By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers..  
 Following a consistent programming style often helps readability.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
 Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL).  
Scripting and breakpointing is also part of this process.  
However, readability is more than just programming style.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.