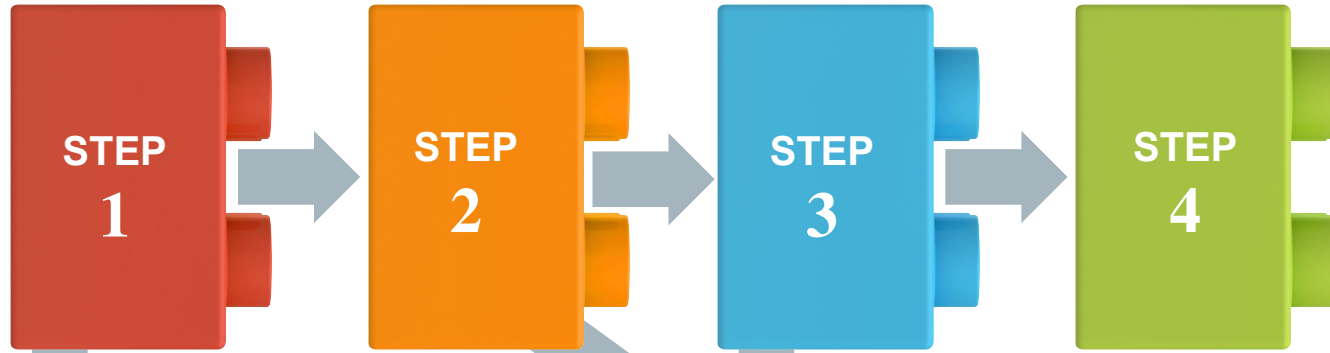


# Functions

# Functions: When One Block Can't Do It All

In essence, a **function** is a sort of subprocess. With functions, you can create premade, reusable blocks of code that can be called on demand.

**Main Process**



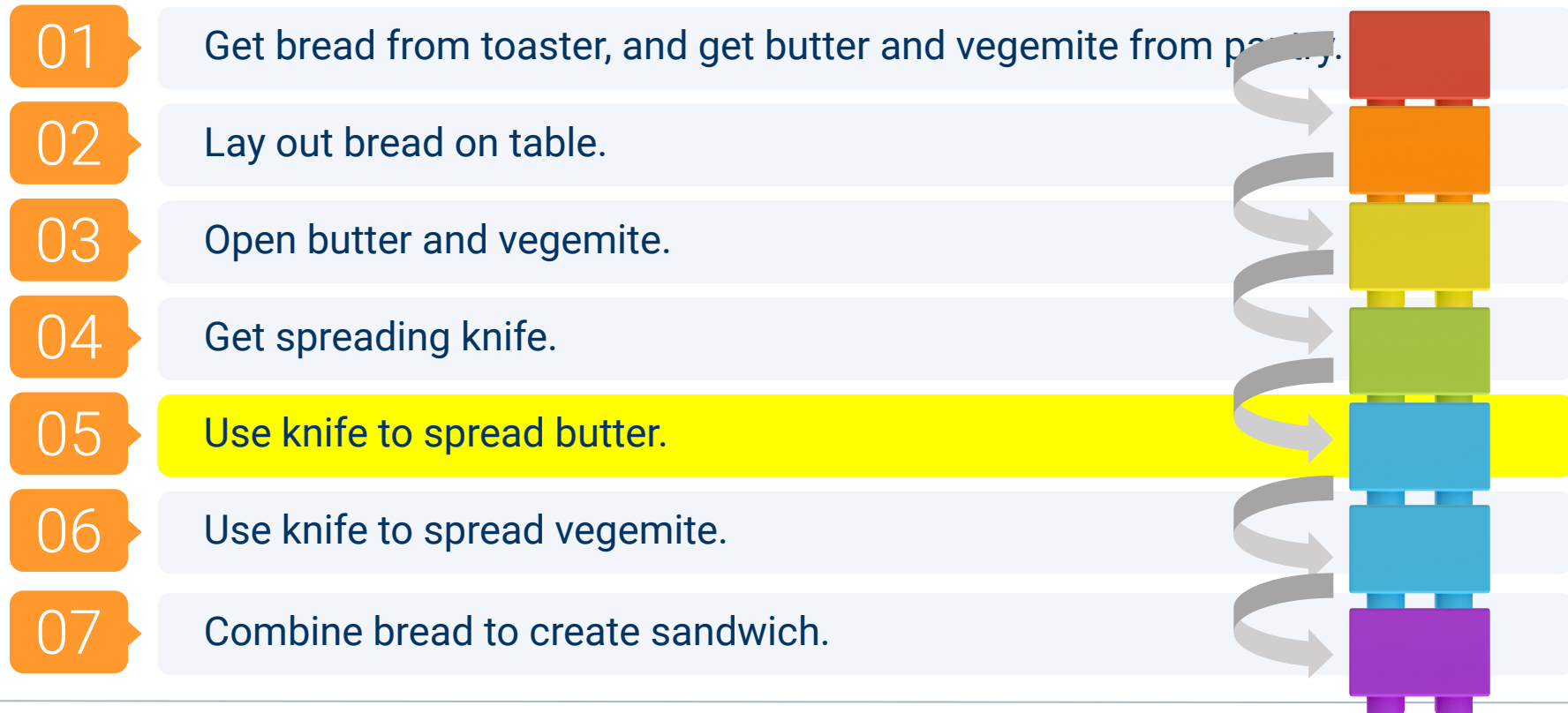
**Subprocesses**



# Putting It All Together

# To Make Vegemite on Toast

Logical Procedure:



# To Make Vegemite on Toast (Full Logic)

01

Get items.

02

Repeatedly spread the butter.

03

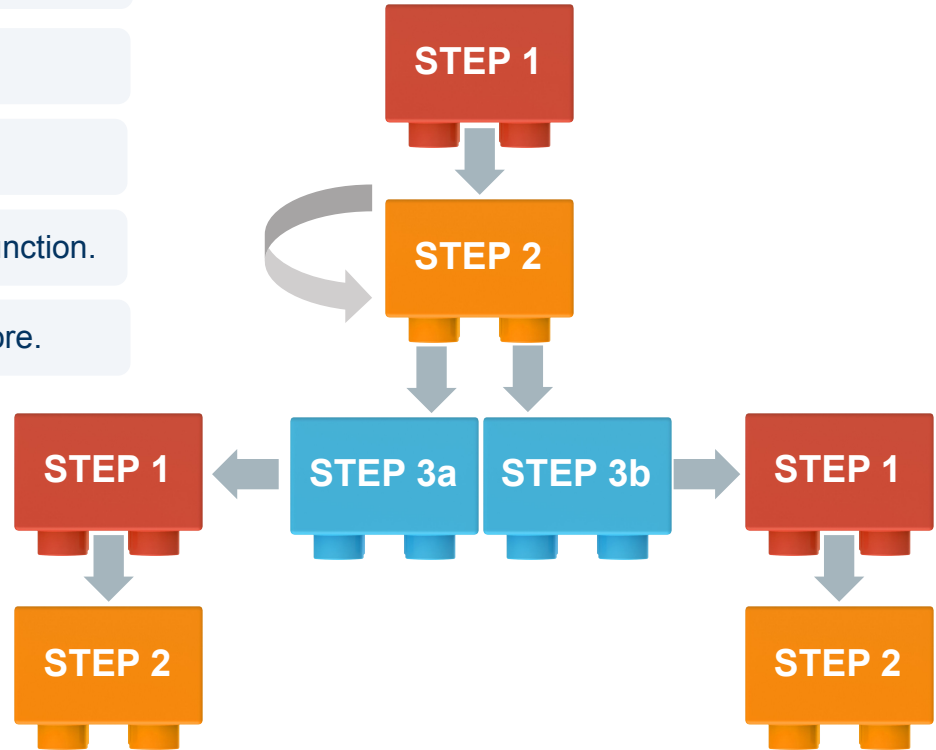
Check if the thickness condition is met.

3a

If the thickness condition is met, run the stop function.

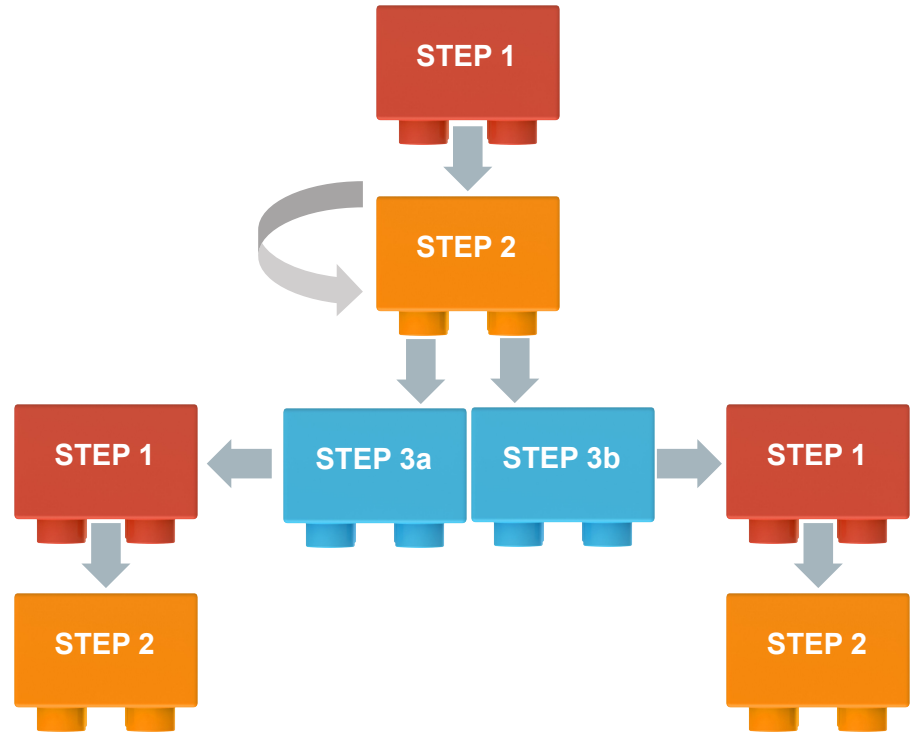
3b

If the thickness condition is not met, spread more.



# To Make Vegemite on Toast (in Code)

```
1 Sub Butter():
2
3     ' Get ingredients
4     dim ing1, ing2, ing3 as String
5     ing1 = "Butter"
6     ing2 = "Vegemite"
7     ing3 = "Bread"
8
9     ' Repeat the spreading process a max of 5 times
10    for i = 1 to 5
11
12        'Each time, check that you haven't spread too much
13        if bThickness >= 1.0 then
14
15            ' If you have spread too much
16            stopSpreading()
17
18            ' Otherwise...
19        else:
20
21            ' Keep spreading.
22            SpreadMore()
23        end if
24    next i
25
26 End Sub
27
28
29 ' Define the spreadMore function
30 Sub SpreadMore():
31
32     ' Use another set of sub-functions to move the knife
33     dipIntoButter()
34     horizontalShiftKnife()
35
36 End Sub
```



# Coding Overview

---

Coding = creating building blocks and putting them together



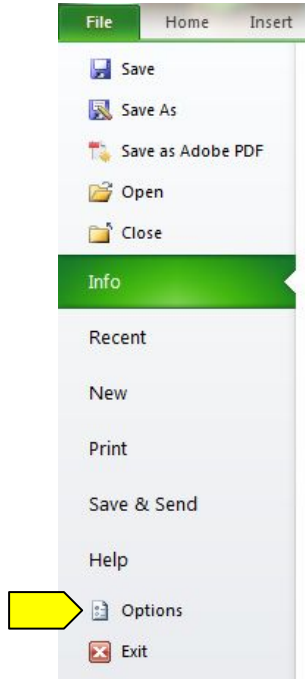


The background of the slide is dark gray with a pattern of diagonal lines that create a sense of depth and movement, resembling a series of overlapping planes or a stylized architectural design.

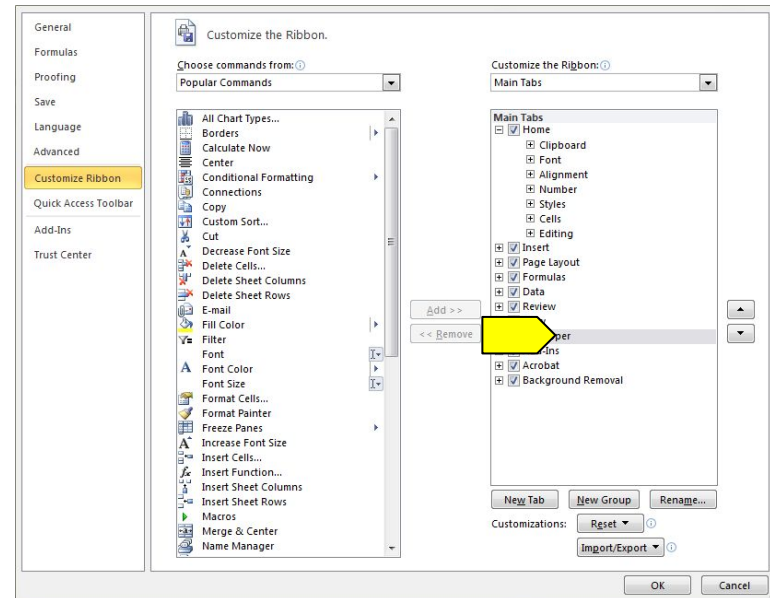
# Let's Get Coding!

# Add Developer Tools: Windows

01 Go to **File > Excel Options**.

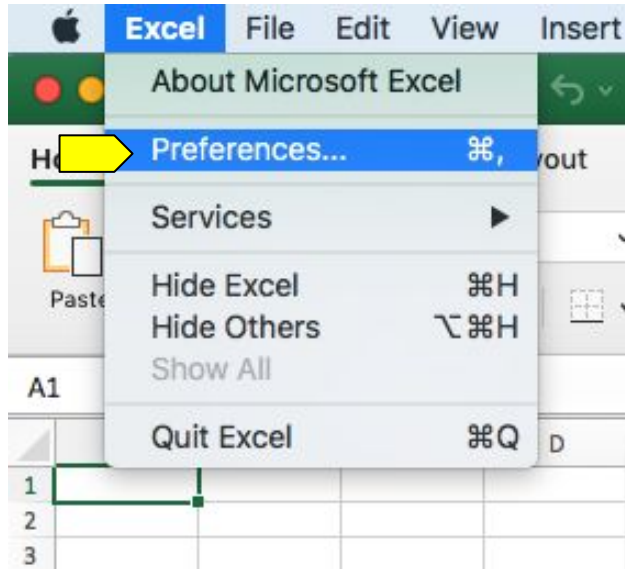


02 Go to **Customize Ribbon**, select **Main Tabs** in the right pane, and then make sure that **Developer** is checked.

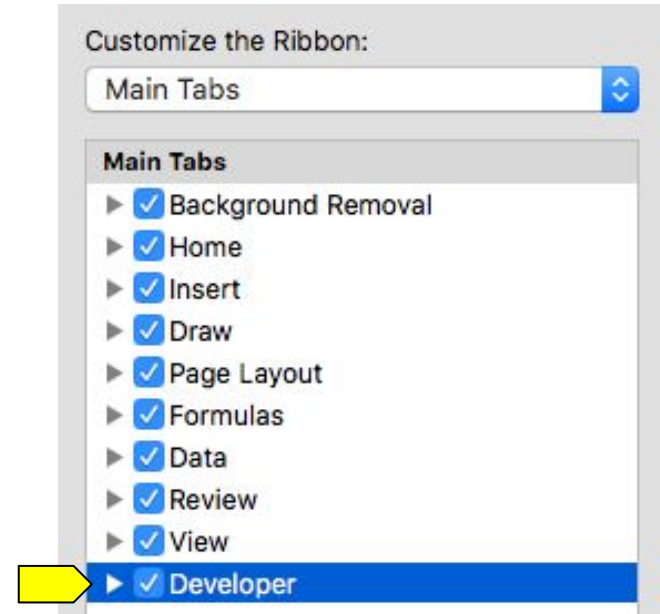


# Add Developer Tools: Mac

01 Go to **Excel > Preferences**.



02 Go to **Ribbon & Toolbar**, select **Main Tabs** in the right pane, and then make sure **Developer** is checked.



# Questions?

Questions?

