PRINT command:

PRINT statement is the statement which causes output to be displayed on the screen. If one wants words or letters printed they must be put between quotation marks ("). The words or letters between the quotation marks are called a **string**. To print numbers one does not require quotation marks.

Notes:

- Variables can be PRINTed by adding the variables title after the PRINT command.
- Math problems can also be PRINTed.
- More than one thing can be PRINTed on a line by adding semicolons between separate things you want PRINTed.
- You must PRINT spaces between string variables and other things you want PRINTed.
- If a semicolon follows a PRINT statement, the next thing PRINTed will be on the same line.
- If you type PRINT without any variables or text after it, it will PRINT a blank line.

Examples:

```
PRINT "Hello "                    [ENTER then F5]
Hello

PRINT " 512+456 "                 [ENTER then F5]
512+456

PRINT 523                         [ENTER then F5]
523
```

□ INPUT command:

INPUT is a command that takes information from the user of the program. The person types in information, and it is assigned to a variable. You INPUT like this:

INPUT [*Variable name*].

This prints a question mark (?) and puts a cursor on the screen where you should begin to type. After all desired information is inputted, press enter, and it is saved in the array specified by the INPUT command.

String and numeric variables can be written with this command, but if you type characters other than numbers for numeric variables, the computer will not accept the information, and will write "Redo from start", and you will be given a second chance to INPUT the information.

(input number or string without a message)

```
INPUT w   (for a number)
INPUT w$   (for a string)
```

(input with a message)

```
INPUT ‟prompt ″, w
INPUT ‟prompt ″, w$
```

Example:

```
PRINT “What is your name ? "
INPUT name$
PRINT " Hello “; name$; " nice to meet you ! “
PRINT “How old are you ? "
INPUT age
PRINT “ You are “; age; “years old “; name$
END
```

```
PRINT “This Program computes the perimeter of a rectangle.”
PRINT “Please enter the length and width of the rectangle, ”
PRINT “separated by commas. ”
INPUT L,W                     or        INPUT “Length =“, L
                                        INPUT “Width =“, W
PRINT “The Perimeter is”; 2*L+2*W
END
```

Non-Trigonometric Functions

Function	Description	Example	Result
^	Exponentiation	PRINT 7^3	343
ABS (x)	Absolute Value (converts a negative number to a positive number)	PRINT ABS (16.2) PRINT ABS (-27.3)	16.2 27.3
EXP (x)	Natural Exponent (e^x)	PRINT EXP (4)	
FIX (x)	Integer Truncation (chops off anything after the decimal point)	PRINT FIX (4.728)	4
INT (x)	Largest Integer Truncation	PRINT FIX (-4.728)	4
LOG (x)	Natural Logarithm (inverse of Natural Exponent)	LOG(10)	1
RND (x)	Random Number, produce a random number from 0 to 1		
SGN (x)	Sign Determination	PRINT SGN (14) PRINT SGN (0) PRINT SGN (-244)	1 0 -1
SQR (x)	Square Root	PRINT SQR (625)	25

Trigonometric Functions

Function	Description	Example	Result
COS (x)	Cosine	CONST PI=3.141593 PRINT COS (PI / 4)	0.7071067
SIN (x)	Sine	PRINT SIN (PI / 3)	0.8660254
TAN (x)	Tangent	PRINT TAN (-PI / 2)	6137956
ATN (x)	Arctangent, inverse of TAN	PRINT ATN (TAN (-PI / 2)	1.570796

IF...THEN...ELSE

General Form

```
IF condition THEN
          statements
END IF
```

Purpose: One-way selection

```
IF condition THEN
          statements
ELSE
          statements
END IF
```

Purpose: Two-way selection

Used to compare an expression and then perform some task based on that expression.

ELSE: using this command enable the program to perform a different action if the statement is false.

END IF: allows to have multiple commands after the IF...THEN statement, but they must start on the line after the IF statement.END IF should appear right after the list of commands.

ELSEIF: this command allows to perform a secondary action if the first expression was false. Unlike ELSE, this task is only performed if a specified statement is true.

This command checks if an argument involving a variable is true. An argument may look like this:
IF b=18 THEN...

If the argument is true (b=18), then QBasic executes the command after the IF...THEN.

- IF...THEN asks only yes or no questions. It's either true or false.
- If the statement is true, the command(s) after the THEN are executed.
- If the statement is false, the command(s) after the THEN command are skipped.
- **AND** and **OR** can make IF...THEN commands more powerful by making the command ask more questions.
- Answers to questions asked about string variables must be in quotes. ("")
- The **ELSE** command follows the THEN command, and is executed if the IF...THEN question is false.

Example:

```
IF b=18 THEN PRINT "OK "
```

If the argument is not true (if b is not equal to 18), QBasic bypasses this line and goes to the next. In some cases, ELSE command could be used which tells QBasic exactly what to do if the argument is not true.

```
IF b=18 THEN
                PRINT "OK “
ELSE
                PRINT "It is not 18 “
END IF
```

Example:

```
X=16
IF (X=5) THEN
                INPUT a$
                PRINT a$
ELSE
                PRINT X*2
END IF
```

```
Output:
32
```

Example:

```
1 CLS
score=0
PRINT “How many days are there in a week?”
INPUT a
IF a=7 THEN
                GOTO 2
ELSE
                PRINT “Wrong answer!”
                PRINT “To try again – press y”
                INPUT a$
IF a$= “y” THEN
                 PRINT “OK, let’s try again.”
                GOTO 1
ELSE
                END

2 score =10
PRINT “It’s the right answer!”
PRINT “Your score is now”; score; “!”
PRINT “Thanks for playing.”
END
```

Example:

```
CLS
X=5
IF (X >= 5) THEN PRINT “X is greater than or equal to 5”
IF (X <= 5) THEN PRINT “X is less than or equal to 5”
IF (X <> 5) THEN PRINT “X does not equal to 5”
```

Output :

```
X is greater than or equal to 5
X is less than or equal to 5
```

Example:

```
G=8
IF (G=5) THEN
                PRINT “Statement 1 is true “
ELSEIF (G=6) THEN
                PRINT “Statement 2 is true “
ELSEIF (G=7) THEN
                PRINT “Statement 3 is true “
ELSE
                PRINT “No above statements are true “
END IF
```

Output:

```
No above statements are true
```

IF...THEN [Multiple expressions]:

You can have more than one expression in IF...THEN by using either the **OR** operator or the **AND** operator.
The **OR** operator only *requires one expression to be true* in order to print “Yes” in the following program:

```
K=20
IF (K=5 OR K=20) THEN PRINT “Yes”
```

Output:

```
Yes
```

The **AND** operator *requires both expressions to be true*.

Example:

```
K=7
IF (K>5 AND K<20) THEN PRINT “True”
```

Output:

```
True
```

Example:

```
X=16
Y=3
IF ((X>5 AND X<10) OR Y=3) THEN PRINT “Correct”
```

Output:

```
Correct
```

Using Strings in IF...THEN:

Example:

```
M$="Hello "
IF (M$= "Hello " OR M$= "World ") THEN PRINT M$
```

Output:

```
Hello
```

Also, You can compare two variable strings.

Example:

```
M$="Hello "
Z$= "World "
IF (M$<>Z$) THEN PRINT M$;"  ";Z$
```

Output:

```
Hello World
```