QBASIC PROJECT

by James Singleton

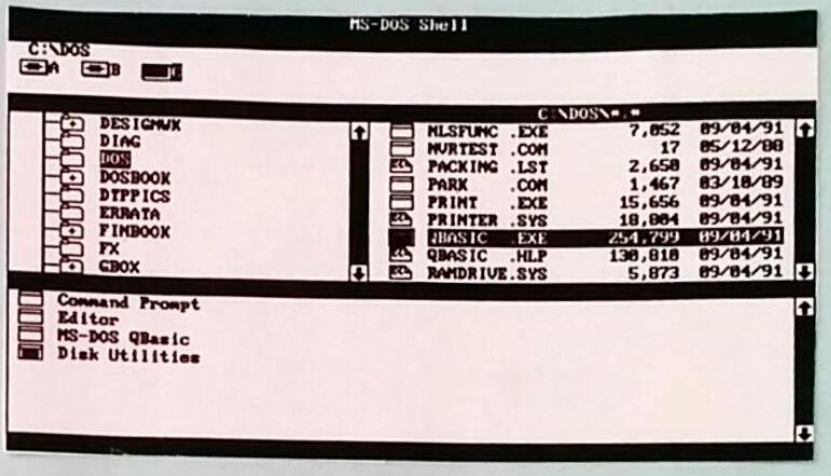
Age 11

Getting Started

To start learning Q Basic programing you have to have a copy of MS DOS 5.0 or higher. If you have a lower version of Dos it will come with GW Basic. If you own Dos 6.0 or higher you won't have the sample programs like Gorillas and Nibbles. You can send away for them from Microsoft.

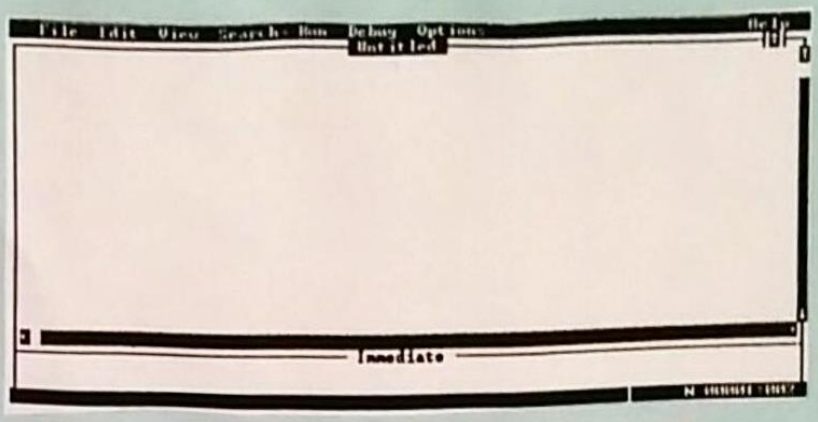
To load up Q basic type CD\DOS or whatever your Dos directory is then ENTER Then type QBASIC then ENTER.

Or use 'Windows' or 'Dos shell'

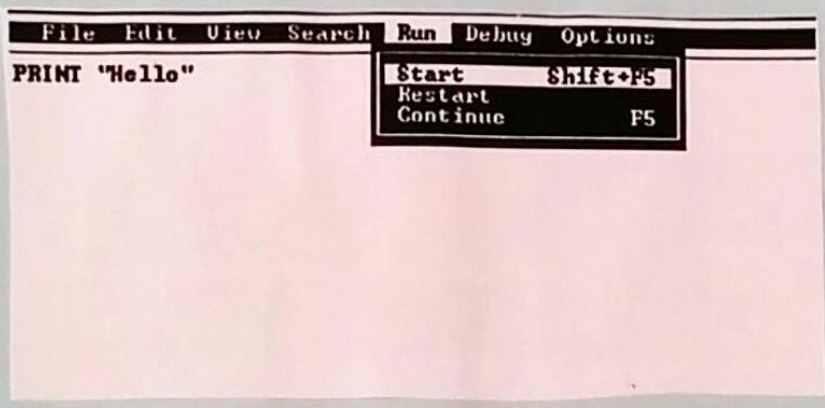


When you get in to QBasic press ESC to get in to the editor.

The editor menu is much the same as a word prosseser except for the RUN and DEBUG menus.



I'll talk about the DEBUG menu later. For now click on the RUN menu with your mouse if you have one or press ALT+R click on START or press S. Notice the SHIFT+F5 by the side of it. This is the shortcut key(s).



You can press these any time to do the function they are written next to, even without going into the menu.

Either way this runs your programme once you've typed it.

Simple Programmes

In a programme it is always best to type in CLS. This clears the screen otherwise the output screen would show what was on there before you started Qbasic

Result from programme without CLS. →

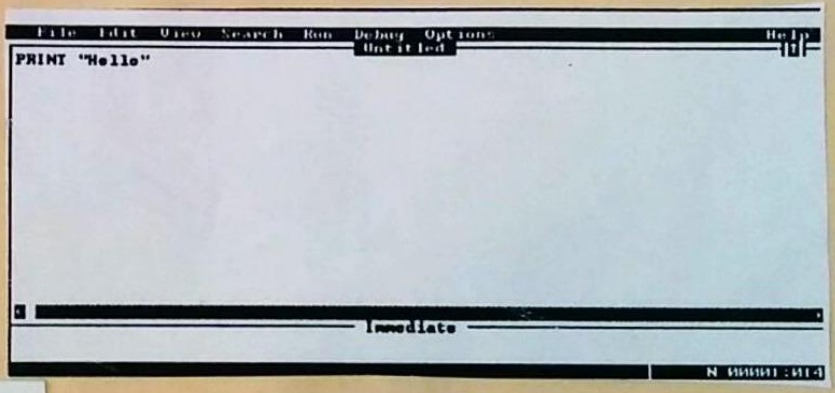
```
C:\DOS>qbasic
Hello
```

Type in beep in lower case letters and then press ENTER. Qbasic recognised the beep as a comand and changed it to upper case letters. Now press Shift+F5. The editor changes to a black screen and your speaker should beep.

PRINT

Now start a new programme and type:

```
CLS
PRINT "Hello"
```



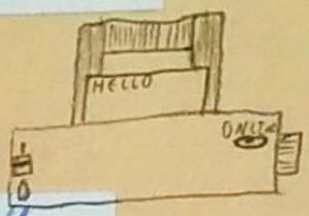
Programme without CLS. ↗

Then press Shift + F5. The output screen should look something like this.

Hello

The PRINT command displays the words in the speech marks.

If you type in LPRINT it will print the words on your printer.



INPUT

You still can't get information into the program. Type in:

```
CLS
INPUT "What is your name"; Name$
PRINT "Hello" Name$
```

Start the program and enter your name. Your output screen should look like this with your name instead of the example.

```
What is your name? James
Hello James
```

The Input comand lets you put information into your program. Notice that after the Input statment there is a word. This is called the variable. The variable stores the answer to the Input statment. The symbol at the end of the variable determins what kind of variable it is. In this case its a string variable. The semi colon

at the end of the statment gives the questionmark. To stop this put a comma instead of the semi colon.

IF-THEN

The input statment lets you get information from the user and store it in a variable. You can use this information with the IF-THEN statment like this:

```
CLS
INPUT "What is your name"; Name$
IF Name$ = James THEN
PRINT "Hello James."
ELSE PRINT "Go Away!"
ENDIF
```

This pass program only lets a person get into it. The else in the program does the function if anything else is typed in. At the end of the If-Then statment you have to put endif.

What if you had some friends you wanted to let in as well? Use the elseif statment like this:

P.T.O.

```
CLS
INPUT "What is your name"; Name$
IF Name$ = James THEN
PRINT "Hello James."
ELSEIF Name$ = Adam THEN
PRINT "Hi Adam."
ESLE PRINT "Go Away!"
ENDIF
```

This program will let you or your friend in.

```
What is your name? Adam
Hi Adam

What is your name? James
Hello James

What is your name? Ben
Go Away!
```

Variable types

Apart from the string variable Qbasic works out what kind of variable you need, but if you want to specify what kind of variable you want put these signs after the name.

Integer : %

Long integer : &

Single precision : !

Double precision : #

String : $

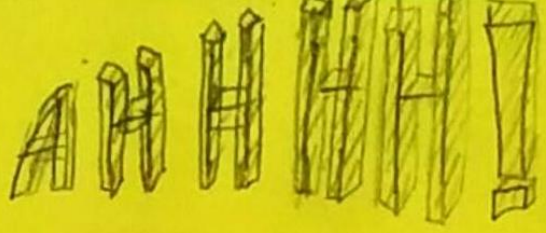
An integer has a range of -32,768 to 32,767 and can't be a decimal number.

A long integer can't be a decimal but has a longer range :-2,147,483,648 to 2,147,483,647 to be exact

A single precision number can have 6 decimal places whilst double precision numbers can have up to 14 decimal places.

Don't use long integers or double precision unless you have to because they take up more memory.

Maths



Using sums in your programmes is simple. You use a symble like + and -. But perhaps some are different. Use this table to help you:

Usually you use = at the end of a sum. Here are some alternatives:

Operator	Meaning
=	Equals.
<7	Not Equal.
<	Less than.
>	Greater than.
<=	Less than or Equal
>=	Greater than or Equal

Here are the results:

Operation	Result
5+8	13
12-7	5
3*6	18
10/3	3.3333r
10\3	3
5^2	25
10 MOD 3	1

Operator	Name	Same as / Explanation
+	Addition	+
-	Subtraction	-
*	Multiplication	x
/	Division	÷
\	Integer Division	Whole number division
^	Exponentiation	4^{3}
MOD	Modulus	Remainder

Loops:

As your programmes get more complex they will get longer. Loops help make them get shorter by repeating part of the programme which otherwise would have to be retyped.

For - Next

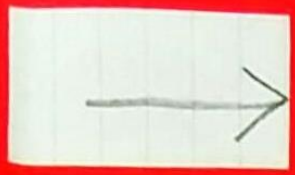
For - Next repeats a program part a certain number of times. eg.

```
CLS
INPUT "What is your name"; Name$
FOR x = 1 TO 100
PRINT ; Name$;
Next x
```

This programme prints your name 100 times using the For—Next loop. The value of x changes every time. The part of the programme that is repeated is between the For and Next.

While

The While loop doesn't loop a specific number of times like the For - Next loop. It loops until a condition is met. At the end of the loop you put wend. Try this programme:



```
CLS
num = 1
WHILE num <> 0
    INPUT "Please enter a number.", num
WEND
PRINT
PRINT "Looping is finished."
```

You must initialize a WHILE loop's control variable.

If you don't initialize your variable QBasic will set it to 0.

Here is what your screen could look like:

```
Please enter a number: 8
Please enter a number: 45
Please enter a number: -5
Please enter a number: 0

Looping is finished.
```

You can use a WHILE loop to replay an IF-THEN statement.

Do

The **Do loop** does much the same as the WHILE loop except that the do loop is executed at least once while the WHILE loop might not be executed at all. Try this programme:

```
CLS
DO
  INPUT "Please enter a number", num
LOOP UNTIL num = 0
PRINT
PRINT "Looping is finished."
```

The result could look like the While loop's result.

The LOOP UNTIL waits until the expresion is true. If you want it to wait until the expresion is false use LOOP WHILE.

This chart shows you when to use different loops:

Circumstance	Best Method
1. Loop a specific number of times.	For - Next
2. Loop until a condition is met.	WHILE or DO
3. Loop 0 or more times.	WHILE
4. Loop at least once.	DO

Debug Menu

As your programs get longer finding errors in them can sometimes prove impossible. QBasic can help. Click on the Debug menu or press ALT-D. The first command is step - It lets you execute one line at a time. The line highlighted is the next to be executed. Press F8 as an easier way. Press F4 to see the output screen. You can check the value of a variable by printing it in the immediate window.