

History of BASIC

History of the BASIC Programming Language by Andrea M. Marconi

The Birth of BASIC

The BASIC (Beginners All-Purpose Symbolic Instruction Code) programming language was born in 1964 at the Dartmouth College, New Hampshire (USA), where it was developed by John G. Kemeney (1926-93) and Thomas E. Kurtz (Kemeney, who had been working at the Manhattan Project (1945) and afterwards (1948-49) as Albert Einstein's assistant, met Kurtz at Dartmouth in 1956. At that time Kemeney was chairman of the Dept. of Mathematics and Kurtz just joined the faculty coming from Princeton University. The two started immediately working together to a new simplified programming language. Their efforts ended up with the Darsimco or Dartmouth Simplified Code which did not attract much interest. The Dartmouth Oversimplified Programming Experiment or DOPE which was their second experiment got even less attention. Nevertheless, they kept working on Kemeney's idea that a new language was necessary both for teaching and learning computer science to all the world of unexperienced users.

It is noteworthy to remember something about that era of computing. Computers were completely different than they are now; they were rather big, expensive and difficult to program. Moreover only a few of these machines were able to accept programs provided by the user and even in that case users needed to provide programs coded on small strips of paper or punched cards. In the latter case the whole procedure was more or less the following. A programmer with his program ready went to a punching center where, on the basis of the program, a series of cards were punched. Afterwards he would go to the computer room and place his punched card 'deck' on the queue. Once his punched cards reached its turn, the program was read by the computer, the instructions compiled and executed. It can be left to the imagination of the reader the case of a program with a bug. It is easy to imagine the whole approach to programming was, under those conditions, rather inefficient.

Kurtz, who had become director of Dartmouth's Kiewit Computational Center, realized all this and, strongly believing that every student should have had computing facility available, got the idea of what he called "Time Sharing System". This system also needed a new, simple and efficient language. Thus, while Kemeney was writing the programming language Kurtz was developing his Time Sharing System. They both recognized that the new language should meet the following specifications:

1) general purpose 2) easy of use 3) expandable 4) interactive 5) with debugging help 6) efficient 7) hardware independent 8) OS independent

To obtain all this they started from FORTRAN and ALGOL, taking from them the general structure and the English words. Program lines were numbered both because as such the flow of the program was unequivocally determined, and because it helped simplify editing (rows could be deleted just writing the line number or changed by writing a new line with the same number). The final version of what was called "Dartmouth BASIC" had 14 instructions. Limitations were present because programs should contain only one instruction per row and every row should start with a command, but the "Dartmouth BASIC" was compiled and rather fast for that time.

The efforts of Kemeney and Kurtz were repaid on May 1, 1964 when, at 4:00 am, two BASIC programs ran simultaneously on the General Electric 225 mainframe at Dartmouth College.

The growth of BASIC

Kemeney and Kurtz did not patent nor protect in any way their invention, they said they wanted

their language be in the public domain. This made possible the growth and differentiation of BASIC. Among the first users of BASIC was General Electric, which had sold the GE-225 to Dartmouth. The spread of BASIC dialects started around 1970 when G.E. released machines with the fifth version of BASIC instead of waiting for the new version of BASIC to come up. This happened one year later in 1971 when Kemeny and Kurtz released "BASIC the Sixth". Different BASIC versions were originated from Gordon Eubanks (The CEO and president of Symantec) who, in 1970, developed BASIC-E. BASIC-E used a technique similar to the one currently used by Java; instructions were transformed into a kind of intermediate code and then converted into machine readable code. Eubanks also did not protect BASIC-E but he did with his following version, CBASIC. CBASIC was marketed by his company, Compiler Systems (which in 1981 was acquired by Digital Research).

BASIC versions were so rapidly appearing that ANSI (American National Standard Institute) recognised in 1974 the need for a standard implementation of the BASIC language. A committee started working on two standard one called minimal BASIC and the other Standard BASIC. Standardization was completed long after the use of BASIC had spread worldwide, and Standard BASIC never appeared. ANSI issued two specifications: one for Minimal Basic in 1978 (specification X3.60-1978) and the other in 1987 for the Standard BASIC (specification X3.113-1987). These standards were also issued by the ISO (ISO 6373-1984 for the Minimal BASIC and ISO 10279-1991 for the Full BASIC). In the meanwhile (1975) Bob Albrecht and Dennis Allison created TinyBASIC which could run on 2K of RAM. In 1975 a different BASIC, created by William H. Gates III (1955-, better known as Bill) and Paul Allen (1953-), appeared on the MITS Altair. This was the first appearance of interpreted BASIC. Allen and Gates decided for an interpreter to overcome the limited amount of memory available, and, in fact, they were able to pack everything in 4K. A compiled language would not have left enough memory for both running the program and holding the data. The interpreted BASIC had another advantage, it was more interactive making debugging easier. Kemeny and Kurtz were very critical toward the interpreted language, despite this, some years later they acknowledged the positive effect interpreted BASIC had on the diffusion of BASIC.

During the same years many producers introduced a copy of BASIC on a ROM chip in their computers. The race had started. By the end of the 1970s home computers appeared with their own BASIC version: Radio Shack Level 1 BASIC (TRS 80), Apple Integer BASIC (Apple II, 1977), Timex Sinclair 1000 BASIC (Sinclair ZX80, 1980), Sinclair ZX81 BASIC (Sinclair ZX81, 1981), PET BASIC (Commodore PET, 1977), Atari BASIC (Atari 400/800, both 1978), Commodore BASIC (VIC 20 in 1981 & C64 in 1982), TI-BASIC (Texas TI-99), and so on.

The Evolution of BASIC

Since the early-1980s the history of BASIC and computing followed a tightly-bound and parallel course. IBM on the PC released an interpreted BASIC on ROM which could be enhanced by loading an additional extension known as BASICA. Essentially BASICA, which was on every PC-DOS distribution diskettes set, and its MS-DOS analog, the Gee-Witz (or GW) BASIC, dominated the market in the PC market. In 1984 Microsoft released the BASIC compiler (known as BASCOM from the .exe file) which again permitted compiled BASIC. This was a new breakpoint in BASIC's history. From that time on new powerful versions of BASIC appeared on the market every few months. Microsoft QuickBASIC, launched in 1985 as version 1.0, reached version 4.5 in 1988. In the meanwhile BASIC changed a lot, acquiring structured syntax, sub-functions, user defined data, multi-file programs and, in 1990 with the Microsoft BASIC Professional Development System 7.1, gained even full access to the memory outside 64K.

Again the operating system for computers drove the direction of BASIC when the Windows platform was introduced. This resulted in the development of Visual BASIC which was designed for GUI applications. Visual BASIC claims also to be an object-oriented language although many

programmers do not completely agree on this. They mean Visual Basic is only object based. Nevertheless, in the late 1990s, an estimate calculates that 90% of the software produced under Windows 9x was developed with Visual Basic.

Published on <http://www.q7basic.org/>
All rights reserved.