Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Integrated development environments (IDEs) aim to integrate all such help.  
There exist a lot of different approaches for each of those tasks.  
Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
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.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Programming languages are essential for software development.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
 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.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
Techniques like Code refactoring can enhance readability.