Visual Basic for Applications

IIM

Visual Basic for Applications

Parameters

procedures & functions

Materials
`http://staff.uz.zgora.pl/ipajak`
`http://staff.uz.zgora.pl/gpajak`

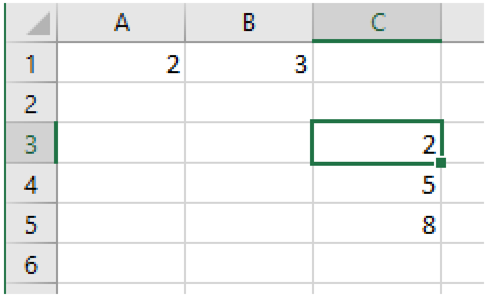
Procedures

macro Sequence1

Assumptions

macro creates creates 3 consecutive terms of an arithmetic sequence, it:

- gets initial term of the sequence from the cell A1
- gets common difference from the cell B1
- puts initial term into the active cell
- puts the second term into 1st row below the active cell
- puts the third term into 2nd row below the active cell



	A	B	C
1	2	3	
2			
3			2
4			5
5			8
6			

```
Sub Sequence1()
Dim a1 As Double
Dim d As Double

  a1 = Range("A1").Value
  d  = Range("B1").Value
  ActiveCell.Value = a1
  ActiveCell.Offset(1, 0).Value = a1 + d
  ActiveCell.Offset(2, 0).Value = a1 + 2 * d
End Sub
```

Procedures with parameters

Parameters

Sub-s can take parameters (arguments), in this case action of the sub depends upon the parameters, it makes the code more reusable.

```
   Sub name_of_sub(parameter1, parameter2, …)


   End Sub
```

When calling sub, arguments can be passed:

- positionally

  arguments in list delimited by a comma in the order that they appear in the sub definition

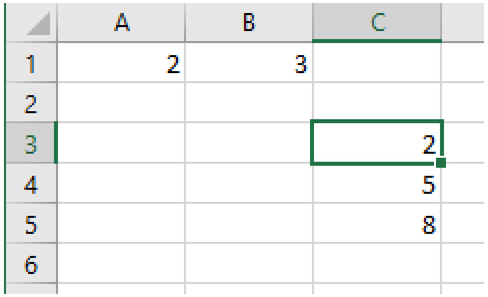
- by name

  arguments passed without regard to position

macro Sequence2

Assumptions

macro creates 3 consecutive terms of an arithmetic sequence (as macro Sequence1), but it gets initial term and common difference from input parameters



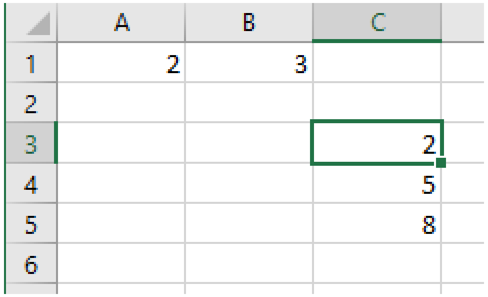
	A	B	C
1	2	3	
2			
3			2
4			5
5			8
6			

```
Sub Sequence2(a1 As Double, d As Double)
  ActiveCell.Value = a1
  ActiveCell.Offset(1, 0).Value = a1 + d
  ActiveCell.Offset(2, 0).Value = a1 + 2 * d
End Sub
```

macro Sequence2

Macro calls

- macro <u>with parameters</u> cannot be called directly,
- it is necessary to write macro without parameters that will call the first macro by passing it the appropriate parameters,
- thanks to this approach macro can be used multiple times for many different input parameters



	A	B	C
1	2	3	
2			
3			2
4			5
5			8
6			

```
Sub test1_Sequence2()
Dim a1 As Double
Dim d As Double

  a1 = Range("A1").Value
  d = Range("B1").Value
  Sequence2 a1, d
End Sub
```

```
Sub test3_Sequence2()
  Sequence2 2, 3
End Sub
```

```
Sub test4_Sequence2()
  Sequence2 a1:=2, d:=3
End Sub
```

```
Sub test2_Sequence2()
  Sequence2 Range("A1").Value, Range("B1").Value
End Sub
```

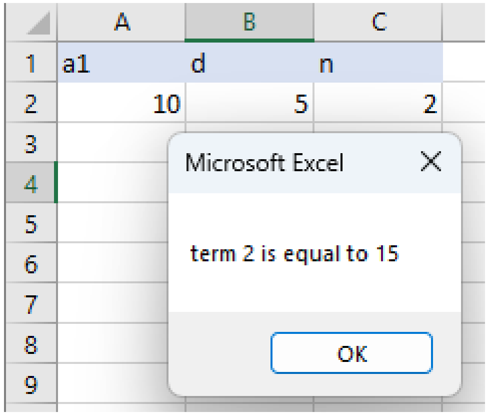
macro ArithmeticTerm

Assumptions

macro computes a single specified term of an arithmetic sequence, using four parameters:

- initial term,
- common difference,
- term number,
- calculated value of the term. ← output parameter

input parameters



	A	B	C
1	a1	d	n
2	10	5	2
3			
4			
5			
6			
7			
8			
9			

term 2 is equal to 15

```
Function ArithmeticTerm2(a1 As Double, d As Double, n As Integer) As Double)
  ArithmeticTerm2 = a1 + (n-1) * d
End Sub
```

Testing

macro test_ArithmeticTerm2 reads sequence parameters from cells A2–C2 and displays the computed term in a message box.

```
Sub test_ArithmeticTerm2()
Dim a1 As Double, d As Double, n As Integer, an As Double
  a1 = Range("A1").Value
  d = Range("B1").Value
  n = Range("C1").Value
  an = ArithmeticTerm2(a1, d, n, an)
  MsgBox "term " & n & " is equal to " & an
End Sub
```

Functions

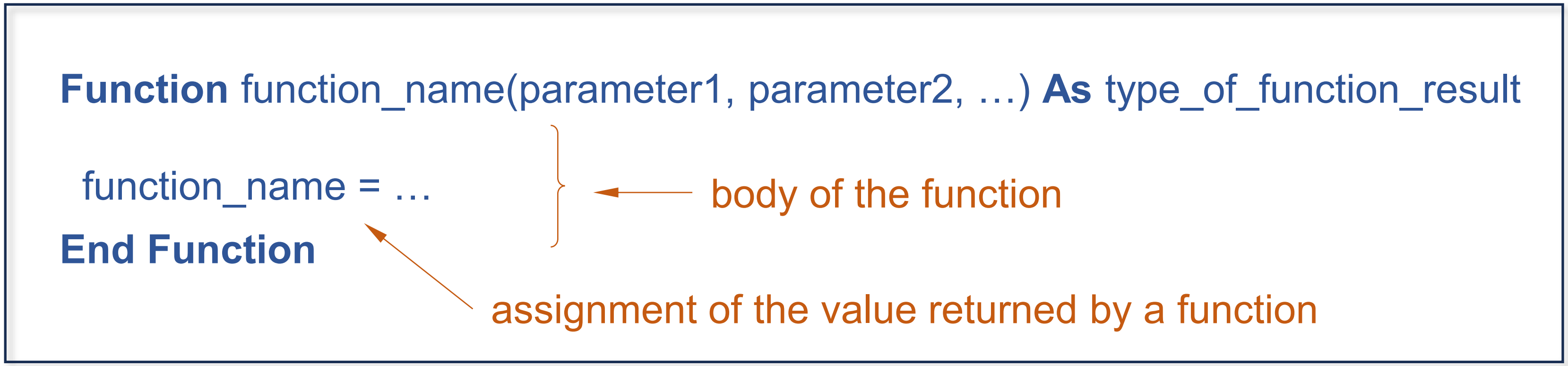
Functions

Function

is a specific type of procedure (macro) that performs a particular task by calculating a result value,

in Excel, the result returned by a function can be used:

- in a worksheet (just like built-in functions, e.g., the SUM function),
- in other macros (for performing calculations).



Function function_name(parameter1, parameter2, …) **As** type_of_function_result

function_name = …

End Function

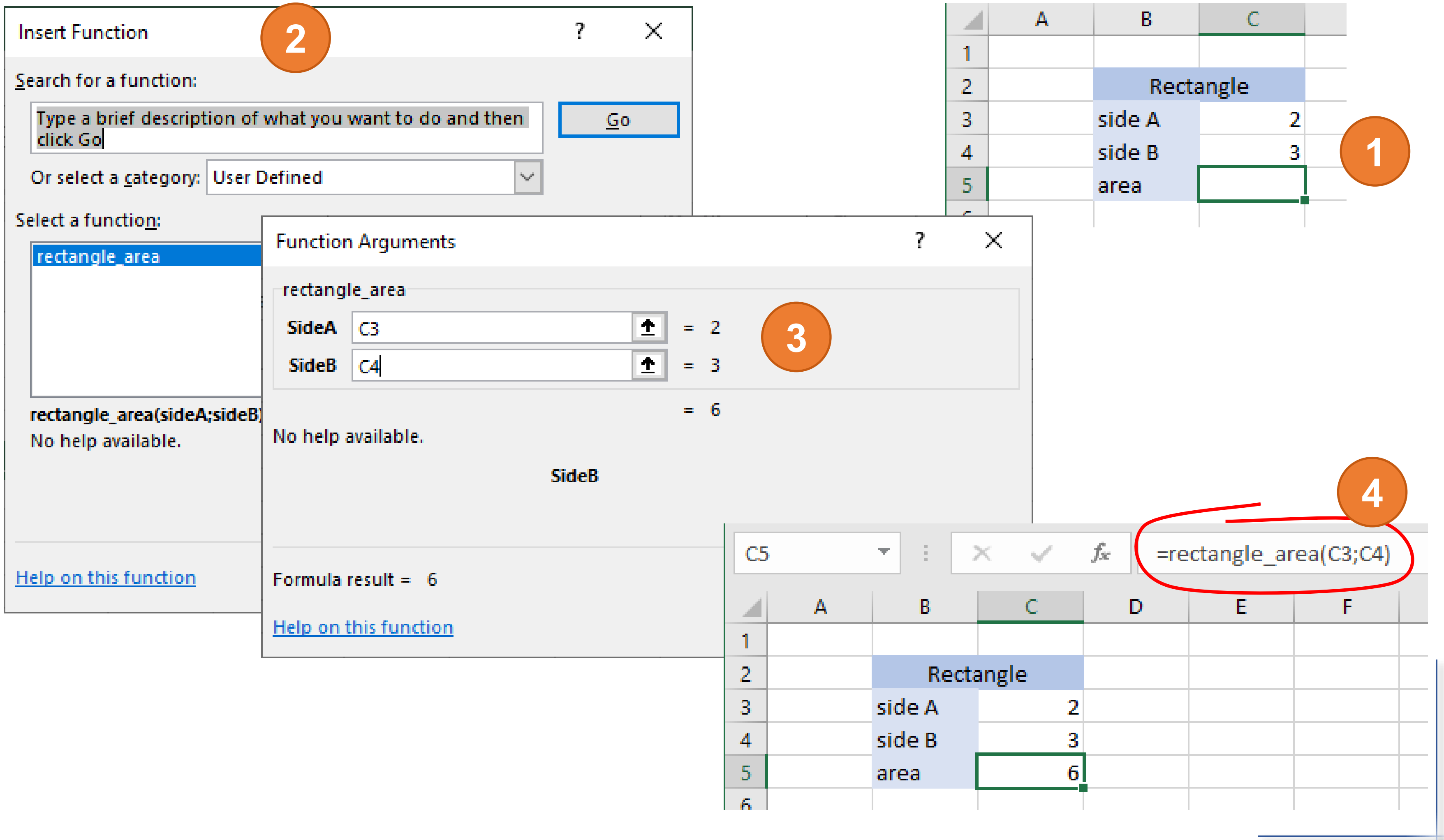
body of the function

assignment of the value returned by a function

```
Function rectangle_area(sideA As Double, sideB As Double) As Double
  rectangle_area = sideA * sideB
End Function
```

Functions – usage in a worksheet

```
Function rectangle_area(sideA As Double, sideB As Double) As Double
  rectangle_area = sideA * sideB
End Function
```



1

	A	B	C
1			
2		Rectangle	
3		side A	2
4		side B	3
5		area	

2

Insert Function

Search for a function: Type a brief description of what you want to do and then click Go

Or select a category: User Defined

Select a function: rectangle_area

rectangle_area(sideA;sideB)

No help available.

3

Function Arguments

rectangle_area

SideA: C3 = 2

SideB: C4 = 3

= 6

No help available.

SideB

Formula result = 6

4

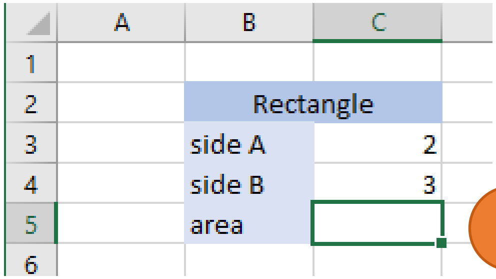
C5 | =rectangle_area(C3;C4)

	A	B	C	D	E	F
1						
2		Rectangle				
3		side A	2			
4		side B	3			
5		area	6			

Functions – usage in macros

```
Function rectangle_area(sideA As Double, sideB As Double) As Double
  rectangle_area = sideA * sideB
End Function
```

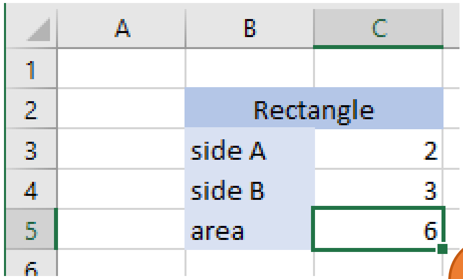
```
Sub test_area()
  Dim a As Double, b As Double
  Dim area As Double

  a = ActiveCell.Offset(-2, 0).Value
  b = ActiveCell.Offset(-1, 0).Value

  area = rectangle_area(a, b)

  ActiveCell.Value = area
End Sub
```



	A	B	C
1			
2		Rectangle	
3		side A	2
4		side B	3
5		area	
6			

1

after running the macro test_area →



	A	B	C
1			
2		Rectangle	
3		side A	2
4		side B	3
5		area	6
6			

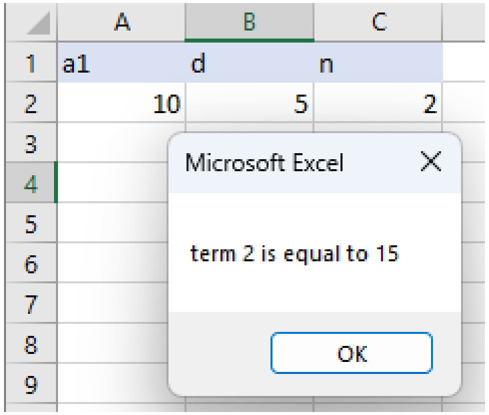
2

function ArithmeticTerm

Assumptions

function computes a single specified term of an arithmetic sequence, using four input parameters:

- initial term,
- common difference,
- term number,
- calculated value of the term.



	A	B	C
1	a1	d	n
2	10	5	2

Microsoft Excel

term 2 is equal to 15

```
Sub ArithmeticTerm(a1 As Double, d As Double, n As Integer, an As Double)
  an = a1 + (n-1) * d
End Sub
```

Testing

macro test_ArithmeticTerm reads sequence parameters from cells A2–C2 and displays the computed term in a message box.

```
Sub test_ArithmeticTerm()
Dim a1 As Double, d As Double, n As Integer, an As Double
  a1 = Range("A1").Value
  d = Range("B1").Value
  n = Range("C1").Value
  ArithmeticTerm a1, d, n, an
  MsgBox "term " & n & " is equal to " & an
End Sub
```