Scripting and breakpointing is also part of this process.  
Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years.  
To produce machine code, the source code must either be compiled or transpiled.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Integrated development environments (IDEs) aim to integrate all such help.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
Scripting and breakpointing is also part of this process.  
The purpose of programming is to find a sequence of instructions that will automate the performance of a task (which can be as complex as an operating system) on a computer, often for solving a given problem.  
There are many approaches to the Software development process.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Many applications use a mix of several languages in their construction and use.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 Different programming languages support different styles of programming (called programming paradigms).  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference.