

- The **OR**-function can also be used to produce more complex tests. It returns the logical value TRUE if **at least one** condition in its argument is true.

Syntax: =OR(condition1,condition2,condition3,...)

Expl.: - =OR(A1>5, A2>5, B1>5, D11>5)

Returns TRUE if any of the values in A1, A2,B1,D11 is greater than 5 and otherwise FALSE.

- =IF(OR(A1<=-5, A1>5) ,0 , 1)

Produces the same function f(A1) as the example for the nested IF-function.

- The **NOT**-function reverses the values of its logical argument, i.e. TRUE is changed into FALSE and vice versa.

Syntax: =NOT(condition)

Expl.: - =IF(NOT(OR(A1<=-5, A1>5)) ,1 , 0)

Produces again the function f(A1).

35

- There are useful combinations of AND, OR and NOT:

NAND : =NOT(AND(A,B)) ≡ not both are true

NOR: =NOT(OR(A,B)) ≡ neither is true

XOR: = OR(AND(A, NOT(B)), AND(B, NOT(A)))
≡ only one is true

- The boolean values TRUE or FALSE can be entered as: TRUE, =TRUE, =TRUE() and similar for FALSE.

- Lookup & Reference Functions

Lookup functions can be used for various purposes. They can be used to retrieve information from a reference list of data and use them in some other part of the WS or WB. In general they are equivalent to some combination of multivalued IF-functions. **Reference functions** return informations about the cell reference as text values, such as the entire address, the row or column.

36

Syntax:

=VLOOKUP(lookup_value, table_array, column_index, match)

=HLOOKUP(lookup_value, table_array, row_index, match)

lookup_value ≡ The value to be located in the first column of a vertical table (or the first row of a horizontal table). It can be numeric, text or a cell reference.

table_array ≡ The range reference or name of the lookup table.

column(row)_index ≡ The column (row) of the table from which the value is to be returned.

match ≡ Is a logical value, i.e. TRUE or FALSE, which specifies whether you want an exact or approximate value. It is optional with default value TRUE. In that case the functions returns the next largest value which is less than the lookup value. For FALSE it only returns exact matches. If there is no exact match → #N/A

37

Examples: Consider the following table

	A	B	C	D	E
1	1	2	1	AA	
2	2	4	3	BB	
3	3	6	5	CC	
4	4	8	7	DD	
5	5	10	9	EE	
6	6	12	11	FF	
7	7	14	13	GG	
8	8	16	15	HH	
9	9	18	17	II	
10	10	20	19	JJ	
11					

- =VLOOKUP(6,A1:D10,2) → 12

- =VLOOKUP(4,A1:D10,3) → 7

- =VLOOKUP(8,A1:D10,4) → HH

- =VLOOKUP(3.2,A1:D10,3) → 5

- =VLOOKUP(16,A1:D10,2) → 20

- =VLOOKUP(16,A1:D10,2,FALSE) → #N/A

- =VLOOKUP(8,A1:D10,5) → #REF!

- =VLOOKUP(8,A1:E10,5) → 0

- =VLOOKUP(8,B1:D10,2) → 7

- =VLOOKUP(F1,A1:D10,2) → 6 for F1=3

→ 18 for F1=9

→ #N/A for F1≠1,...,10

38

- Improve the currency conversion table of Lab-session 1 task 2.

D	E	F
Euro	Pounds	Dollar
€ 0.10	£0.0688	\$ 0.1238
€ 0.20	£0.1377	\$ 0.2476
€ 0.30	£0.2065	\$ 0.3715
€ 0.40	£0.2754	\$ 0.4953
€ 0.50	£0.3442	\$ 0.6191
€ 0.60	£0.4130	\$ 0.7429
€ 0.70	£0.4819	\$ 0.8667
€ 0.80	£0.5507	\$ 0.9906
€ 0.90	£0.6196	\$ 1.1144
€ 1.00	£0.6884	\$ 1.2382
€ 2.00	£1.3768	\$ 2.4764
€ 3.00	£2.0652	\$ 3.7146
€ 4.00	£2.7536	\$ 4.9528
€ 5.00	£3.4420	\$ 6.1910
€ 6.00	£4.1304	\$ 7.4292
€ 7.00	£4.8188	\$ 8.6674
€ 8.00	£5.5072	\$ 9.9056
€ 9.00	£6.1956	\$ 11.1438
€ 10.00	£6.8840	\$ 12.3820
€ 20.00	£13.7680	\$ 24.7640
€ 30.00	£20.6520	\$ 37.1460
€ 40.00	£27.5360	\$ 49.5280
€ 50.00	£34.4200	\$ 61.9100
€ 60.00	£41.3040	\$ 74.2920
€ 70.00	£48.1880	\$ 86.6740
€ 80.00	£55.0720	\$ 99.0560
€ 90.00	£61.9560	\$ 111.4380
€ 100.00	£68.8400	\$ 123.8200

Into the cells I20:I22 we put:

```
0.7
=VLOOKUP(I20,$D$6:$F$33,2)
=VLOOKUP(I20,$D$6:$F$33,3)
```

Into the cells J20:J22 we put:

```
8
=VLOOKUP(J20,$D$6:$F$33,2)
=VLOOKUP(J20,$D$6:$F$33,3)
```

Into the cells K20:K22 we put:

```
50
=VLOOKUP(K20,$D$6:$F$33,2)
=VLOOKUP(K20,$D$6:$F$33,3)
```

This will produce:

€ 58.70	€ 0.70	€ 8.00	€ 50.00
£40.41	£0.48	£5.51	£34.42
\$72.68	\$0.87	\$9.91	\$61.91

39

- A geologist wants to grade some ore samples found on four different sites based on their rare metal content. Ore with a rare metal content of 50-59 ppm is given a low grade, 60-79 ppm is medium grade, 80-99 ppm is high grade and anything greater or equal 100 ppm is very high grade.

The following worksheet performs this task.

	A	B	C	D	E
1		Quality			
2	ppm	50	60	80	100
3	grade	low	medium	high	very high
4					
5	site	ppm	grade		
6	A	55	low		
7	D	111	very high		
8	C	60	medium		
9	B	77	medium		
10	A	44	#N/A		
11	B	88	high		
12	C	99	high		
13	C	56	low		
14	D	102	very high		
15					

- The lookup_values are in row B6:B14.

- The lookup_table is the range B2:E3.

- The values to be selected depending on the grade are in the column B3:E3.

- The HLOOKUP functions are in the column C6:C14.

Produce this WS in Lab-session 2.

40

- Other lookup functions are MATCH, INDEX etc.
- Use the help option to find out how reference functions work.

► **Protecting** and **hiding** worksheet informations:

- When writing workbooks or worksheets you may want to protect parts of them to make sure that your work will not be changed by accident (or deliberately). Possibly some of the informations on the WS might be confidential and should only be visible to certain users.

You set a protection by: Tools → Protection → Protect Sheet

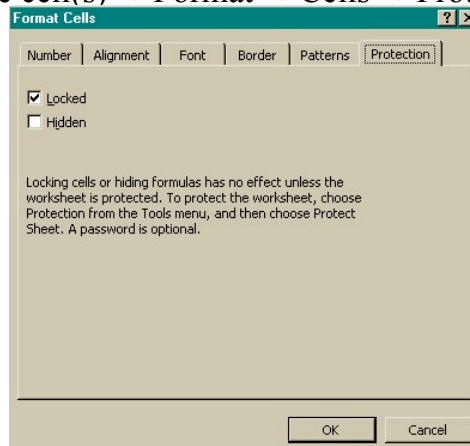
You can choose now which type of data you want to protect either contents, scenarios or objects on the WS. Optionally you can type a password, such that only with the use of this password the entire WS will be unprotected.

Unlock a protection by: Tools → Protection → Unprotect Sheet
→ Password

41

You can also just protect or hide parts of the worksheet:

Select some cell(s) → Format → Cells → Protection →



Expl.: The Excel file for the solutions of Lab-session 2 will be on the course website. The sheets are protected with “Hidden“ so that you can get an idea what the solution should look like without being able to see the formulae.

42

User-defined (Custom) Functions

► Excel is equipped with the powerful programming language **Visual Basic for Applications** (VBA). VBA allows you to write your own programs, such as **user-defined functions** (UDF) and **subroutines** (see later in the course).

- What is a UDF?

Just like a built-in function, a UDF is a pre-defined formula which can be executed in the same way. The difference is that you design the definition exploiting the flexibility of VBA.

- When and why do you use a UDF?

You use a UDF for the same reason as a built in function, namely to make calculations (operations) which are repeated more efficient.

- Before writing a UDF make sure that it or parts of it do not already exist as built-in Excel functions.

43

► Creating a UDF

• First activate the **Visual Basic Editor** (VBE)

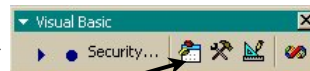
- Tools → Macro → Visual Basic Editor (LC)

- or use the keyboard shortcut Alt + F11

- When you program it is useful to include the Visual Basic toolbar into your menu:

View → Toolbars → Visual Basic (LC) →

Now you can also activate the VBE by LC on



• The anatomy of the VBE is like most other applications. It is equipped with a menu and a toolbar at the top of the window and has several subwindows:

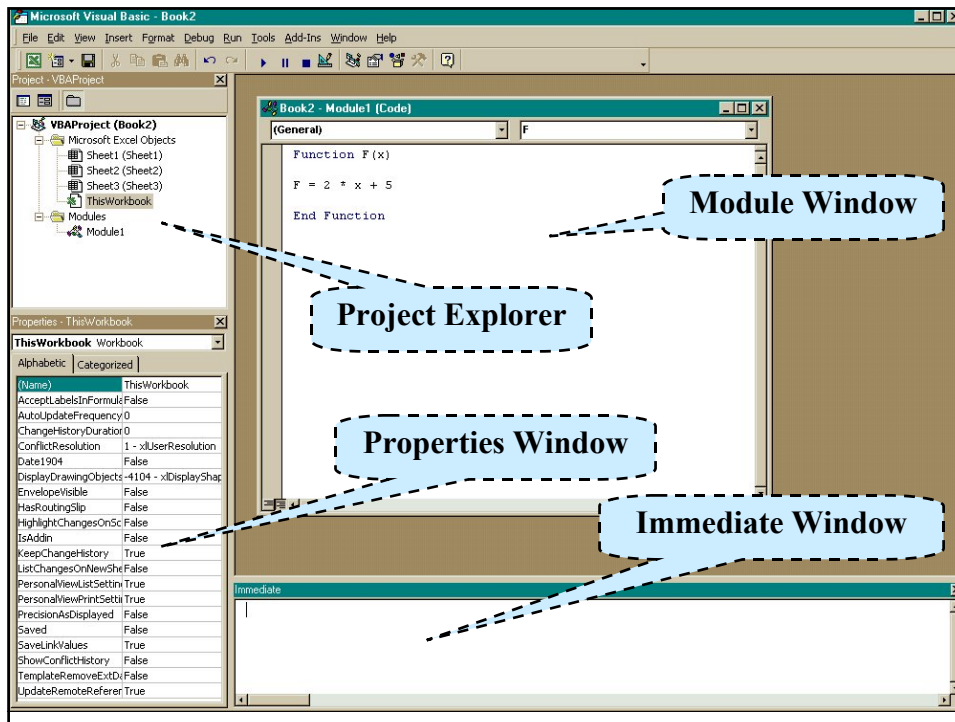
- The **Project Explorer** displays the hierarchical structure of projects.

- The **Properties Window** displays the properties of the projects.

- The **Module Window** contains the VBA-code of your project.

- The **Immediate Window** displays compiling messages.

44



- The Module Window might not be visible when you open VBE.
VBE menu bar: Insert → Module (LC)
 - The Immediate Window is made visible by
VBE menu bar: View → Immediate Window (LC)
 - You return to the Excel window by:
 - LC on the Excel icon in the windows toolbar.
 - LC on the Excel icon in the VBE toolbar.
 - Use the keyboard shortcut Alt+F11.
 - Writing any kind of computing program consists of three basic principal steps:
 - i) Design an **algorithm** which will perform the task you want.
 - ii) Translate the algorithm into a computer language (code) with a certain **syntax**, e.g. VBA in our case.
 - iii) Test (**debug**) your program thoroughly.
- These steps are not carried out just once in consecutive order! **46**