

BASIC

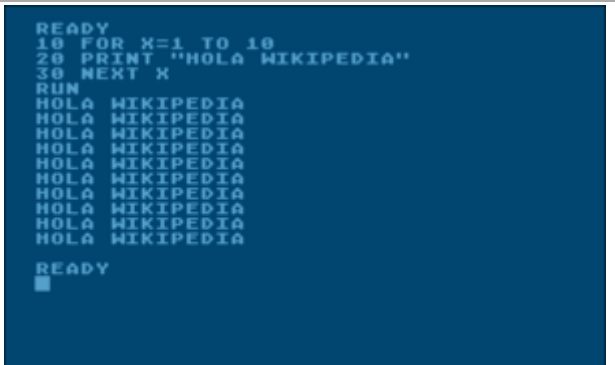
BASIC (Beginners' All-purpose Symbolic Instruction Code)^[1] is a family of general-purpose, high-level programming languages whose design philosophy emphasizes ease of use. The original version was designed by John G. Kemeny and Thomas E. Kurtz and released at Dartmouth College in 1964. They wanted to enable students in fields other than science and mathematics to use computers. At the time, nearly all use of computers required writing custom software, which was something only scientists and mathematicians tended to learn.

In addition to the language itself, Kemeny and Kurtz developed the Dartmouth Time Sharing System (DTSS), which allowed multiple users to edit and run BASIC programs at the same time. This general model became very popular on minicomputer systems like the PDP-11 and Data General Nova in the late 1960s and early 1970s. Hewlett-Packard produced an entire computer line for this method of operation, introducing the HP2000 series in the late 1960s and continuing sales into the 1980s. Many early video games trace their history to one of these versions of BASIC.


The emergence of early microcomputers in the mid-1970s led to the development of a number of BASIC dialects, including Microsoft BASIC in 1975. Due to the tiny main memory available on these machines, often 4 kB, a variety of Tiny BASIC dialects was also created. BASIC was available for almost any system of the era, and naturally became the *de facto* programming language for the home computer systems that emerged in the late 1970s. These machines almost always had a BASIC interpreter installed by default, often in the machine's firmware or sometimes on a ROM cartridge.

BASIC fell from use during the later 1980s as newer machines with far greater capabilities came to market and other programming languages (such as Pascal and C) became tenable. In 1991, Microsoft released Visual Basic, combining a greatly updated version of BASIC with a visual forms builder. This reignited use of the language and "VB" remains a major programming language in the form of VB.NET.

BASIC



Atari BASIC (1979) for the Atari 8-bit family

Paradigm	Non-structured, later <u>procedural</u> , later <u>object-oriented</u>
Designed by	<u>John G. Kemeny</u> <u>Thomas E. Kurtz</u>
First appeared	May 1, 1964
Major implementations	
<u>Dartmouth BASIC</u> · <u>Tiny BASIC</u> · <u>Applesoft BASIC</u> · <u>Atari BASIC</u> · <u>Sinclair BASIC</u> · <u>Commodore BASIC</u> · <u>BBC BASIC</u> · <u>TI-BASIC</u> · <u>Casio BASIC</u> · <u>Microsoft BASIC</u> · <u>QB64</u> · <u>Liberty BASIC</u> · <u>PowerBASIC</u> · <u>QBASIC</u>	
Influenced by	
<u>ALGOL 60</u> · <u>FORTRAN II</u> · <u>JOSS</u>	
Influenced	
<u>COMAL</u> · <u>Visual Basic</u> · <u>Visual Basic .NET</u> · <u>GRASS</u>	
 <u>BASIC Programming at Wikibooks</u>	

Contents

Origin

Spread on time-sharing services

Spread on minicomputers

Explosive growth: the home computer era

IBM PC and compatibles

Visual Basic

Post-1990 versions and dialects

Calculators

Windows command line

Other

Legacy

Syntax

Typical BASIC keywords

Data manipulation

Program flow control

Input and output

Mathematical functions

Miscellaneous

Data types and variables

Examples

Unstructured BASIC

Structured BASIC

Object-oriented BASIC

Standards

See also

Notes

References

General references

External links

Origin

John G. Kemeny was the math department chairman at Dartmouth College. Based largely on his reputation as an innovator in math teaching, in 1959 the school won an Alfred P. Sloan Foundation award for \$500,000 to build a new department building.^[2] Thomas E. Kurtz had joined the department in 1956, and from the 1960s Kemeny and Kurtz agreed on the need for programming literacy among students outside the traditional STEM fields. Kemeny later noted that "Our vision was that every student on campus should have access to a computer, and any faculty member should be able to use a computer in the classroom whenever appropriate. It was as simple as that."^[3]

Kemeny and Kurtz had made two previous experiments with simplified languages, DARSIMCO (Dartmouth Simplified Code) and DOPE (Dartmouth Oversimplified Programming Experiment). These did not progress past a single freshman class. New experiments using Fortran and ALGOL followed, but Kurtz concluded these languages were too tricky for what they desired. As Kurtz noted, Fortran had numerous oddly-formed commands, notably an "almost impossible-to-memorize convention for specifying a loop: 'DO 100, I = 1, 10, 2'. Is it '1, 10, 2' or '1, 2, 10', and is the comma after the line number required or not?"^[3]

Moreover, the lack of any sort of immediate feedback was a key problem; the machines of the era used batch processing and took a long time to complete a run of a program. While Kurtz was visiting MIT, Marvin Minsky suggested that time-sharing offered a solution; a single machine could divide up its processing time among many users, giving them the illusion of having a (slow) computer to themselves. Small programs would return results in a few seconds. This led to increasing interest in a system using time-sharing and a new language specifically for use by non-STEM students.^[3]

Kemeny wrote the first version of BASIC. The acronym *BASIC* comes from the name of an unpublished paper by Thomas Kurtz.^[4] The new language was heavily patterned on FORTRAN II; statements were one-to-a-line, numbers were used to indicate the target of loops and branches, and many of the commands were similar or identical to Fortran. However, the syntax was changed wherever it could be improved. For instance, the difficult to remember DO loop was replaced by the much easier to remember `FOR I = 1 TO 10 STEP 2`, and the line number used in the DO was instead indicated by the `NEXT I`.^[a] Likewise, the cryptic IF statement of Fortran, whose syntax matched a particular instruction of the machine on which it was originally written, became the simpler `IF I=5 THEN GOTO 100`. These changes made the language much less idiosyncratic while still having an overall structure and feel similar to the original FORTRAN.^[3]

The project received a \$300,000 grant from the National Science Foundation, which was used to purchase a GE-225 computer for processing, and a Datanet-30 realtime processor to handle the Teletype Model 33 teleprinters used for input and output. A team of a dozen undergraduates worked on the project for about a year, writing both the DTSS system and the BASIC compiler.^[3] The first version BASIC language was released on 1 May 1964.^{[5][6]} One of the graduate students on the implementation team was Mary Kenneth Keller, one of the first people in the United States to earn a Ph.D. in computer science and the first woman to do so.^[7]

Initially, BASIC concentrated on supporting straightforward mathematical work, with matrix arithmetic support from its initial implementation as a batch language, and character string functionality being added by 1965. Usage in the university rapidly expanded, requiring the main CPU to be replaced by a GE-235,^[3] and still later by a GE-635. By the early 1970s there were hundreds of terminals connected to the machines at Dartmouth, some of them remotely.

Wanting use of the language to become widespread, its designers made the compiler available free of charge. In the 1960s, software became a chargeable commodity; until then, it was provided without charge as a service with the very expensive computers, usually available only to lease. They also made it available to high schools in the Hanover, New Hampshire area and regionally throughout New England on Teletype Model 33 and Model 35 teleprinter terminals connected to Dartmouth via dial-up phone lines, and they put considerable effort into promoting the language. In the following years, as other dialects of BASIC appeared, Kemeny and Kurtz's original BASIC dialect became known as Dartmouth BASIC.

New Hampshire recognized the accomplishment in 2019 when it erected a highway historical marker recognizing the creation of BASIC.^[8]

Spread on time-sharing services

The emergence of BASIC took place as part of a wider movement towards time-sharing systems. First conceptualized during the late 1950s, the idea became so dominant in the computer industry by the early 1960s that its proponents were speaking of a future in which users would "buy time on the computer much the same way that the average household buys power and water from utility companies."^[9]

General Electric, having worked on the Dartmouth project, wrote their own underlying operating system and launched an online time-sharing system known as Mark I. It featured BASIC as one of its primary selling points. Other companies in the emerging field quickly followed suit; Tymshare introduced SUPER BASIC in

1968, CompuServe had a version on the DEC-10 at their launch in 1969, and by the early 1970s BASIC was largely universal on general-purpose mainframe computers. Even IBM eventually joined the club with the introduction of VS-BASIC in 1973.^[10]

Although time-sharing services with BASIC were successful for a time, the widespread success predicted earlier was not to be. The emergence of minicomputers during the same period, and especially low-cost microcomputers in the mid-1970s, allowed anyone to purchase and run their own systems rather than buy online time which was typically billed at dollars per minute. Tymshare charged about \$10 per hour (equivalent to \$66 in 2019) for accessing their systems.^[11]

Spread on minicomputers

BASIC, by its very nature of being small, was naturally suited to porting to the minicomputer market, which was emerging at the same time as the time-sharing services. These machines had very small main memory, perhaps as little as 4 kB in modern terminology,^[b] and lacked high-performance storage like hard drives that make compilers practical. On these systems, BASIC was normally implemented as an interpreter rather than a compiler due to the reduced need for working memory.^[c]

A particularly important example was HP Time-Shared BASIC, which, like the original Dartmouth system, used two computers working together to implement a time-sharing system. The first, a low-end machine in the HP 2100 series, was used to control user input and save and load their programs to tape or disk. The other, a high-end version of the same underlying machine, ran the programs and generated output. For the low-low cost of about \$100,000, one could own a machine capable of running between 16 and 32 users at the same time.^[12] The system, bundled as the HP 2000, was the first mini platform to offer time-sharing and was an immediate runaway success, catapulting HP to become the third-largest vendor in the minicomputer space, behind DEC and Data General (DG).^[13]



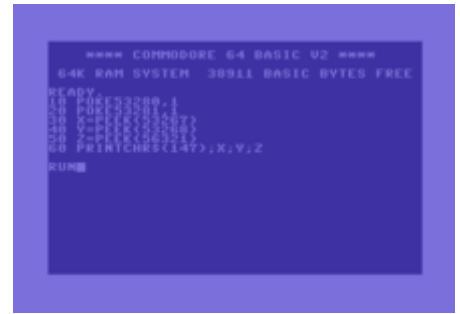
The HP 2000 system was designed to run time-shared BASIC as its primary task.

DEC, the leader in the minicomputer space since the mid-1960s, had initially ignored BASIC. This was due to their work with RAND Corporation, who had purchased a PDP-6 to run their JOSS language, which was conceptually very similar to BASIC.^[14] This led DEC to introduce a smaller, cleaned up version of JOSS known as FOCAL, which they heavily promoted in the late 1960s. However, with timesharing systems widely offering BASIC, and all of their competition in the minicomputer space doing the same, DEC's customers were clamoring for BASIC. After management repeatedly ignored their pleas, David H. Ahl took it upon himself to buy a BASIC for the PDP-8, which was a major success in the education market. By the early 1970s, FOCAL and JOSS had been forgotten and BASIC had become almost universal in the minicomputer market.^[15] DEC would go on to introduce their updated version, BASIC-PLUS, for use on the RSTS/E time-sharing operating system.

During this period a number of simple text-based games were written in BASIC, most notably Mike Mayfield's Star Trek. David Ahl collected these, some ported from FOCAL, and published them in an educational newsletter he compiled. He later collected a number of these into book form, *101 BASIC Computer Games*, published in 1973.^[16] During the same period, Ahl was involved in the creation of a small computer for education use, an early personal computer. When management refused to support the concept, Ahl left DEC in 1974 to found the seminal computer magazine, *Creative Computing*. The book remained popular, and was re-published on several occasions.^[17]

Explosive growth: the home computer era

The introduction of the first microcomputers in the mid-1970s was the start of explosive growth for BASIC. It had the advantage that it was fairly well known to the young designers and computer hobbyists who took an interest in microcomputers, many of whom had seen BASIC on minis or mainframes. Despite Dijkstra's famous judgement in 1975, "It is practically impossible to teach good programming to students that have had a prior exposure to BASIC: as potential programmers they are mentally mutilated beyond hope of regeneration",^[18] BASIC was one of the few languages that was both high-level enough to be usable by those without training and small enough to fit into the microcomputers of the day, making it the *de facto* standard programming language on early microcomputers.



Commodore BASIC v2.0 on the Commodore 64

The first microcomputer version of BASIC was co-written by Gates, Allen, and Monte Davidoff for their newly-formed company, Micro-Soft. This was released by MITS in punch tape format for the Altair 8800 shortly after the machine itself,^[19] immediately cementing BASIC as the primary language of early microcomputers. Members of the Homebrew Computer Club began circulating copies of the program, causing Gates to write his Open Letter to Hobbyists, complaining about this early example of software piracy.



MSX BASIC version 3.0

Partially in response to Gate's letter, and partially to make an even smaller BASIC that would run usefully on 4 kB machines,^[d] Bob Albrecht urged Dennis Allison to write their own variation of the language. How to design and implement a stripped-down version of an interpreter for the BASIC language was covered in articles by Allison in the first three quarterly issues of the People's Computer Company newsletter published in 1975 and implementations with source code published in Dr. Dobb's Journal of Tiny BASIC Calisthenics & Orthodontia: Running Light Without Overbyte. This led to a wide variety of Tiny BASICs with added features or other improvements, with versions from Tom Pittman and Li-Chen Wang becoming particularly well known.^[20]



"Train Basic every day!"—reads a poster (bottom center) in a Russian school (c. 1985–1986).

Micro-Soft, by this time Microsoft, ported their interpreter for the MOS 6502, which quickly become one of the most popular microprocessors of the 8-bit era. When new microcomputers began to appear, notably the "1977 trinity" of the TRS-80, Commodore PET and Apple II, they either included a version of the MS code, or quickly introduced new models with it. By 1978, MS BASIC was a *de facto* standard and practically every home computer of the 1980s included it in ROM. Upon boot, a BASIC interpreter in direct mode was presented.

Commodore Business Machines included Commodore BASIC, based on Microsoft BASIC. The Apple II and TRS-80 each had two versions of BASIC, a smaller introductory version introduced with the initial releases of the machines and an MS-based version introduced as interest in the platforms increased. As new companies entered the field, additional versions were added that subtly changed the BASIC family. The Atari 8-bit family had its own Atari BASIC that was modified in order to fit on an 8 kB ROM cartridge. Sinclair BASIC was introduced in 1980 with the Sinclair ZX-80, and was later extended for the Sinclair ZX-81 and the Sinclair ZX Spectrum. The BBC published BBC BASIC, developed by Acorn Computers Ltd, incorporating many extra structured programming keywords and advanced floating-point operation features.

As the popularity of BASIC grew in this period, computer magazines published complete source code in BASIC for video games, utilities, and other programs. Given BASIC's straightforward nature, it was a simple matter to type in the code from the magazine and execute the program. Different magazines were published featuring programs for specific computers, though some BASIC programs were considered universal and could be used in machines running any variant of BASIC (sometimes with minor adaptations). Many books of type-in programs were also available, and in particular, Ahl published versions of the original 101 BASIC games converted into the Microsoft dialect and published it from *Creative Computing* as BASIC Computer Games. This book, and its sequels, provided hundreds of ready-to-go programs that could be easily converted to practically any BASIC-running platform.^{[16][21][22]} The book reached the stores in 1978, just as the home computer market was starting off, and it became the first million-selling computer book. Later packages, such as Learn to Program BASIC would also have gaming as an introductory focus. On the business-focused CP/M computers which soon became widespread in small business environments, Microsoft BASIC (MBASIC) was one of the leading applications.^[23]

IBM PC and compatibles

When IBM was designing the IBM PC they followed the paradigm of existing home computers in wanting to have a built-in BASIC. They sourced this from Microsoft – IBM Cassette BASIC – but Microsoft also produced several other versions of BASIC for MS-DOS/PC DOS including IBM Disk BASIC (BASIC D), IBM BASICA (BASIC A), GW-BASIC (a BASICA-compatible version that did not need IBM's ROM) and QBasic, all typically bundled with the machine. In addition they produced the Microsoft BASIC Compiler aimed at professional programmers. Turbo Pascal-publisher Borland published Turbo Basic 1.0 in 1985 (successor versions are still being marketed by the original author under the name PowerBASIC). Microsoft wrote the windowed AmigaBASIC that was supplied with version 1.1 of the pre-emptive multitasking GUI Amiga computers (late 1985 / early 1986), although the product unusually did not bear any Microsoft marks.



IBM Cassette BASIC 1.10

These later variations introduced many extensions, such as improved string manipulation and graphics support, access to the file system and additional data types. More important were the facilities for structured programming, including additional control structures and proper subroutines supporting local variables. However, by the latter half of the 1980s, users were increasingly using pre-made applications written by others rather than learning programming themselves; while professional programmers now had a wide range of more advanced languages available on small computers. C and later C++ became the languages of choice for professional "shrink wrap" application development.^{[24][25]}

Visual Basic

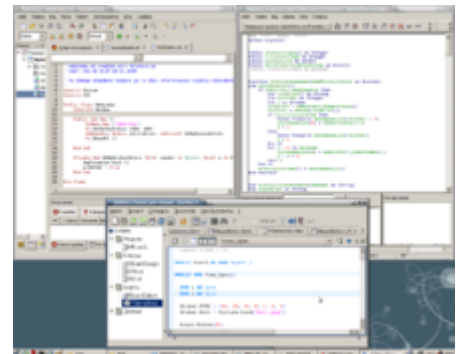
In 1991 Microsoft introduced Visual Basic, an evolutionary development of QuickBasic. It included constructs from that language such as block-structured control statements, parameterized subroutines and optional static typing as well as object-oriented constructs from other languages such as "With" and "For Each". The language retained some compatibility with its predecessors, such as the Dim keyword for declarations, "Gosub"/Return statements and optional line numbers which could be used to locate errors. An important driver for the development of Visual Basic was as the new macro language for Microsoft Excel, a spreadsheet program. To the surprise of many at Microsoft who still initially marketed it as a language for hobbyists, the language came into widespread use for small custom business applications shortly after the release of VB version 3.0, which is widely considered the first relatively stable version.

While many advanced programmers still scoffed at its use, VB met the needs of small businesses efficiently as by that time, computers running Windows 3.1 had become fast enough that many business-related processes could be completed "in the blink of an eye" even using a "slow" language, as long as large amounts of data were not involved. Many small business owners found they could create their own small, yet useful applications in a few evenings to meet their own specialized needs. Eventually, during the lengthy lifetime of VB3, knowledge of Visual Basic had become a marketable job skill. Microsoft also produced VBScript in 1996 and Visual Basic .NET in 2001. The latter has essentially the same power as C# and Java but with syntax that reflects the original Basic language.

Post-1990 versions and dialects

Many other BASIC dialects have also sprung up since 1990, including the open source QB64 and FreeBASIC, inspired by QBasic, and the Visual Basic-styled RapidQ, Basic For Qt and Gambas. Modern commercial incarnations include PureBasic, PowerBASIC, Xojo, Monkey X and True BASIC (the direct successor to Dartmouth BASIC from a company controlled by Kurtz).

Several web-based simple BASIC interpreters also now exist, including Quite BASIC (<http://www.quitebasic.com/>) and Microsoft's Small Basic. Many versions of BASIC are also now available for smartphones and tablets via the Apple App Store, or Google Play store for Android. On game consoles, an application for the Nintendo 3DS and Nintendo DSi called Petit Computer allows for programming in a slightly modified version of BASIC with DS button support.



Three modern Basic variants: Mono Basic, OpenOffice.org Basic and Gambas

Calculators

Variants of BASIC are available on graphing and otherwise programmable calculators made by Texas Instruments, HP, Casio, and others.

Windows command line

QBasic, a version of Microsoft QuickBASIC without the linker to make EXE files, is present in the Windows NT and DOS-Windows 95 streams of operating systems and can be obtained for more recent releases like Windows 7 which do not have them. Prior to DOS 5, the Basic interpreter was GW-Basic. QuickBasic is part of a series of three languages issued by Microsoft for the home and office power user and small-scale professional development; QuickC and QuickPascal are the other two. For Windows 95 and 98, which do not have QBasic installed by default, they can be copied from the installation disc, which will have a set of directories for old and optional software; other missing commands like Exe2Bin and others are in these same directories.

Other

The various Microsoft, Lotus, and Corel office suites and related products are programmable with Visual Basic in one form or another, including LotusScript, which is very similar to VBA 6. The Host Explorer terminal emulator uses WWB as a macro language; or more recently the programme and the suite in which it is contained is programmable in an in-house Basic variant known as Hummingbird Basic. The VBScript variant is used for programming web content, Outlook 97, Internet Explorer, and the Windows Script Host. WSH also has a Visual Basic for Applications (VBA) engine installed as the third of the default engines along with

VBScript, JScript, and the numerous proprietary or open source engines which can be installed like [PerlScript](#), a couple of Rexx-based engines, Python, Ruby, Tcl, Delphi, XLNT, PHP, and others; meaning that the two versions of Basic can be used along with the other mentioned languages, as well as LotusScript, in a WSF file, through the component object model, and other WSH and VBA constructions. VBScript is one of the languages that can be accessed by the 4Dos, 4NT, and Take Command enhanced shells. SaxBasic and WWB are also very similar to the Visual Basic line of Basic implementations. The pre-Office 97 macro language for Microsoft Word is known as WordBASIC. Excel 4 and 5 use Visual Basic itself as a macro language. [Chipmunk Basic](#), an old school interpreter similar to BASICs of the 1970s, is available for [Linux](#), [Microsoft Windows](#) and [macOS](#).



BASIC came to some video game systems, such as the Nintendo [Famicom](#).

Legacy

The ubiquity of BASIC interpreters on personal computers was such that textbooks once included simple "Try It In BASIC" exercises that encouraged students to experiment with mathematical and computational concepts on classroom or home computers. Popular computer magazines of the day typically included [type-in programs](#).

Futurist and sci-fi writer [David Brin](#) mourned the loss of ubiquitous BASIC in a 2006 *Salon* article^[26] as have others who first used computers during this era. In turn, the article prompted Microsoft to develop and release [Small Basic](#).^[27] Dartmouth held a 50th anniversary celebration for BASIC on 1 May 2014,^[28] as did other organisations; at least one organisation of VBA programmers organised a 35th anniversary observance in 1999.^[29]

Dartmouth College celebrated the 50th anniversary of the BASIC language with a day of events^[30] on April 30, 2014. A short documentary film was produced for the event.^[31]

Syntax

Typical BASIC keywords

Data manipulation

- LET**
assigns a value (which may be the result of an [expression](#)) to a variable. In most dialects of BASIC, LET is optional, and a line with no other identifiable keyword will assume the keyword to be LET.
- DATA**
holds a list of values which are assigned sequentially using the READ command.
- READ**
reads a value from a DATA statement and assigns it to a variable. An internal pointer keeps track of the last DATA element that was read and moves it one position forward with each READ.
- RESTORE**
resets the internal pointer to the first DATA statement, allowing the program to begin READING from the first value.

Program flow control

IF ... THEN ... {ELSE}

used to perform comparisons or make decisions. ELSE was not widely supported, especially in earlier versions.

FOR ... TO ... {STEP} ... NEXT

repeat a section of code a given number of times. A variable that acts as a counter is available within the loop.

WHILE ... WEND and REPEAT ... UNTIL

repeat a section of code while the specified condition is true. The condition may be evaluated before each iteration of the loop, or after. Both of these commands are found mostly in later dialects.

DO ... LOOP {WHILE} OR {UNTIL}

repeat a section of code indefinitely or while/until the specified condition is true. The condition may be evaluated before each iteration of the loop, or after. Similar to WHILE, these keywords are mostly found in later dialects.

GOTO

jumps to a numbered or labelled line in the program.

GOSUB

jumps to a numbered or labelled line, executes the code it finds there until it reaches a RETURN command, on which it jumps back to the statement following the GOSUB, either after a colon, or on the next line. This is used to implement subroutines.

ON ... GOTO/GOSUB

chooses where to jump based on the specified conditions. See Switch statement for other forms.

DEF FN

a pair of keywords introduced in the early 1960s to define functions. The original BASIC functions were modelled on FORTRAN single-line functions. BASIC functions were one expression with variable arguments, rather than subroutines, with a syntax on the model of $\text{DEF FND}(x) = x*x$ at the beginning of a program. Function names were originally restricted to FN, plus one letter, *i.e.*, FNA, FNB ...

Input and output

LIST

displays the full source code of the current program.

PRINT

displays a message on the screen or other output device.

INPUT

asks the user to enter the value of a variable. The statement may include a prompt message.

TAB

used with PRINT to set the position where the next character will be shown on the screen or printed on paper. AT is an alternative form.

SPC

prints out a number of space characters. Similar in concept to TAB but moves by a number of additional spaces from the current column rather than moving to a specified column.

Mathematical functions

ABS

Absolute value

ATN

Arctangent (result in radians)

COS

Cosine (argument in radians)

EXP

	Exponential function
INT	Integer part (typically <u>floor function</u>)
LOG	Natural logarithm
RND	<u>Random number generation</u>
SIN	Sine (argument in <u>radians</u>)
SQR	Square root
TAN	Tangent (argument in <u>radians</u>)

Miscellaneous

REM	holds a programmer's comment or REMark; often used to give a title to the program and to help identify the purpose of a given section of code.
USR	transfers program control to a <u>machine language</u> subroutine, usually entered as an alphanumeric <u>string</u> or in a list of DATA statements.
CALL	alternative form of USR found in some dialects. Does not require an artificial parameter to complete the function-like syntax of USR, and has a clearly defined method of calling different routines in memory.
TRON	turns on display of each line number as it is run ("TRace ON"). This was useful for <u>debugging</u> or correcting of problems in a program.
TROFF	turns off the display of line numbers.
ASM	some compilers such as Freebasic, ^[32] Purebasic, ^[33] and Powerbasic ^[34] also support <u>inline assembly language</u> , allowing the programmer to intermix high-level and low-level code, typically prefixed with "ASM" or "!" statements.

Data types and variables

Minimal versions of BASIC had only integer variables and one- or two-letter variable names, which minimized requirements of limited and expensive memory (RAM). More powerful versions had floating-point arithmetic, and variables could be labelled with names six or more characters long. There were some problems and restrictions in early implementations; for example, Applesoft BASIC allowed variable names to be several characters long, but only the first two were significant, thus it was possible to inadvertently write a program with variables "LOSS" and "LOAN", which would be treated as being the same; assigning a value to "LOAN" would silently overwrite the value intended as "LOSS". Keywords could not be used in variables in many early BASICs; "SCORE" would be interpreted as "SC" OR "E", where OR was a keyword. String variables are usually distinguished in many microcomputer dialects by having \$ suffixed to their name, and values are often identified as strings by being delimited by "double quotation marks". Arrays in BASIC could contain integers, floating point or string variables.

Some dialects of BASIC supported matrices and matrix operations, useful for the solution of sets of simultaneous linear algebraic equations. These dialects would directly support matrix operations such as assignment, addition, multiplication (of compatible matrix types), and evaluation of a determinant. Many

microcomputer BASICs did not support this data type; matrix operations were still possible, but had to be programmed explicitly on array elements.

Examples

Unstructured BASIC

New BASIC programmers on a home computer might start with a simple program, perhaps using the language's PRINT statement to display a message on the screen; a well-known and often-replicated example is Kernighan and Ritchie's "Hello, World!" program:

```
10 PRINT "Hello, World!"
20 END
```

An infinite loop could be used to fill the display with the message:

```
10 PRINT "Hello, World!"
20 GOTO 10
```

Note that the END statement is optional and has no action in most dialects of BASIC. It was not always included, as is the case in this example. This same program can be modified to print a fixed number of messages using the common FOR . . . NEXT statement:

```
10 LET N=10
20 FOR I=1 TO N
30 PRINT "Hello, World!"
40 NEXT I
```

Most first-generation BASIC versions, such as MSX BASIC and GW-BASIC, supported simple data types, loop cycles, and arrays. The following example is written for GW-BASIC, but will work in most versions of BASIC with minimal changes:

```
10 INPUT "What is your name: "; U$
20 PRINT "Hello "; U$
30 INPUT "How many stars do you want: "; N
40 S$ = ""
50 FOR I = 1 TO N
60 S$ = S$ + "*"
70 NEXT I
80 PRINT S$
90 INPUT "Do you want more stars? "; A$
100 IF LEN(A$) = 0 THEN GOTO 90
110 A$ = LEFT$(A$, 1)
120 IF A$ = "Y" OR A$ = "y" THEN GOTO 30
130 PRINT "Goodbye "; U$
140 END
```

The resulting dialog might resemble:

```
What is your name: Mike
Hello Mike
How many stars do you want: 7
*****
Do you want more stars? yes
How many stars do you want: 3
***
Do you want more stars? no
Goodbye Mike
```

The original Dartmouth Basic was unusual in having a matrix keyword, MAT.^[e] Although not implemented by most later microprocessor derivatives, it is used in this example from the 1968 manual^[35] which averages the numbers that are input:

```
5 LET S = 0
10 MAT INPUT V
20 LET N = NUM
30 IF N = 0 THEN 99
40 FOR I = 1 TO N
45 LET S = S + V(I)
50 NEXT I
60 PRINT S/N
70 GO TO 5
99 END
```

Structured BASIC

Second-generation BASICs (for example, VAX Basic, SuperBASIC, True BASIC, QuickBASIC, BBC BASIC, Pick BASIC, PowerBASIC, Liberty BASIC and (arguably) COMAL) introduced a number of features into the language, primarily related to structured and procedure-oriented programming. Usually, line numbering is omitted from the language and replaced with labels (for GOTO) and procedures to encourage easier and more flexible design.^[36] In addition keywords and structures to support repetition, selection and procedures with local variables were introduced.

The following example is in Microsoft QuickBASIC:

```
REM QuickBASIC example

REM Forward declaration - allows the main code to call a
REM subroutine that is defined later in the source code
DECLARE SUB PrintSomeStars (StarCount!)

REM Main program follows
INPUT "What is your name: ", UserName$
PRINT "Hello "; UserName$
DO
    INPUT "How many stars do you want: ", NumStars
    CALL PrintSomeStars(NumStars)
    DO
        INPUT "Do you want more stars? ", Answer$
        LOOP UNTIL Answer$ <> ""
        Answer$ = LEFT$(Answer$, 1)
    LOOP WHILE UCASE$(Answer$) = "Y"
PRINT "Goodbye "; UserName$
END

REM subroutine definition
SUB PrintSomeStars (StarCount)
    REM This procedure uses a local variable called Stars$
    Stars$ = STRING$(StarCount, "*")
    PRINT Stars$
END SUB
```

Object-oriented BASIC

Third-generation BASIC dialects such as Visual Basic, Xojo, StarOffice Basic, BlitzMax and PureBasic introduced features to support object-oriented and event-driven programming paradigm. Most built-in procedures and functions are now represented as *methods* of standard objects rather than *operators*. Also, the operating system became increasingly accessible to the BASIC language.

The following example is in Visual Basic .NET:

```

Public Module StarsProgram
    Private Function Ask(prompt As String) As String
        Console.Write(prompt)
        Return Console.ReadLine()
    End Function

    Public Sub Main()
        Dim userName = Ask("What is your name: ")
        Console.WriteLine("Hello {0}", userName)

        Dim answer As String

        Do
            Dim numStars = CInt(Ask("How many stars do you want: "))
            Dim stars As New String("*"c, numStars)
            Console.WriteLine(stars)

            Do
                answer = Ask("Do you want more stars? ")
            Loop Until answer <> ""
        Loop While answer.StartsWith("Y", StringComparison.OrdinalIgnoreCase)

        Console.WriteLine("Goodbye {0}", userName)
    End Sub
End Module

```

Standards

- ANSI/ISO/IEC Standard for Minimal BASIC:
 - ANSI X3.60-1978 "For minimal BASIC"
 - ISO/IEC 6373:1984 "Data Processing—Programming Languages—Minimal BASIC"
- ECMA-55 Minimal BASIC (*withdrawn, similar to ANSI X3.60-1978*)
- ANSI/ISO/IEC Standard for Full BASIC:
 - ANSI X3.113-1987 "Programming Languages Full BASIC"
 - INCITS/ISO/IEC 10279-1991 (R2005) "Information Technology – Programming Languages – Full BASIC"
- ANSI/ISO/IEC Addendum Defining Modules:
 - ANSI X3.113 Interpretations-1992 "BASIC Technical Information Bulletin # 1 Interpretations of ANSI 03.113-1987"
 - ISO/IEC 10279:1991/ Amd 1:1994 "Modules and Single Character Input Enhancement"
- ECMA-116 BASIC (*withdrawn, similar to ANSI X3.113-1987*)

See also

- List of BASIC dialects
- List of Computers With On-Board BASIC

Notes

- a. Fortran's DO had a `continue` for this purpose, but still required the line number to be entered.
- b. Widely regarded as the first "true" mini, the PDP-8's 12-bit memory space allowed 4,096 address of 12-bits each, or 6,144 bytes.
- c. Interpreters are ultimately similar to compilers in the tasks they perform, converting source code to machine code, but differ in when they perform it. Compilers convert the entire program at once and output a separate runnable program. Interpreters generally convert only a single line

- at a time (of even just a portion of it) and then immediately release that code once the line has completed running. This means they require only enough memory to run a single line, and do not require some form of high-performance secondary memory like a hard drive.
- d. Microsoft BASIC left 780 bytes free for user program code and variable values on a 4k machine, and that was running a cut-down version lacking string variables and other functionality.
 - e. From version 3 onwards.

References

1. Kemeny, John G.; Kurtz, Thomas E. (1964). *Basic: a manual for BASIC, the elementary algebraic language designed for use with the Dartmouth Time Sharing System* (http://bitsavers.trailing-edge.com/pdf/dartmouth/BASIC_Oct64.pdf) (PDF) (1st ed.). Hanover, N.H.: Dartmouth College Computation Center.
2. "High Math at Hanover" (<http://content.time.com/time/subscriber/article/0,33009,825616,00.htm>). *Time*. February 23, 1959.
3. *Time* 2014.
4. "BASIC" (<http://www.catb.org/jargon/html/B/BASIC.html>). *Jargon File*. Retrieved June 13, 2017.
5. "Thomas E. Kurtz – History of Computer Programming Languages" (<http://cis-alumni.org/TKurtz.html>). *cis-alumni.org*. Retrieved June 13, 2017.
6. Alfred, Randy (January 5, 2008). "May 1, 1964: First Basic Program Runs" (<https://www.wired.com/2008/05/dayintech-0501-2/>). *Wired*. Retrieved June 13, 2017.
7. Gurer, Denise (January 1995). "Pioneering women in computer science". *Communications of the ACM*. **38** (1): 45–54. doi:10.1145/204865.204875 (<https://doi.org/10.1145%2F204865.204875>).
8. Brooks, David (11 June 2019). "Finally, a historical marker that talks about something important" (<https://web.archive.org/web/20190611180750/https://granitegeek.concordmonitor.com/2019/06/11/finally-a-historical-marker-that-talks-about-something-important/>). *Concord Monitor*. Archived from the original (<https://granitegeek.concordmonitor.com/2019/06/11/finally-a-historical-marker-that-talks-about-something-important/>) on 11 June 2019. Retrieved 11 August 2019.
9. Bauer, W. F., *Computer design from the programmer's viewpoint* (<https://www.computer.org/web/csdl/index/-/csdl/proceedings/afips/1958/5053/00/50530046.pdf>) (Eastern Joint Computer Conference, December 1958) One of the first descriptions of computer time-sharing.
10. "IBM VS the World: That's How It Is" (<https://books.google.ca/books?id=sWeKU3wFLREC&pg=PA15>). *Computerworld*. December 5, 1973.
11. Bourne, Charles; Hahn, Trudi Bellardo (August 2003). *A History of Online Information Services, 1963–1976* (<https://books.google.ca/books?id=LTTvmUU8rskC&pg=PA387>). p. 387. ISBN 9780262261753.
12. "2000 Timeshare System" (https://www.hpmuseum.net/display_item.php?hw=411).
13. "Passing the 10-year mark" (http://hpmemoryproject.org/news/tenyears_comp/measure_page_00.htm). *MEASURE Magazine*. Hewlett Packard. October 1976.
14. Marks, Shirley (December 1971). *The JOSS Years: Reflections on an experiment* (<https://www.rand.org/content/dam/rand/pubs/reports/2008/R918.pdf>) (PDF) (Technical report). Rand.
15. Savetz, Kevin (April 2013). "Dave Ahl and Betsy Ahl" (https://computingpioneers.com/index.php?title=Dave_Ahl_and_Betsy_Ahl) (Interview).
16. Ahl, David H. (1973). *101 Basic computer games* (<http://www.atariarchives.org/basicgames/>). Morristown, N.J.: Creative Computing Press. OCLC 896774158 (<https://www.worldcat.org/oclc/896774158>).

17. Ahl, David H. (May 11, 1981). "Computer Games" (<https://books.google.com/books?id=Cz4EAAAMBAJ&pg=PA44>). *InfoWorld*. Vol. 3 no. 9. p. 44. ISSN 0199-6649 (<https://www.worldcat.org/issn/0199-6649>).
18. Dijkstra, Edsger W. (June 18, 1975). "How do we tell truths that might hurt" (<https://www.cs.utexas.edu/users/EWD/ewd04xx/EWD498.PDF>) (PDF). *Selected Writings on Computing: A Personal Perspective* (<https://archive.org/details/selectedwritings0000dijk/page/129>). Springer-Verlag (published 1982). pp. 129–131 (<https://archive.org/details/selectedwritings0000dijk/page/129>). ISBN 978-0387906522. OCLC 693424350 (<https://www.worldcat.org/oclc/693424350>).
19. "We have a BASIC" (<https://web.archive.org/web/20121130051115/http://startup.nmnaturalhistory.org/gallery/story.php?ii=20&sid=4>). New Mexico Museum of Natural History and Science. Archived from the original (<http://startup.nmnaturalhistory.org/gallery/story.php?ii=20&sid=4>) on November 30, 2012. Retrieved April 18, 2007.
20. Pittman, Tom. "you had to pay \$5 up front to get it..." (<http://www.ittybittycomputers.com/IttyBitty/iNotes.html#TinyBasic>). *www.ittybittycomputers.com*. Retrieved June 14, 2017.
21. Ahl, David H. (1979). *More basic computer games* (<http://www.atariarchives.org/morebasicgames/>). Morristown: Creative Computing Press. ISBN 978-0894801372. OCLC 839377789 (<http://www.worldcat.org/oclc/839377789>).
22. Ahl, David H. (1984). *Big computer games* (<http://www.atariarchives.org/bigcomputergames/>). Morris Plains, N.J.: Creative Computing Press. ISBN 978-0916688400. OCLC 872675092 (<http://www.worldcat.org/oclc/872675092>).
23. "Osborne 1" (<http://oldcomputers.net/osborne-1.html>). *oldcomputers.net*. Retrieved June 14, 2017.
24. Pravin, Jain (2011). *The Class Of Java* (<https://books.google.com/books?id=IGRpeBZQYPoC>). Pearson Education India. ISBN 9788131755440.
25. "GNE: the C programming language" (http://www.fysh.org/~zefram/gne/c_programming_language/). *fysh.org*. Retrieved June 14, 2017. "During the 1980s, C compilers spread widely, and C became an extremely popular language."
26. Brin, David (September 14, 2006). "Why Johnny Can't Code" (https://web.archive.org/web/20130918213750/http://www.salon.com/2006/09/14/basic_2/singleton/). *Salon*. Archived from the original (https://www.salon.com/2006/09/14/basic_2/) on September 18, 2013. Retrieved June 19, 2017.
27. "Small Basic" (<https://web.archive.org/web/20140317001756/http://msdn.microsoft.com/en-us/beginner/hh182234.aspx>). *Microsoft Developer Network*. Archived from the original (<http://msdn.microsoft.com/en-us/beginner/hh182234.aspx>) on March 17, 2014. Retrieved June 19, 2017.
28. "Dartmouth plans celebration for 50th anniversary of BASIC computer language" (<http://www.unionleader.com/apps/pbcs.dll/article?AID=/20140429/NEWS04/140429054>). *New Hampshire Union Leader*. April 28, 2014. Retrieved June 14, 2017.
29. "50th anniversary of BASIC – Google Search" (<https://www.google.com/search?q=50th+anniversary+of+BASIC>). *Google*.
30. "BASIC at 50 – Event Schedule" (<https://www.dartmouth.edu/basicfifty/events.html>). *Dartmouth College*. Retrieved June 14, 2017.
31. "BASIC at 50" (<https://www.dartmouth.edu/basicfifty/>). *Dartmouth College*. Retrieved June 14, 2017.
32. "KeyPgAsm" (<http://www.freebasic.net/wiki/wikka.php?wakka=KeyPgAsm>). *FreeBasic Wiki*. Retrieved August 2, 2017.
33. "Inline x86 ASM" (<https://www.purebasic.com/documentation/reference/inlinedasm.html>). *Pure Basic*. Retrieved August 2, 2017.
34. "Using assembly-language in your code" (https://web.archive.org/web/20170802164032/https://www.powerbasic.com/help/pbcc/using_assembly-language_in_your_code.htm). *Power Basic*. Archived from the original (https://www.powerbasic.com/help/pbcc/using_assembly-language_in_your_code.htm) on August 2, 2017. Retrieved August 2, 2017.

35. Kemeny, John G.; Kurtz, Thomas E. (January 1968). *Basic: a manual for BASIC, the elementary algebraic language designed for use with the Dartmouth Time Sharing System* (http://bitsavers.trailing-edge.com/pdf/dartmouth/BASIC_4th_Edition_Jan68.pdf) (PDF) (4th ed.). Hanover, N.H.: Dartmouth College Computation Center. p. 53.
36. "Differences Between GW-BASIC and QBasic" (<https://web.archive.org/web/20131019181140/http://support.microsoft.com/kb/73084>). May 12, 2003. Archived from the original (<http://support.microsoft.com/kb/73084>) on October 19, 2013. Retrieved June 28, 2008.

General references

- Sammet, Jean E. (1969). *Programming languages: history and fundamentals*. Englewood Cliffs, N.J.: Prentice-Hall. OCLC 819683527 (<https://www.worldcat.org/oclc/819683527>).
- Kurtz, Thomas E. (1981). "BASIC" (<http://doi.acm.org/10.1145/800025.1198404>). In Wexelblat, Richard (ed.). *History of programming languages* (<https://archive.org/details/historyofprogram000hist/page/515>). *History of Programming Languages I*. New York: ACM. pp. 515–537 (<https://archive.org/details/historyofprogram000hist/page/515>). doi:10.1145/800025.1198404 (<https://doi.org/10.1145%2F800025.1198404>). ISBN 978-0127450407.
- Kemeny, John G.; Kurtz, Thomas E. (1985). *Back To BASIC: The History, Corruption, and Future of the Language*. Addison-Wesley. p. 141. ISBN 9780201134339. OCLC 11399298 (<https://www.worldcat.org/oclc/11399298>).
- Lien, David A. (1986). *The Basic Handbook: Encyclopedia of the BASIC Computer Language* (3rd ed.). Compusoft Publishing. ISBN 9780932760333. OCLC 12548310 (<https://www.worldcat.org/oclc/12548310>).
- "Fifty Years of BASIC, the Programming Language That Made Computers Personal" (<http://time.com/69316/basic/>). *Time*. April 29, 2014.

External links

- BASIC (<https://curlie.org/Computers/Programming/Languages/BASIC>) at Curlie
- "BASIC—Beginners All-purpose Symbolic Instruction Code" (<http://hopl.info/showlanguage.prx?exp=176>). *The Encyclopedia of Computer Languages*. Murdoch University.
- The Birth of Basic (<https://www.youtube.com/watch?v=WYPNjSoDrqw>) on YouTube

Retrieved from "<https://en.wikipedia.org/w/index.php?title=BASIC&oldid=970659870>"

This page was last edited on 1 August 2020, at 17:10 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.