Differences between JSON and XML

Feature	JSON (JavaScript Object Notation)	XML (eXtensible Markup Language)
Syntax	Lightweight, human-readable, uses key-value pairs	More verbose, uses nested tags with attributes and values
Data Structure	Represents objects as collections of key-value pairs	Represents data in a tree structure with nested elements
Type of Data	Supports data types like strings, numbers, arrays, objects, booleans, and null	Primarily string-based, supports attributes, elements, text, and mixed content
Ease of Use	Easier to read and write, less verbose	More complex due to its verbose nature and mixed content model
Schema Definition	Schema definitions are less formal, often implied (JSON Schema available)	Supports formal schemas like DTD, XSD, and Relax NG
Namespaces	Does not support namespaces	Supports namespaces to avoid element name conflicts
Parsing	Generally faster and easier to parse	Slower and more resource-intensive to parse due to its complexity
Support for Comments	Does not natively support comments	Supports comments within the data using `<!-- -->` tags
Extensibility	Less flexible in terms of extensibility	Highly extensible, can define custom tags and attributes
Data Interchange	Well-suited for data interchange between applications, especially web services	Also used for data interchange, but more common in document storage and complex data structures
Error Handling	Less strict, may fail silently or with minimal error messages	More strict, provides detailed error messages for validation and parsing errors
Use Cases	Widely used in web APIs, configuration files, and data interchange between servers and web applications	Used in a variety of applications, including web services (SOAP), document storage, and configuration files
Tooling and Support	Strong support in modern programming languages and tools	Also well-supported, particularly in enterprise and legacy systems