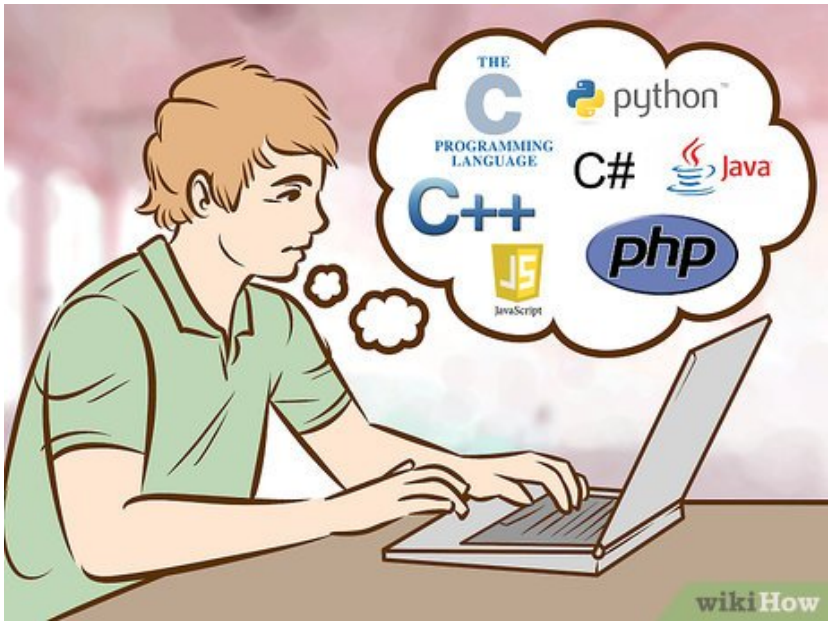
How to Start Learning Computer Programming

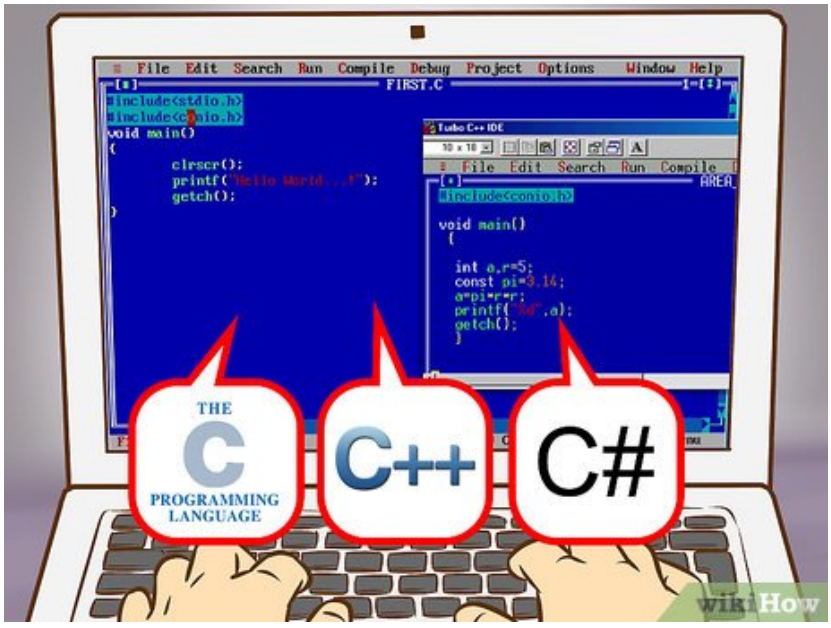
Programming is lots of fun and extraordinarily useful. It allows you to be creative and also opens up a wide range of new careers for you. If you want to learn how to program, read the tutorial below for an explanation of where to go and...

Part 1 of 3:

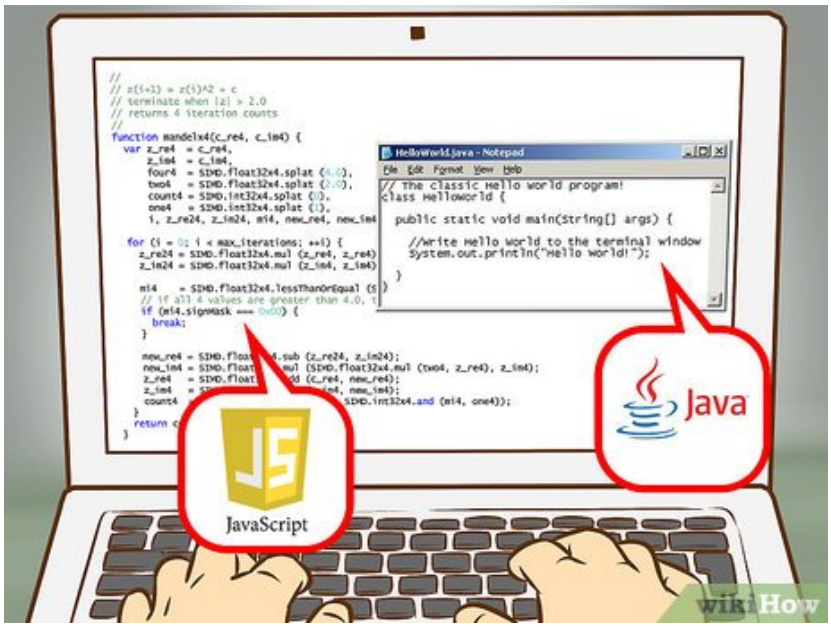
Choosing a Language

1. 

Choose a programming language. Computer programming is done as essentially a set of written instructions that the computer follows (also known as binary coding). These instructions can be written in several different "languages", or which are simply different ways of organizing the instructions and text. Different languages tend to be used to create different types of programs, however, so choose a language that you feel is relevant to what you want to do.[1] If you decide that a language does not suit your needs, you can always move on to a new language.

2. 

Consider C, C++, C# and related languages. These languages are mainly used for creating standalone computer applications such as games. C and C++ are difficult languages to learn for a beginner, but not impossible. Learning them will give you an in-depth understanding of not only programming (most programming languages inherit some concept or the other from C and C++) but also of how a computer works. They are popular and widely used, though C#, a language very similar to Java, is starting to become much more common.[2]

3. 

Consider Java or JavaScript**.** These are good languages to learn if you want to work on making web plugins (JavaScript) or mobile apps (Java). These languages are very much in demand right now, so they are handy to know. Keep in mind that Java and JavaScript are completely different languages, despite the similarity in names.[3]

4. 

Try Python. Python is a very versatile language used widely across several platforms.[4] Despite being extremely powerful, it is an easy language for a beginner to pick up, so give it a try!

5. 

Consider PHP. PHP stands for PHP: Hypertext Processor. It is a web programming language and relatively easy to learn due to its weak typing and popularity (popularity means there will be several useful tutorials on the language). It is a great language for server-side programming.[5]

6. 

Don't limit yourself to these languages! There are tons of programming languages, all with varying uses. If you want to work as a programmer, you will need to know more than one, so learn as many as you can.

1. Your best bet will be to look at ads for the sort of jobs you want to get and look for the common languages that they ask for.

Part 2 of 3:

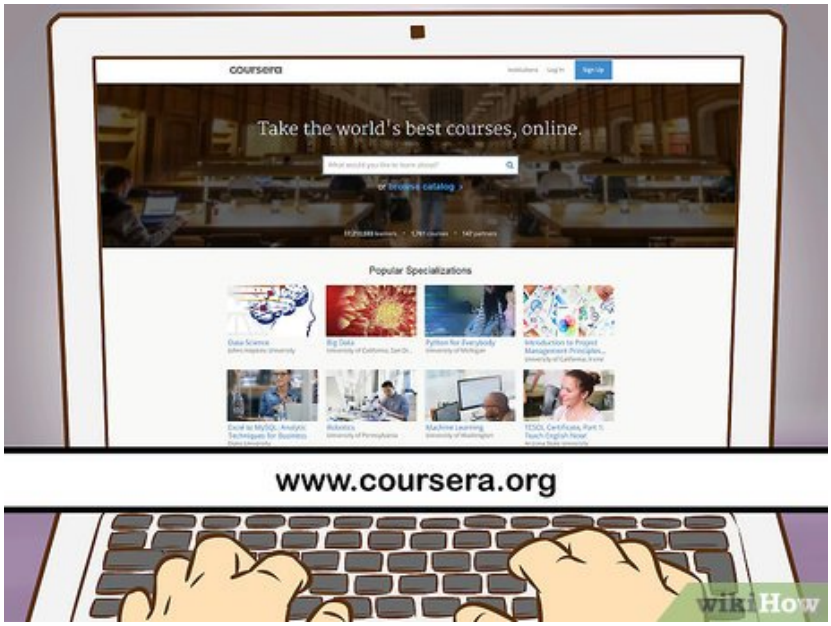
Learning the Language

1. 

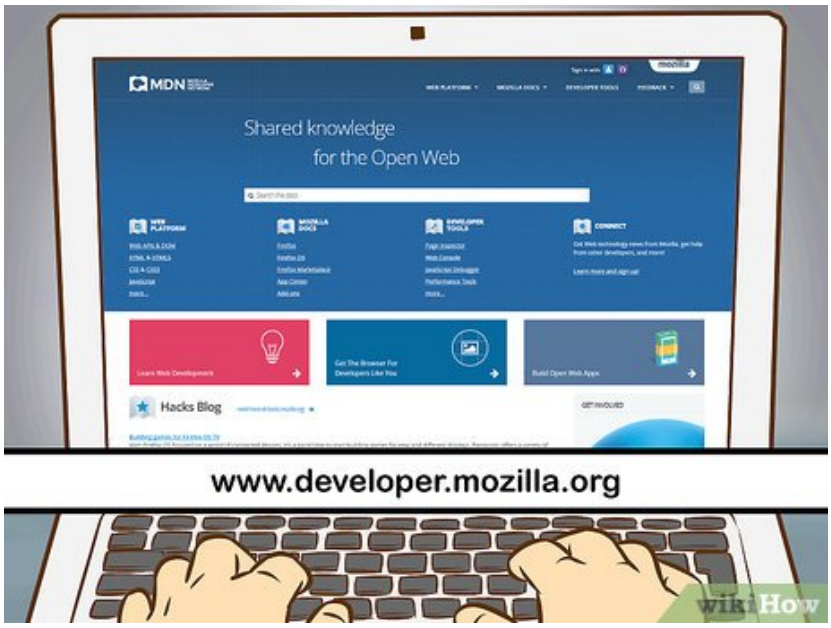
Think about going to school. While most companies hiring a programmer will care more about your skills than the college you went to or your grades, it greatly helps to have a college degree to point to. You will learn more efficiently than if you teach yourself, all while getting expert guidance from your teachers

(and maybe your friends).

1. There are often scholarships and grants available to those doing degrees in this field. Don't feel intimidated by the price tag of a degree: it is possible!

2. 

Learn from online universities. Whether you do an online degree with fees and an actual degree at the end or you're attending a free program like MIT's wonderful Coursera, you can learn a lot about programming from these structured courses.

3. 

Try using online tools. Use free services like Google's University Consortium or Mozilla's Developer Network to learn more about programming. These companies want more developers to help their platforms flourish and their resources can be some of the best on the web.

4. 

Learn using online tutorials. There are loads of programmers with websites where they will teach you the individual basics, as well as a few tricks. Look up tutorials on the language you want to learn to find these.

1. Many free online classes are available to learn to code from. The Khan Academy teaches computer coding, with easy tutorials and videos. Codecademy is another free site to learn from, with step-by-step tutorials.

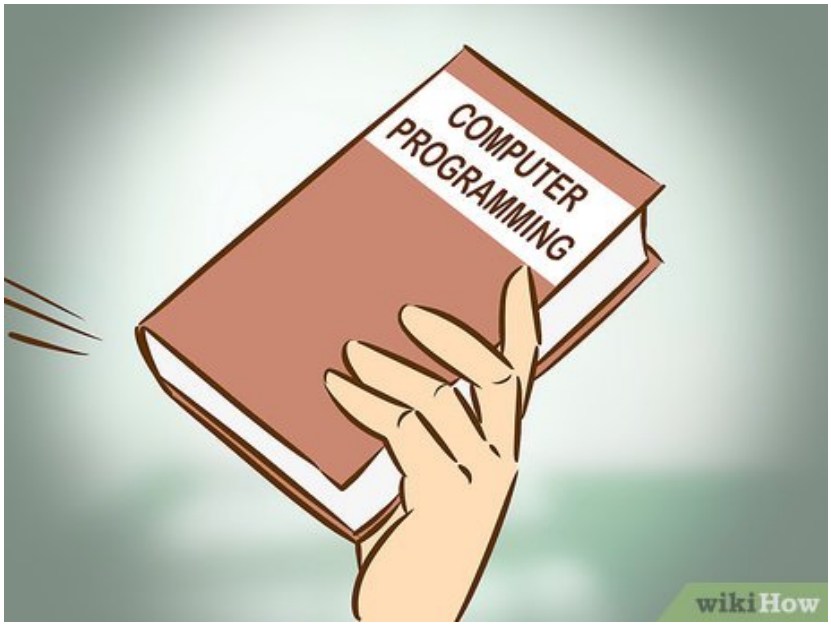
5. 

Start young if you can. There are several programs designed to teach kids to program. Programs like MIT's Scratch are very helpful and the younger you are, the easier it will be to pick up (like any language).

1. Avoid kits, as these rarely teach anything useful.

Part 3 of 3:

Teaching Yourself

1. 

Start with a good book or tutorial on programming. Get a good, current book on the programming language you want to learn. Reviews on Amazon or similar sites will usually help you identify helpful books from unhelpful ones.

2. 

Get an interpreter for that language. An interpreter is just another computer program but it will convert ideas you've written in a programming language into "machine code" so you can see things work. Lots of programs are available and you will need to choose one that is appropriate for you.[6]

3. 

Read the book! Take examples of the programming language from the book and put them into your interpreter. Try changing the examples and making the program do different things.

4. 

Try putting together your ideas to form a working program. Start with simple things, like a program to convert currencies, and work your way up to more complex things as you continue reading and learning about your programming language.

5. 

Learn another language. Once you start actively programming in your first language, you may want to learn a second one. You'll get the most out of learning a second programming language if you pick one that uses a radically different paradigm than the one you started with. For instance, if you started in Scheme, you might try learning C or Java next. If you started in Java, you could learn Perl or Python.

6. 

Continue programming and trying new things! To be a good programmer, you, at the very least, have to keep up with changing technology. It's a constant learning process, and you should always be learning new languages, new paradigms, and most importantly: programming new things!

1. Being a successful programmer means learning to think like one. You'll need to look at challenges as learning opportunities, desire to improve your skills and be open to new ways of improving your programming process.

articles on tips and guides. Thank you for reading and for following us regularly.