**History**  
  
Prolog has an exciting and diverse history that spans several decades. Its origins can be traced back to the late 1960s and early 1970s, when researchers, Alain Colmerauer and Phillipe Roussel, at the University of Aix-Marseille, collaborated with Robert Kowalski of the University of Edinburgh to design the underlying framework of Prolog. Kowalski provided the theoretical foundation of Prolog, while Colmerauer's research helped formalize the Prolog language.

1972 is widely regarded as the birthdate of Prolog, and since then, it has branched out into several different dialects. The first Prolog interpreter was built by Phillipe Roussel, and the first compiler was developed by David Warren of the University of Edinburgh. Prolog's popularity continued growing in North America and Europe, mainly through its use in the European Esprit program. Its use in the Japanese Fifth-Generation Computer Project in 1981 brought considerable attention to the language and its capabilities.

Prolog's heritage is also rooted in the research on automated deduction systems and theorem provers that were developed in the 1960s and 1970s. The inference mechanism of Prolog is based on Robinson's Resolution Principle, which was proposed in 1965, and the answer-extracting mechanism by Green in 1968. These ideas were combined with linear resolution procedures, which gave rise to the development of a general-purpose logic programming system.

Today, Prolog remains a widely used programming language that has made significant contributions to the field of artificial intelligence and automated reasoning.