BASIC

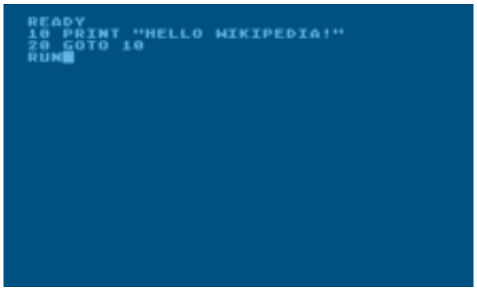
From Wikipedia, the free encyclopedia

BASIC is a family of general-purpose, high-level programming languages whose design philosophy emphasizes ease of use - the name is an acronym from **Beginner's All-purpose Symbolic Instruction Code**.

The original Dartmouth BASIC was designed in 1964 by John George Kemeny and Thomas Eugene Kurtz at Dartmouth College in New Hampshire, USA to provide computer access to non-science students. At the time, nearly all use of computers required writing custom software, which was something only scientists and mathematicians tended to do. The language and its variants became widespread on microcomputers in the late 1970s and 1980s, when it was typically a standard feature, and often actually built into the firmware of the machine.

BASIC remains popular to this day in a handful of highly modified dialects and new languages influenced by BASIC such as Microsoft Visual Basic. In 2006, 59% of developers for the .NET Framework used Visual Basic .NET as their only programming language.[1]

BASIC



```
READY
10 PRINT "HELLO WIKIPEDIA!"
20 GOTO 10
RUN
```

Screenshot of Atari BASIC, an early BASIC language for small computers

Paradigm(s)	unstructured, later procedural, later object-oriented
Appeared in	1964
Designed by	John George Kemeny and Thomas Eugene Kurtz
Major implementations	Apple BASIC, Atari BASIC, Commodore BASIC, Microsoft BASIC, Liberty BASIC, BBC BASIC, TI-BASIC
Influenced by	ALGOL 60, FORTRAN II, JOSS
Influenced	COMAL, Visual Basic, Visual Basic .NET, Realbasic, GRASS

Contents

History

Before the mid-1960s, computers were extremely expensive and used only for special-purpose tasks. A simple batch processing arrangement ran only a single "job" at a time, one after another. But during the 1960s faster and more affordable computers became available. and as prices decreased newer computer systems supported time-sharing, a system which allows multiple users or processes to use the CPU and memory. In such a system the operating system alternates between running processes, giving each one running time on the CPU before switching to another. The machines had become fast enough that most users could feel they had the machine all to themselves.

By this point the problem of interacting with the computer was a concern. In the batch processing model, users never interacted with the machine directly, instead they tended their jobs to the computer operators. Under the time-sharing model the users were given individual computer terminals and interacted directly. The need for system to simplify this experience, from command line interpreters to programming languages was an area of intense research during the 1960s and 70s.

Origin

Main article: Dartmouth BASIC

The original BASIC language was designed in 1964 by John Kemeny and Thomas Kurtz[2] and implemented by a team of Dartmouth students under their direction. The acronym is tied to the name of an unpublished paper by Thomas Kurtz and is not a backronym.[3] BASIC was designed to allow students to write programs for the Dartmouth Time-Sharing System. It was intended specifically for the new class of users that time-sharing systems allowed—that is, a less technical user who did not have the mathematical background of the more traditional users and was not interested in acquiring it. Being able to use a computer to support teaching and research was quite novel at the time.

The language was based on FORTRAN II, with some influences from ALGOL 60 and with additions to make it suitable for timesharing. Initially, BASIC concentrated on supporting straightforward mathematical work, with matrix arithmetic support from its initial implementation as a batch language and full string functionality being added by 1965.

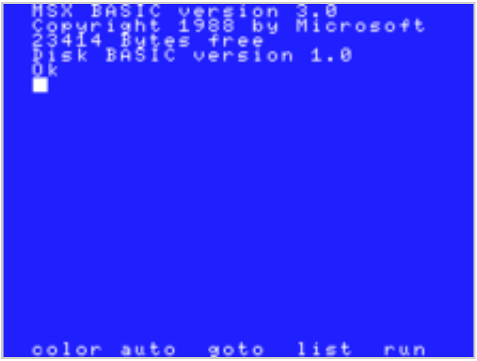
The designers of the language decided to make the compiler available free of charge so that the language would become widespread. They also made it available to high schools in the Hanover area and put a considerable amount of effort into promoting the language. In the following years, as other dialects of BASIC appeared, Kemeny and Kurtz's original BASIC dialect became known as *Dartmouth BASIC*.

Spread on minicomputers

As a result of its free availability, knowledge of BASIC became relatively widespread (for a computer language) and BASIC was implemented by a number of manufacturers, becoming fairly popular on newer minicomputers like the DEC PDP series and the Data General Nova. The BASIC language was also central to the HP Time-Shared BASIC system in the late 1960s and early 1970s, and the Pick operating system. In these instances the language tended to be implemented as an interpreter, instead of, or in addition to, a compiler.

It was during this period that a number of simple computer games were written in BASIC, most notably Mike Mayfield's *Star Trek*. A number of these were collected by David H. Ahl while he worked at DEC and published in a newsletter he compiled. He later collected a number of these into book form, "101 BASIC Computer Games", which was first published in 1973.[4][5] 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.[6]

Explosive growth: the home computer era



```
MSX BASIC version 3.0
Copyright 1988 by Microsoft
23414 Bytes free
Disk BASIC version 1.0
Ok
```

MSX BASIC version 3.0

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.

One of the first to appear was Tiny BASIC, a simple BASIC variant designed by Dennis Allison at the urging of Bob Albrecht of the Homebrew Computer Club. He had seen BASIC on minicomputers and felt it would be the perfect match for new machines like the MITS Altair 8800. 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. Version were written by Dr. Li-Chen Wang, and Tom Pittman.[7]

In 1975, MITS released Altair BASIC, developed by Bill Gates and Paul Allen as the company Micro-Soft,[8] which grew into today's corporate giant, Microsoft. The first Altair version was co-written by Gates, Allen, and Monte Davidoff.

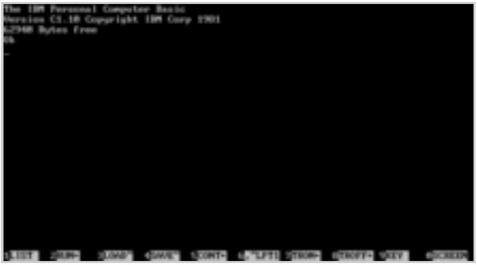
When the Apple II, PET 2001 and TRS-80 were all released in 1977, all three had BASIC as their primary programming language and operating environment. Commodore Business Machines included a version of Micro-Soft BASIC. The Apple II and TRS-80 each had two versions of BASIC, a smaller introductory version introduced with the initial releases of the machines and a more advanced version developed 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 their own Atari BASIC that was modified in order to fit on an 8 kB ROM cartridge. The BBC published BBC BASIC, developed for them by Acorn Computers Ltd, incorporating many extra structuring keywords. Almost universally, home computers of the 1980s had a ROM-resident BASIC interpreter, allowing the machines to boot directly into BASIC which therefore constituted a significant and visible part of the user interface of many home computers' rudimentary operating systems.[notes 1]

As the popularity of BASIC grew in this period, magazines (such as *Creative Computing* in the US) published complete source code in BASIC for 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 platform.[9][10][11] 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.[12]

IBM PC, and compatibles

When IBM were 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 QuickBASIC, all typically bundled with the machine. In addition they produced the Microsoft BASIC Compiler aimed at professional programmers.



```
The IBM Personal Computer Basic
Version C1.10 Copyright IBM Corp 1981
62940 Bytes free
Ok
```

IBM Cassette BASIC 1.10

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 windowing based 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.

These languages introduced many extensions to the original home computer BASIC, 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 applications written by others, rather than learning programming themselves, while professional programmers now had a wide range of more advanced languages available and BASIC tended to become the butt of derogatory comments such as Dijkstra's famous *"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*".[13]

Visual Basic

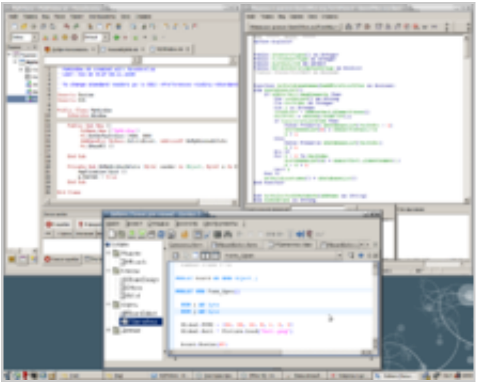
BASIC's fortunes reversed once again with the introduction in 1991 of Visual Basic ("VB"), by Microsoft. The only significant similarity to older BASIC dialects was familiar syntax. Syntax itself no longer "fully defined" the language, since much development was done using "drag and drop" methods without exposing all code for commonly used objects such as buttons and scrollbars to the developer. While this could be considered an evolution of the language, few of the distinctive features of early Dartmouth BASIC, such as line numbers and the `INPUT` keyword, remain (although Visual Basic still uses `INPUT` to read data from files, and `INPUTBOX` is available for direct user input; line numbers can also optionally be used in all VB versions, even VB.NET, albeit they cannot be used in certain places, for instance before `SUB`).

Ironically given the origin of BASIC as a "beginner's" language, and apparently even to the surprise of many at Microsoft who still initially marketed it as a language for hobbyists, the language had come 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 wherever processing speed was less of a concern than easy development. 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 massive 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.

Recent versions

Many BASIC dialects have also sprung up in the last few years, including Bywater BASIC and True BASIC (the direct successor to Dartmouth BASIC from a company controlled by Kurtz). One notable variant is RealBasic which although first released in 1998 for Macintosh computers, has since 2005 fully compiled programs for Microsoft Windows, Mac OS X and 32-bit x86 Linux, from the same object-oriented source code. RealBasic compiled programs may execute natively on these platforms as services, consoles or windowed applications. However in keeping with BASIC tradition, one-platform hobbyist versions are also still maintained. Many other BASIC variants and adaptations have been written by hobbyists, equipment developers, and others, as it is a relatively simple language to develop translators for. More complex examples of free software BASIC implementations (development tools and compilers) include Gambas and FreeBASIC.



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

Nostalgia

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 and computer magazines typically included type-in programs. Futurist and sci-fi writer David Brin mourned the loss of ubiquitous BASIC in a 2006 *Salon* article[14] as have others who first used computers during this era.

Syntax

Typical BASIC keywords

Data manipulation

- LET: assigns a value (which may be the result of an expression) to a variable.
- DATA: holds a list of values which are assigned sequentially using the READ command.

Program flow control

- IF ... THEN ... ELSE: used to perform comparisons or make decisions.
- 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.
- DO ... LOOP {WHILE} or {UNTIL}: repeat a section of code Forever or While/Until the specified condition is true . The condition may be evaluated before each iteration of the loop, or after.
- GOTO: jumps to a numbered or labelled line in the program.
- GOSUB: temporarily jumps to a numbered or labelled line, returning to the following line after encountering the RETURN Command. This is used to implement subroutines.
- ON ... GOTO/GOSUB: chooses where to jump based on the specified conditions. See Switch statement for other forms.

Input and output

- 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 or AT: sets the position where the next character will be shown on the screen or printed on paper.

Miscellaneous

- REM: holds a programmer's comment; often used to give a title to the program and to help identify the purpose of a given section of code.

Data types and variables

Minimal versions of BASIC had only integer variables and one-letter variable names. More powerful versions had floating-point arithmetic, and variables could be labelled with names six or more characters long.

String variables are usually distinguished in many microcomputer dialects by having $ suffixed to their name, and string values are typically enclosed in 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 support matrix operations such as assignment, addition, multiplication (of compatible matrix types), and evaluation of a determinant. Microcomputer dialects often lacked this data type and required a programmer to provide subroutines to carry out equvalent operations.

Examples

The original Dartmouth Basic was unusual in having a matrix keyword, MAT.[notes 2] Although dropped by most later microprocessor derivatives it is used in this example from the 1968 manual[15] 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
```

New BASIC programmers on a home computer might start with a simple program similar to the Hello world program made famous by Kernighan and Ritchie. This generally involves simple use of the language's PRINT statement to display the message (such as the programmer's name) to the screen. Often an infinite loop was used to fill the display with the message. Most first generation BASIC languages 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
```

Second generation BASICs (for example QuickBASIC and PowerBASIC) 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.[16]

```
INPUT "What is your name: ", UserName$
PRINT "Hello "; UserName$
DO
  INPUT "How many stars do you want: ", NumStars
  Stars$ = STRING$(NumStars, "*")
  PRINT Stars$
  DO
    INPUT "Do you want more stars? ", Answer$
  LOOP UNTIL Answer$ <> ""
  Answer$ = LEFT$(Answer$, 1)
LOOP WHILE UCASE$(Answer$) = "Y"
PRINT "Goodbye "; UserName$
```

Third generation BASIC dialects such as Visual Basic, REALbasic, StarOffice Basic and BlitzMax 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*.

The following example is in Visual Basic .NET:

```
Public Class StarsProgram
    Public Shared Sub Main()
        Dim UserName, Answer, stars As String, NumStars As Integer
        Console.Write("What is your name: ")
        UserName = Console.ReadLine()
        Console.WriteLine("Hello {0}", UserName)
        Do
            Console.Write("How many stars do you want: ")
            NumStars = CInt(Console.ReadLine())
            stars = New String("*", NumStars)
            Console.WriteLine(stars)
            Do
                Console.Write("Do you want more stars? ")
                Answer = Console.ReadLine()
            Loop Until Answer <> ""
            Answer = Answer.Substring(0, 1)
        Loop While Answer.ToUpper() = "Y"
        Console.WriteLine("Goodbye {0}", UserName)
    End Sub
End Class
```

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"
- 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" (http://www.iso.org/iso/catalogue_detail.htm?csnumber=18321)
- 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

References

Notes

1. ^ Probably the only exception was the Jupiter Ace, which instead used Forth.
2. ^ From version 3 onwards.

Citations

1. ^ Steven Vaughan-Nichols, "Mono brings Visual Basic programs to Linux" (http://www.linuxfordevices.com/c/a/News/Mono-brings-Visual-Basic-programs-to-Linux/) , LinuxDevices, 20 February 2007
2. ^ Thomas E. Kurtz - History of Programming Languages (http://cis-alumni.org/TKurtz.html)
3. ^ BASIC (http://www.catb.org/jargon/html/B/BASIC.html) , The Jargon File
4. ^ David H. Ahl, "BASIC Computer Games" (http://www.atariarchives.org/basicgames/showpage.php?page=i11) , p. XI
5. ^ David H. Ahl, "101 BASIC Computer Games" (http://books.google.ca/books?id=Ief0QwAACAAJ) , DEC, 1975
6. ^ David H. Ahl, "Computer Games" (http://books.google.ca/books?id=Cz4EAAAAMBAJ&pg=PA44) , *InfoWorld*, 11 May 1981, p. 44
7. ^ "you had to pay $5 up front to get it..." (http://www.ittybittycomputers.com/IttyBitty/iNotes.html#TinyBasic) , Tom Pittman's site
8. ^ "We have a BASIC" (http://startup.nmnaturalhistory.org/gallery/story.php?ii=20&sid=4) . New Mexico Museum of Natural History and Science. http://startup.nmnaturalhistory.org/gallery/story.php?ii=20&sid=4. Retrieved 2007-04-18.
9. ^ Table of Contents: BASIC Computer Games (http://www.atariarchives.org/basicgames)
10. ^ Table of Contents: More BASIC Computer Games (http://www.atariarchives.org/morebasicgames)
11. ^ Table of Contents: Big Computer Games (http://www.atariarchives.org/bigcomputergames)
12. ^ "Osborne 1" (http://oldcomputers.net/osborne.html) . OldComputers.net. http://oldcomputers.net/osborne.html. Retrieved 21 May 2010.
13. ^ Edsger Dijkstra, "How do we tell truths that might hurt?" (http://www.cs.virginia.edu/~evans/cs655/readings/ewd498.html) , 18 June 1975, from "Selected Writings on Computing: A Personal Perspective", Springer-Verlag, 1982
14. ^ Why Johnny Can't Code (http://www.salon.com/tech/feature/2006/09/14/basic/index_np.html) , By David Brin, Sept. 14, 2006, Salon Technology
15. ^ BASIC, 4th Edition, 1968 (http://www.bitsavers.org/pdf/dartmouth/BASIC_4th_Edition_Jan68.pdf) , page 53
16. ^ "Differences Between GW-BASIC and QBasic" (http://support.microsoft.com/kb/73084) . 2003-05-12. http://support.microsoft.com/kb/73084. Retrieved 2008-06-28.

Bibliography

- (PDF) *A Manual for BASIC, the elementary algebraic language designed for use with the Dartmouth Time Sharing System* (http://www.bitsavers.org/pdf/dartmouth/BASIC_Oct64.pdf) . Dartmouth College Computation Center. 1964. http://www.bitsavers.org/pdf/dartmouth/BASIC_Oct64.pdf.
- Lien, David A. (1986). *The Basic Handbook: Encyclopedia of the BASIC Computer Language* (3rd ed. ed.). Compusoft Publishing. ISBN 0-932760-33-3.
- Kemeny, John G.; Kurtz, Thomas E. (1985). *Back To BASIC: The History, Corruption, and Future of the Language*.

Addison-Wesley. pp. 141 pp. ISBN 0-201-13433-0.

- Sammet, Jean E. (1969). *Programming languages: History and fundamentals*. Englewood Cliffs, N.J.: Prentice-Hall.
- The Encyclopedia of Computer Languages. *BASIC — Beginners All-purpose Symbolic Instruction Code* (http://hopl.murdoch.edu.au/showlanguage.prx?exp=176) . Murdoch University.

External links

- BASIC (http://www.dmoz.org/Computers/Programming/Languages/BASIC/) at the Open Directory Project
- Basic Computer Games (http://www.moorecad.com/classicbasic/index.html) by David Ahl
- More Basic Computer Games (http://www.atariarchives.org/morebasicgames) by David Ahl
- Big Computer Games (http://www.atariarchives.org/bigcomputergames) by David Ahl

Retrieved from "http://en.wikipedia.org/w/index.php?title=BASIC&oldid=469157911"

Categories: BASIC programming language family | Programming languages created in 1964 | American inventions

- This page was last modified on 2 January 2012 at 17:39.
-