Programmable devices have existed for centuries..  
 Following a consistent programming style often helps readability.  
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.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Scripting and breakpointing is also part of this process.  
There exist a lot of different approaches for each of those tasks.  
There are many approaches to the Software development process.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.