A **Trie** is a tree-based data structure used for efficient storage and retrieval of strings, particularly suited for tasks like autocomplete, spell checking, and searching words in lengthy texts. [It follows the property that if two strings share a common prefix, they have the same ancestor in the trie1](https://www.geeksforgeeks.org/introduction-to-trie-data-structure-and-algorithm-tutorials/).

Here are **five free reference links** where you can learn more about Tries:

1. [**GeeksforGeeks**: Provides an **introduction to Trie**, along with practical examples and implementation in various programming languages like C++, Java, Python, and C#](https://www.geeksforgeeks.org/introduction-to-trie-data-structure-and-algorithm-tutorials/) [1](https://www.geeksforgeeks.org/introduction-to-trie-data-structure-and-algorithm-tutorials/).
2. [**Codecademy**: Offers a tutorial on **tries and binary indexed trees**, explaining their use for efficient search implementations](https://www.geeksforgeeks.org/introduction-to-trie-data-structure-and-algorithm-tutorials/) [2](https://www.codecademy.com/article/introduction-to-tries).
3. [**Devsenv**: Presents an **introduction to Trie data structure** with practical examples, covering its applications and advantages](https://www.geeksforgeeks.org/introduction-to-trie-data-structure-and-algorithm-tutorials/) [3](https://devsenv.com/tutorials/trie).
4. [**TutorialsPoint**: Discusses the **basic operations in Tries**, including standard, compressed, and suffix tries, along with real-world applications](https://www.geeksforgeeks.org/introduction-to-trie-data-structure-and-algorithm-tutorials/) [4](https://www.tutorialspoint.com/data_structures_algorithms/tries.htm).
5. [**Wikipedia**: Provides an overview of Tries, their applications in predictive text, approximate matching algorithms, and more](https://www.geeksforgeeks.org/introduction-to-trie-data-structure-and-algorithm-tutorials/) [5](https://en.wikipedia.org/wiki/Trie).

Feel free to explore these resources to deepen your understanding of Trie data structures! 🌟