

COMP1022Q
Introduction to Computing with Excel VBA

More on Subroutines and Functions

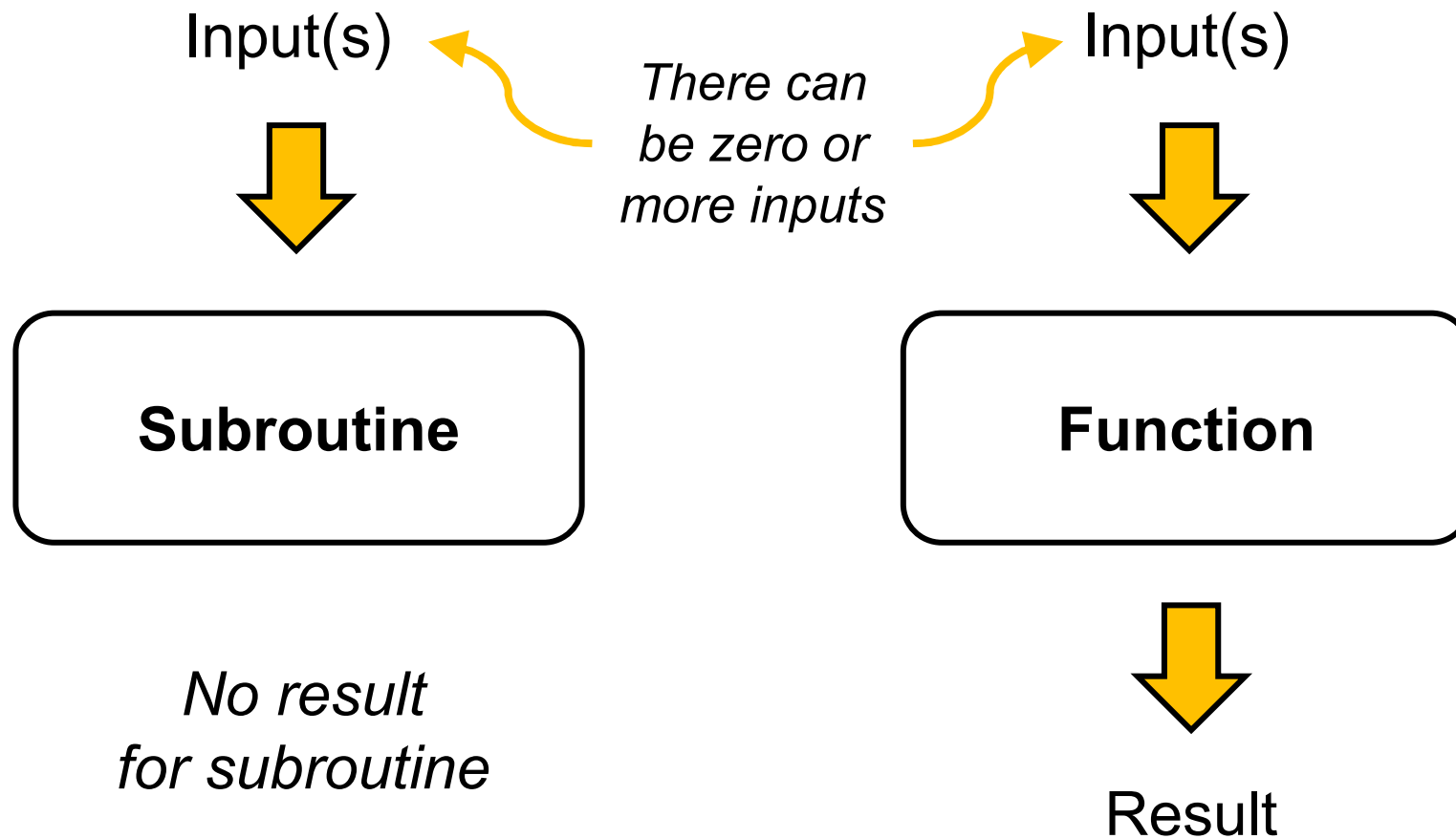
David Rossiter and Gibson Lam

Outcomes

- After completing this presentation, you are expected to be able to:
 1. Specify data types for inputs of a function or a subroutine
 2. Demonstrate the difference between a function and a subroutine
 3. Know how to finish a function or a subroutine early

VBA Subroutines and Functions

We have seen
this before

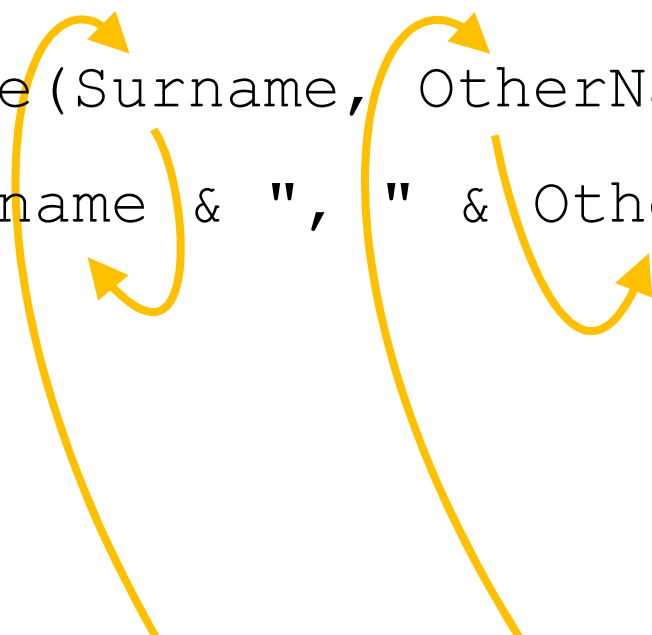


Passing More Values

We have seen
this before

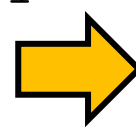
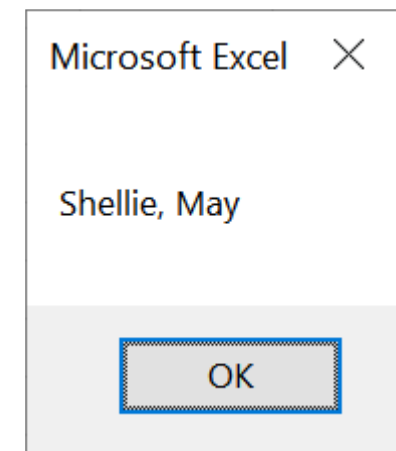
- You can have as many inputs as you want
- In this example there are 2 inputs

```
Function BuildName(Surname, OtherNames)  
    BuildName = Surname & ", " & OtherNames  
End Function
```

Two yellow curved arrows originate from the arguments "Shellie" and "May" in the function call below. One arrow points to the parameter "Surname" in the function definition, and the other points to the parameter "OtherNames".

- Using the
function:

```
Result = BuildName("Shellie", "May")  
MsgBox Result
```

A large yellow arrow points from the "MsgBox Result" line of code to the Microsoft Excel dialog box on the right.

Using Data Types

- When you make a variable, you typically tell VBA which data type you want, e.g.:

```
Dim Name As String
```

- You can also do that for the inputs of both functions and subroutines, and the output of functions
- Using data types for the inputs and outputs allow you to have better control of the values you expect to use

Specifying the Input Type

Subroutine

Function

- If you want to, you can specify the input types, like this:

```
Function SquareSize(SideLength As Integer)
```

```
    SquareSize = SideLength * SideLength
```

```
End Function
```

- Any input is automatically converted to an integer
- For example, with the above function
SquareSize(10.6) returns 121
- It doesn't return 112.36, because 10.6 is first converted to 11 and stored in SideLength inside the function

Specifying the Output Type

- You can also specify the returning result type, like this:

```
Function SquareSize(SideLength) As Integer Output
```

```
    SquareSize = SideLength * SideLength
```

```
End Function
```

- The result is automatically converted to an integer before it is returned
- For example, using the above function
`SquareSize(10.6)` returns 112, not 112.36

Zero or more
Input(s)



Output



Using Brackets

Subroutine


Function

- Brackets (also called parentheses), i.e. (), are sometimes used for functions and subroutines – when exactly do we need them?
- Here are three separate examples:

```
MsgBox "It's a sunny day!"
```

Now is a function which returns the current time

```
CurrentTime = Now
```



```
Age = InputBox("What is your age?")
```

- In the next slides, we will look at the VBA rules which say whether you need to use them or not

When You Have to Use Brackets

Function

- You need to use brackets when you run a function that has one or more input parameters **and** you are going to use the result returned by the function
 - For example:

Result = Mid ("the fat cat sat" , 4 , 3)

- Use () to enclose the parameters
- Parameters are separated using commas
- In this example, we put the result in a variable

When You Don't Have to

Subroutine

Function

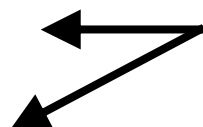
- You don't need brackets if there are no parameters

- For example, both of these work:

```
MyFirstRandomNumber = Rnd()
```

```
MySecondRandomNumber = Rnd
```

*Rnd returns a
random number
in the range
0 to 0.99999*



- A subroutine does not return anything, which means you never need to use brackets for a subroutine
 - For example, you don't need brackets when you use MsgBox:

```
MsgBox "It's a great day!"
```