

1: The Best Way to Computer Programming for Beginners - wikiHow

Computer Programming in Basic the Easy Way: An Introduction to Computer Programming (Barron's Educational Series). Net proceeds from the sale of these goods and.

Origin[edit] John G. Kemeny was the math department chairman at Dartmouth College, and largely on his reputation as an innovator in math teaching, in they won an Alfred P. Kurtz had joined the department in , and from the s they agreed on the need for programming literacy among students outside the traditional STEM fields. It was as simple as that. These did not progress past a single freshman class. As Kurtz noted, Fortran had numerous oddly-formed commands, notably an "almost impossible-to-memorize convention for specifying a loop: Kurtz 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. However, the syntax was changed wherever it could be improved. These changes made the language much less idiosyncratic while still having an overall structure and feel similar to the original FORTRAN. Mary Kenneth Keller , one of the first people in the U. Wanting use of the language to become widespread, its designers made the compiler available free of charge. In the s, 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 put considerable effort into promoting the language. Spread on minicomputers[edit] "Train Basic every day! A version was a core part of the Pick operating system from onward, where a compiler renders it into bytecode , able to be interpreted by a virtual machine. Ahl and published in a newsletter he compiled. When management refused to support the concept, Ahl left DEC in to found the seminal computer magazine, Creative Computing. The book remained popular, and was re-published on several occasions. It had the advantage that it was fairly well known to the young designers and computer hobbyists who took an interest in microcomputers. Running Light Without Overbyte. The Apple II and TRS 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. 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. This book, and its sequels, provided hundreds of ready-to-go programs that could be easily converted to practically any BASIC-running platform. Turbo Pascal -publisher Borland published Turbo Basic 1. 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 s, 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. 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". 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. While many advanced programmers still scoffed at its use, VB met the needs of small businesses efficiently wherever ease of development was more of a concern than processing speed. 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. The latter has essentially the same power as C and Java but with syntax

that reflects the original Basic language. Three modern Basic variants: 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. 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. Excel 4 and 5 use Visual Basic itself as a macro language. Nostalgia[edit] 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 Salon article [19] as have others who first used computers during this era. In turn, the article prompted Microsoft to develop and release Small Basic. A short documentary film was produced for the event.

2: 5 Ways You can Learn Programming Faster - www.enganchecubano.com

Programming Languages Enter your mobile number or email address below and we'll send you a link to download the free Kindle App. Then you can start reading Kindle books on your smartphone, tablet, or computer - no Kindle device required.

There are lots of things you can do to make it easier on yourself when you are learning to program. Now, discover how to get the most out of your learning. One common theme across many of these tips is: When I was teaching C, there were always a few students who came into the class knowing a bit about programming. Inevitably, some of these students did great in the first few weeks only to fall further and further behind as the course went on. They went too fast through the introductory part of the course, thinking they knew it all--but they rarely did. They knew some of the material, but not enough to have a strong grasp of the fundamentals. At the same time, you must not stop making progress--you can go too slow as well as too fast. Look at the Example Code Reading is usually about the words on the page, but learning to program is about code. When I first learned to program, I would sometimes read the code examples before the text, and try to figure out what they did. This tutorial spends a great deal of time talking about the sample code to help you work through exactly what the code does. Then type the sample code into a compiler--if you type it, instead of copying and pasting it, you will really force yourself to go through everything that is there. Typing the code will force you to pay attention to the details of the syntax of the language--things like those funny semicolons that seem to go after every line. Then compile it and run it. Make sure it does what you think it does. Software is the most easily changed machinery on the planet. You can experiment easily, try new things, see what happens; the changes will happen almost immediately, and there is no risk of death or mayhem. The easiest way to learn new language features is to take some code that works one way, and change it. You can find some programming challenges on this site. You can also reimplement the examples from the book or tutorial you are reading. This technique can work especially well if you tweak the sample code. Whether you use them later or not, you will get the same useful experience. But it bears repeating; the sooner you learn good debugging techniques, easier it will be to learn to program. The first step in doing so is to learn how to use a tool called a debugger , which allows you to step through your code. A debugger will allow you to step line by line through a piece of code. It will let you see the values of variables, and whether the code inside an if statement is executed. A debugger can help you quickly answer questions about what your code is doing. After the tenth or so bug, it will really start to pay off. And believe me, you will have way more than ten bugs in your programming career. I often saw students unwilling to use a debugger. These students really made life hard on themselves, taking ages to find very simple bugs. The sooner you learn to use a debugger, the sooner it will pay off. First, look for alternative explanations. The internet is filled with information about programming, and some explanations work better for different people; you might need pictures, someone else might not. There are also lots of good books with detailed explanations. Instead, rephrase your understanding of the text in your words. The more your question reveals about what you are thinking, the easier it will be for a knowledgeable expert to answer it. Programmers sometimes have a reputation for being grumpy about answering questions, but I think the reason is that they want to make progress in a conversation, and that requires both sides to put in effort. If you ask a smart, detailed question that shows you are thinking, you will generally get good results. There are plenty of places you can go to ask questions. You can always email me , or post on our message board , or ask an expert. Stay tuned, by subscribing to our RSS feed , signing up for email notifications , or following alexallain on twitter.

3: Computer Programming

*Computer Programming in Basic the Easy Way: An Introduction to Computer Programming (Barron's Educational Series) [Douglas Downing] on www.enganchecubano.com *FREE* shipping on qualifying offers. Discusses the fundamental techniques of computer programming and supplies step-by-step instructions on programming computers in the BASIC language.*

Let us look at each of these categories. Machine Language Humans do not like to deal in numbers alone—they prefer letters and words. But, strictly speaking, numbers are what machine language is. This lowest level of language, machine language, represents data and program instructions as 1s and 0s—binary digits corresponding to the on and off electrical states in the computer. Each type of computer has its own machine language. In the early days of computing, programmers had rudimentary systems for combining numbers to represent instructions such as add and compare. The computer industry quickly moved to develop assembly languages. Assembly Languages Figure 2: Example Assembly Language Program Today, assembly languages are considered very low level—that is, they are not as convenient for people to use as more recent languages. At the time they were developed, however, they were considered a great leap forward. To replace the 1s and 0s used in machine language, assembly languages use mnemonic codes, abbreviations that are easy to remember. Although these codes are not English words, they are still— from the standpoint of human convenience—preferable to numbers 0s and 1s alone. The programmer who uses an assembly language requires a translator to convert the assembly language program into machine language. A translator is needed because machine language is the only language the computer can actually execute. The translator is an assembler program, also referred to as an assembler. It takes the programs written in assembly language and turns them into machine language. Programmers need not worry about the translating aspect; they need only write programs in assembly language. The translation is taken care of by the assembler. Although assembly languages represent a step forward, they still have many disadvantages. A key disadvantage is that assembly language is detailed in the extreme, making assembly programming repetitive, tedious, and error prone. This drawback is apparent in the program in Figure 2. Assembly language may be easier to read than machine language, but it is still tedious. High-Level Languages The first widespread use of high-level languages in the early 1950s transformed programming into something quite different from what it had been. Programs were written in an English-like manner, thus making them more convenient to use. As a result, a programmer could accomplish more with less effort, and programs could now direct much more complex tasks. These so-called third-generation languages spurred the great increase in data processing that characterized the 1950s and 1960s. During that time the number of mainframes in use increased from hundreds to tens of thousands. The impact of third-generation languages on our society has been enormous. Of course, a translator is needed to translate the symbolic statements of a high-level language into computer-executable machine language; this translator is usually a compiler. There are many compilers for each language and one for each type of computer. Keep in mind, however, that even though a given program would be compiled to different machine language versions on different machines, the source program itself—the COBOL version—can be essentially identical on each machine. Some languages are created to serve a specific purpose, such as controlling industrial robots or creating graphics. Many languages, however, are extraordinarily flexible and are considered to be general-purpose. In addition to these three, another popular high-level language is C, which we will discuss later. Very High-Level Languages Languages called very high-level languages are often known by their generation number, that is, they are called fourth-generation languages or, more simply, 4GLs. Definition Will the real fourth-generation languages please stand up? There is no consensus about what constitutes a fourth-generation language. The 4GLs are essentially shorthand programming languages. An operation that requires hundreds of lines in a third-generation language such as COBOL typically requires only five to ten lines in a 4GL. However, beyond the basic criterion of conciseness, 4GLs are difficult to describe. Characteristics Fourth-generation languages share some characteristics. The first is that they make a true break with the prior generation—they are basically non-procedural. A procedural language tells the computer how a

task is done: Add this, compare that, do this if something is true, and so forth—a very specific step-by-step process. The first three generations of languages are all procedural. In a nonprocedural language, the concept changes. Here, users define only what they want the computer to do; the user does not provide the details of just how it is to be done. Obviously, it is a lot easier and faster just to say what you want rather than how to get it. This leads us to the issue of productivity, a key characteristic of fourth-generation languages. Productivity Folklore has it that fourth-generation languages can improve productivity by a factor of 5 to 10. The folklore is true. Most experts say the average improvement factor is about that is, you can be ten times more productive in a fourth-generation language than in a third-generation language. Produce a report showing the total units sold for each product, by customer, in each month and year, and with a subtotal for each customer. In addition, each new customer must start on a new page. A 4GL request looks something like this: The third-generation language COBOL, however, typically requires over 100 statements to fulfill the same request. If we define productivity as producing equivalent results in less time, then fourth-generation languages clearly increase productivity. Downside Fourth-generation languages are not all peaches and cream and productivity. The 4GLs are still evolving, and that which is still evolving cannot be fully defined or standardized. What is more, since many 4GLs are easy to use, they attract a large number of new users, who may then overcrowd the computer system. One of the main criticisms is that the new languages lack the necessary control and flexibility when it comes to planning how you want the output to look. A common perception of 4GLs is that they do not make efficient use of machine resources; however, the benefits of getting a program finished more quickly can far outweigh the extra costs of running it. Benefits Fourth-generation languages are beneficial because They are results-oriented; they emphasize what instead of how. They improve productivity because programs are easy to write and change. They can be used with a minimum of training by both programmers and nonprogrammers. They shield users from needing an awareness of hardware and program structure. It was not long ago that few people believed that 4GLs would ever be able to replace third-generation languages. These 4GL languages are being used, but in a very limited way. Query Languages A variation on fourth-generation languages are query languages, which can be used to retrieve information from databases. Data is usually added to databases according to a plan, and planned reports may also be produced. But what about a user who needs an unscheduled report or a report that differs somehow from the standard reports? A user can learn a query language fairly easily and then be able to input a request and receive the resulting report right on his or her own terminal or personal computer. A standardized query language, which can be used with several different commercial database programs, is Structured Query Language, popularly known as SQL. Natural Languages The word "natural" has become almost as popular in computing circles as it has in the supermarket. Fifth-generation languages are, as you may guess, even more ill-defined than fourth-generation languages. They are most often called natural languages because of their resemblance to the "natural" spoken English language. And, to the manager new to computers for whom these languages are now aimed, natural means human-like. Instead of being forced to key correct commands and data names in correct order, a manager tells the computer what to do by keying in his or her own words. Example of Natural Language Interaction A manager can say the same thing any number of ways. For example, "Get me tennis racket sales for January" works just as well as "I want January tennis racket revenues. The natural language translates human instructions—bad grammar, slang, and all—into code the computer understands. If it is not sure what the user has in mind, it politely asks for further explanation. Natural languages are sometimes referred to as knowledge-based languages, because natural languages are used to interact with a base of knowledge on some subject. The use of a natural language to access a knowledge base is called a knowledge-based system. Consider this request that could be given in the 4GL Focus: But some natural languages can handle such a request. Users can relax the structure of their requests and increase the freedom of their interaction with the data. Here is a typical natural language request: You can hardly get closer to conversational English than that. An example of a natural language is shown in Figure 3. Natural languages excel at easy data access. Indeed, the most common application for natural languages is interacting with databases. Choosing a Language How do you choose the language with which to write your program? There are several possibilities: In a work environment, your manager may decree that everyone on your project will use a certain language. You may

use a certain language, particularly in a business environment, based on the need to interface with other programs; if two programs are to work together, it is easiest if they are written in the same language. You may choose a language based on its suitability for the task. For example, a business program that handles large files may be best written in the business language COBOL. If a program is to be run on different computers, it must be written in a language that is portable-suitable on each type of computer-so that the program need be written only once.

4: C, C++ Programming Tutorials - www.enganchecubano.com

Get this from a library! Computer programming in BASIC the easy way: an introduction to computer programming. [Douglas Downing] -- Explains the principles of computer programming in BASIC and includes exercises with answers.

Are you looking for an easy-to-learn programming language? Well, while the answer to your question varies according to your need, there are some programming languages that a beginner can learn and excel. Python and Ruby are well established as easiest programming languages for beginners due to their simple and readable syntax. These factors are salary, popularity, job opportunities, and geography. Such languages are easier to learn and pick up as compared to others. Everybody has different needs and inspirations behind learning. So, read about these five top languages for beginners and make your choice accordingly.

5 Easy programming languages for beginners: Python

These days, Python is regarded as one of the best and easiest programming languages for beginners, and it gets mentioned very soon in any coding discussion. Guido van Rossum developed Python in s. With its simple and readable code, the programmers can express the concept in lesser lines of code. Today, it has become one of the most popular introductory programming languages in American and European schools. Python is used to create desktop and web apps. This language supports multiple programming paradigms, including object-oriented, functional, and imperative. This feature makes it a very flexible programming language. If you already know Ruby, you might be knowing that without any need to learn tons of commands and jargon, you can get started. Just like Python, Ruby also enjoys the love of its dedicated community. Ruby acts as a great tool if you need to deal with the backend aspect of your work. Ruby on Rails is a web framework built on Ruby. As said above, Ruby on Rails is used to create web applications. Within a few minutes, one can create a web blog. Over the time, it has established itself as one of the most influential and solid programming languages. But, it allows a learner to think like a programmer and understand how a PC processes information. Learning Java ensures a solid foundation and makes you ready for a coding career. Java is used to build native Android applications. It has established itself as one of the most important programming languages around. These languages have been used to build numerous pieces of software, operating systems, desktop apps, mobile apps, web apps, game, game engines, etc. JavaScript JavaScript is often, and rightly, called the language of the web. But, for an introduction to the world of programming, JS is a good choice. JavaScript, one of the greatest and easiest programming language for beginners, is used often through libraries like JQuery and frameworks like React and AngularJS. JavaScript is also used for mobile app development and web apps.

Easiest Programming Languages for Kids:

5: 5 Easy-to-learn Programming Languages For Beginners

Note: Citations are based on reference standards. However, formatting rules can vary widely between applications and fields of interest or study. The specific requirements or preferences of your reviewing publisher, classroom teacher, institution or organization should be applied.

6: How to Learn a Programming Language (with Pictures) - wikiHow

There is a very good programming language that is very easy to use for the beginners. This is also known as user-friendly. And this is the VB (Visual Basic language), try this it is really amazing.

7: BASIC - Wikipedia

A very EASY and FREE way for ANYONE to write their first computer program in TEN MINUTES. Note: This instructable is for people that think that programming is some sort of magical thing that you need expensive programs or high tech skills to do. Hopefully this instructable will remove the veil from.

Learning impairment vs learning disability Fuck buddies J.R. Mavrick Developing C language portable system call libraries Petroleum and Marine Technology Information Guide Oxford-Duden pictorial Spanish-English dictionary. Blueprints ob gyn 7th edition The two Rs : a study of current early reading and writing practice in Irish primary schools Brian Murphy Three articles from Nouveau Mercure, 1717-1719. Part 2. The reality: so little progress Phase 4: Adulthood (using personas) Journal Of Discourses By Brigham Young, His Two Counsellors, The Twelve Apostles, And Others V3 The doctors gift September 1 1939 analysis American Clocks for Collectors Pictures in patchwork The Straight Pool Bible You Cant Tell A Book By Its Cover Russias economy of favours Shellcoders handbook filetype Learn Spanish Together (LL(R Learn Together) List of things needed for new baby The look of Buster Keaton Prayer and healing: reviewing the research Immigrants and participation beyond the nation-state: opportunity-capability rift in EU immigration polic Narratives of the Educative Surround, John Goodlad Last gas station and other stories Data Structures and Abstract Data Types Holidays/family traditions/reunions Doing business gods way Staad pro tutorial Stedmans oncology words Discovering Computers 2002 Concepts for a Digital World, Web Enhanced, Introductory The living beauty philosophy C sharp basics Microsoft Outlook 97 Cause of hard times Workbook physics for scientists and engineers 12 Rabies (Deadly Diseases and Epidemics) Arresting Addictions Exposition of a new theory on the measurment of risk Specimen theoriae novae de mensura sortis [by Daniel Duralay International Holdings Ltd and Gates Consumer Industrial (part of the Tomkins PLC group)