



# True BASIC is the Ideal First Step

by Thomas E. Kurtz

*Co-inventor of BASIC*

---

*Thomas E. Kurtz & the late John G. Kemeny invented BASIC in 1963 for use in their math and computer science courses at Dartmouth College, in Hanover, NH. 35 years later it remains the most widely used programming language in the world. Here Thomas E. Kurtz tells us why he thinks this is true:*

**Q: Why has BASIC remained so popular all these years?**

**A:** Because it's one of the few programming languages that allows you to write a small program quickly and get results right away. John Kemeny and I both understood that non-science students had to see results right away, else they would lose interest and go on to something else.

**Q: Is BASIC's popularity ebbing with the coming of C, C++ and Java?**

**A:** Just when this seems to be true, we find new evidence that shows that if people have a choice, they will use BASIC whenever possible. After all, C is a professional programmer's language, C++ is a variant of C, and Java is an extension of C++.

**Q: Why wouldn't programmers have a choice?**

**A:** For many years, 'conventional wisdom' claimed that BASIC was an obsolete solution. Many of the folks who made these claims had never used a 'modern' BASIC. In other cases when programming is done for hire, the group or individual commissioning the work often specifies the language they want used. They want their program written in the current 'fad language', many of which are no longer around. Professional programmers often select C, partly because it gives them lots to do; i.e., a form of job security.

**Q: What do you mean by a 'modern BASIC'?**

**A:** The runaway success of BASIC also led to problems. Since we had made our language freely available, many different versions proliferated. Frankly, I would like to disown many of those descendents. For example, forcing beginning programmers and students to know about integers and floating point numbers – Ugh, after all, a number is a number!

At Dartmouth we had a clear philosophy about the language that enabled us to build it from its modest beginnings into its role today as one of the most complete ‘all-purpose’ languages that has ever existed.

BASIC is an acronym for **B**eginner’s **A**ll-purpose **S**ymbolic **I**nstruction **C**ode. The first word of that acronym is still relevant today. No matter how powerful the language has become, we have never forgotten about the needs of beginners. It is interesting to note that, at Microsoft, they changed the name of their version of BASIC to Basic, because they didn’t want to stress the “beginners” part.



**Q: How did Dartmouth BASIC become True BASIC?**

**A:** Teaching at Dartmouth, John Kemeny and I were shielded from some of the worst implementations of BASIC. For example, we stopped using line numbers in 1975, just as personal computers were being invented. Dartmouth BASIC had continued to evolve into a more and more sophisticated language that was a joy to use. However, in 1983, three Dartmouth alums challenged us to look at the versions of BASIC that were out there, all different on the different computers. We were appalled at how terrible these crude ‘street’ versions were, and what high school and college students and teachers had to contend with.

We knew that writing papers or delivering talks would have little effect so we accepted the challenge of forming an independent commercial software publishing company and making available to everyone a high-quality BASIC, one that reflected our years of teaching experience.

**Q: How was your new Basic (or True BASIC) going to be different?**

**A:** We settled on eight important concepts: [1] It should be easy for the beginner. [2] It should be a general-purpose language, allowing for the writing of any type of program. [3] Advanced features had to be implemented in such a way that it was the ‘expert’ who paid the price, not the novice. [4] It should be a fully ‘interactive’ language, allowing the user to interact with the computer as quickly and easily as possible. [5] Error messages had to be clear and in English. [6] It should be fast and trouble free. [7] No understanding of computer hardware should be needed and it should shield the user from the mechanics of the operating system. [8] It was to be a portable solution with the same source code running on all popular computer operating systems.

**Q: How important is the portability feature now that almost 90% of the world uses Windows?**

**A:** More important than ever! What Windows are you talking about? Windows3, Windows95, Windows98, WindowsNT or Windows2000?

When we wrote the first versions of True BASIC no one had any idea what a great impact the Apple Macintosh was going to have on all future computer development. And at that

time the Amiga and Atari were popular and more advanced in many ways. But we all know now that that was not enough to insure their survival.

The architects of True BASIC were smart enough to know that we couldn't guess which operating systems would succeed and which would fail. So, from the beginning, they designed their Language System to be portable.

Today we continue to support all the popular operating systems. A new Linux edition of True BASIC is coming and that gives us unmatched coverage in the industry.

**Q: How can you justify supporting that many different systems? Why would you go to all that trouble?**

**A:** Publishing the world's best programming language is our mission. It is not a 'get-rich-quick' venture for us. Writing programs is hard work and continuing to make sure that True BASIC works on new operating systems is part of our charter. This insures that the work you put into writing True BASIC programs won't be lost as you switch from one operating system to another, or when we upgrade!



**Q: You have talked about True BASIC being both easy-to-use and sophisticated. How can you make this seemingly contradictory claim?**

**A:** Because we have been doing it for more than 15 years! Sometimes at trade shows someone will come to our booth and say, "True BASIC, huh. I didn't know BASIC was still being used."

I'll then ask them if they ever used True BASIC and they often reply something to the effect that, "No, BASIC is not powerful enough to accomplish what I need to do."

Then the fun begins. They describe a typical program need and in a few minutes, while we are still talking, I create a working example to show them how powerful True BASIC really is. It turns out they were still thinking of the first personal computer BASIC's, circa 1975-1980.

**Q: How are these advanced features implemented in True BASIC?**

**A:** With most versions we include sophisticated subroutine libraries. This allows you to call mini -programs within your programs that perform functions that might still be beyond your present understanding. In advanced versions we also give you the actual source code for most of these pre-written subroutines. Study these subroutines and you'll learn new concepts.

True BASIC has been a leader in every phase of modern BASIC development, including machine-independent graphics, powerful built-in matrix functions, libraries, modules and sound. And we've done this without destroying True BASIC's simplicity.

**Q: Why do you promote True BASIC as the ‘ideal first language’?**

**A:** Because this is what we hear again and again from teachers who are on the ‘frontlines’ of education.

We know that if a student gets results from a program they are writing, they are encouraged to continue. This is why more real learning takes place with True BASIC than with most other programming solutions.

The most important thing you need to learn as a programmer is to break down a task into a step-by-step process and plan for all eventualities. When you write a True BASIC program it is much like an easy-to-read script narrative of how to solve a problem.

Some of the newer programming languages divide code into event clumps and ‘hide’ portions behind the scenes. This makes it difficult for the programmer to really understand the full scope of the source code they have written.

With still other languages you have to learn so many ‘rules’ before you even start that it destroys any interest that a student might have.



**Q: So you contend that students will be helped, not hindered by making True BASIC their first programming language?**

**A:** It is not just me that makes this claim. The most outspoken boosters of True BASIC are persons who have learned many different programming languages.

This gives them a valid perspective from which to make a judgment. Over and over I hear from or talk to folks who have mastered Fortran, Cobol, LISP, Pascal, C, C++, and of course, True BASIC. This is what they tell me.

“When I need to test a concept or create a program in the shortest possible time, I always use True BASIC. It’s fast, forgiving, and very powerful. Often I never get to implement it with other solutions because it works just fine.”

**Q: If this is true, why isn’t True BASIC more widely known as the ‘best- first-step’?**

**A:** Programming languages are much like religion. People get very committed to a certain ‘belief’ and it is hard for them to consider anything else. However, we are gaining all the time. More and more people are using True BASIC and they make significant progress while others jump from one solution to another, wasting valuable time and resources and getting more confused all the time.

Teachers in schools and colleges don’t want to have to teach BASIC first, then language X. They seem to feel it’s more efficient to just teach language X. That’s true for the kids who don’t drop the course. We’ve heard of dropout rates as high as 50%.

I once taught the FIRST course in computer science at Dartmouth. It used Pascal. After the course I looked at the syllabus. I could have taught 90% of the course in True BASIC,

dropping only those things that Pascal has but True BASIC does not: pointers, allocated storage, and number types. All the rest could have been done with True BASIC.

Over the past thirty years I've seen many new programming languages come and go. Some introduced interesting new ways of doing things, but BASIC still stands as a standard of comparison. And, even as other Basics have come and gone, True BASIC continues to set the pace as a strong and very cost-effective programming solution.



**Q: What's ahead for True BASIC?**

**A:** We see some very exciting times ahead. Over the past five years we've made huge investments in R&D and have completely recreated True BASIC for the future.

However, I believe our most important future work is not only supplying a programming language, but also creating a wide variety of learning aids for students and teachers. That is why I'm very excited about the **True BASIC Institute**. It along with the Internet access to our continuing education materials give you access to a continuing source of how-to-program information.

The charter of the True BASIC Institute is to:

*'gather, prepare, and share programming knowledge and know-how with True BASIC users'*

In exchange for a moderately priced **annual membership fee**, you receive the wealth of materials and benefits shown on the next page. Three levels of membership are offered. One should be just right for your or your group.

Institute Membership even gives you e-mail access to me, so that I can respond to your programming questions.

I invite you to join the True BASIC Institute and take your programming skills to a new level.

Thomas E. Kurtz

*Co-inventor of BASIC and a founder of True BASIC Inc.*

**Enrollment Form**

*Learn at your own pace . . .*

with help from the creators of the world's best BASIC



## True BASIC Institute Benefits and Services:

### Access to our special Institute website section:

When you enter your membership number and password, you can download tutorial files and other information created especially for True BASIC Institute members. The instructional material is prepared in the Adobe Acrobat™ portable document format for quick downloads and easy reading.

### Summer & Winter Institute CD-ROM:

These special CD-ROMs for Institute members contain all the files and tutorials found at the website so that you can access the material at your convenience. In addition, the CDs contain selected program updates and special preview programs.

### Interactive documents & tutorials:

How-to tutorials and programming projects can be downloaded from the Institute website or from the Institute CDs. The Adobe Acrobat™ document format allows you to quickly find key words, copy code portions, print out information as needed, and link to special website locations.

### Free software updates:

While you are an active Institute member, **updates** to the True BASIC products you have registered are available to you FREE.

### Advance Previews:

See new True BASIC documents and books with an opportunity to send questions and comments to article authors.

### Special E-mail access to BASIC Inventor:

Dr. Thomas E. Kurtz, co-inventor of BASIC has agreed to interact with all Institute members by e-mail. Here is your opportunity to correspond with one of the pioneers of personal computing.

### Discounts on True BASIC purchases.

Institute members receive an additional **10% discount** on purchases from True BASIC.

## 3 Membership Plans:

**Individual Membership: \$50 per year** - Institute access for individual [1] user.

**Group Membership: \$125 per year** - Ideal for groups or small companies; access for [5] users.

**Site Membership: \$250 per year** - Larger groups and sites; access for [10] users.

Memberships are renewed each January, but you can join at any time. Pro-rata fees are:

Enrollment Month:	Jan-Apr	May - Aug	Sept-Dec*
Individual	\$50	\$34	\$67*
Group	\$125	\$85	\$160*
Site	\$250	\$170	\$335*

\* New or renewal memberships in Sept-Dec also include a full membership for the following year.

### Enrollment Form



## Ten Reasons Why True BASIC is Your BEST Choice

- H **It is easy-to-learn!** True BASIC enables you to create your first functioning program within minutes after you begin your first session.
- H **It is widely used.** More people program in BASIC. New programming languages continue to be introduced, but it is estimated that more than 75% of all programs now in use are written in BASIC. .
- H **True BASIC is a modern language.** True BASIC runs both 'classic' BASIC programs with line numbers, **and** fully structured code. It also comes with a wealth of pre-written subroutine libraries.
- H **True BASIC is not a fad language.** True BASIC has been available since 1984. The new versions continue to run programs that were written previously.
- H **True BASIC is portable.** Programs written in True BASIC can be run, without change, on more operating systems than any other programming language. True BASIC runs on more operating systems than any other programming language.
- H **True BASIC is an ideal teaching language.** More than 8,600 schools and colleges use True BASIC in their classes. It is widely used in every area of education, science, and business.
- H **True BASIC has sophisticated features.** True BASIC gives you a unique combination of simplicity and power in one easy-to-use package. It has pioneered every phase of modern BASIC development, including machine-independent graphics, powerful built-in matrix functions, libraries, modules, as well as functions for sound and animation.
- H **True BASIC programs are an easy-to-read script-like narrative.** You get a full sense of the program you are writing. Some programming languages now divide code into events clumps, and 'hide' portions behind the scenes. This makes it more difficult for you to fully comprehend the full scope of your source code.
- H **You do not need to become a computer expert.** True BASIC allows you to spend more time programming and less time worrying about the mechanics of operating systems. With True BASIC you never have to worry about memory management issues or scores of other operating system primitives.
- H **True BASIC is cost effective.** With True BASIC you can working develop programs quickly. Programs written by one person can be understood by others who might have to maintain the code in the future. As operating systems change, you can continue to use the same programs.