



Explore

**APL** (named after the book *A Programming Language*) is a programming language developed in the 1960s by **Kenneth E. Iverson**. [It revolves around **multidimensional arrays** and employs a unique set of **graphic symbols** to represent functions and operators, resulting in remarkably concise code1](https://en.wikipedia.org/wiki/APL_%28programming_language%29). If you’re intrigued by this fascinating language, here are some resources to explore:

1. [**TryAPL**](https://tryapl.org/): Dive into APL with interactive examples and exercises. [It will change the way you think about problems and data](https://en.wikipedia.org/wiki/APL_%28programming_language%29)[2](https://tryapl.org/).
2. [**Learning APL**](https://xpqz.github.io/learnapl/intro.html): This comprehensive tutorial introduces APL’s curious symbols, high-level abstraction, and ultra-concise formulations of algorithms. [It’s perfect for experienced programmers from other languages](https://en.wikipedia.org/wiki/APL_%28programming_language%29)[3](https://xpqz.github.io/learnapl/intro.html).
3. [**MicroAPL APL Tutorial**](https://www.microapl.com/apl/tutorial_contents.html): Explore topics like variables, functions, tables, and error handling in this detailed tutorial[4](https://www.microapl.com/apl/tutorial_contents.html).
4. [**Exploring APL High-Level Programming**](https://marketsplash.com/tutorials/apl/apl-high-level-programming/): Delve deeper into APL’s functional programming, custom functions, and operator usage[5](https://marketsplash.com/tutorials/apl/apl-high-level-programming/).
5. **APL Wiki**: A collaborative platform where you can find additional resources, discussions, and community insights.

Remember, learning APL isn’t just about job prospects—it’s about expanding your problem-solving mindset and experiencing a different way of approaching programming challenges. Happy coding! 🚀🍌