In JavaScript, backticks (`) are used to define template literals, a powerful way to create strings with enhanced features:

Key Features of Template Literals:

1. **String Interpolation:**
   - ○ Embed expressions directly within strings for dynamic content.
   - ○ Use the syntax ${expression} to insert values.
   - ○ Example: const age = 30; const message =I am ${age} years old.;
2. **Multi-Line Strings:**
   - ○ Create strings spanning multiple lines without concatenation.
   - ○ Preserve line breaks and indentation.
   - ○ Example: const poem =Roses are red, Violets are blue, JavaScript is cool, And so are you.;
3. **Tagged Templates:**
   - ○ Customize string processing using functions.
   - ○ Pass the template literal to a function using a tag (e.g., function myTag(strings, ...values) { ... }).
   - ○ Example: const highlighted = myTagThis is a ${adjective} sentence.;

Advantages of Template Literals:

- **Readability:** Improve code clarity, especially for multi-line strings and complex formatting.
- **Convenience:** Eliminate manual concatenation and escaping for quotes within strings.
- **Flexibility:** Tagged templates allow custom string processing for diverse needs.

When to Use Backticks:

- Constructing strings with dynamic content or expressions.
- Creating multi-line strings for better readability and formatting.
- Implementing custom string processing using tagged templates.

Browser Compatibility:

- Template literals are supported in all modern browsers, including Edge, Firefox, Chrome, Safari, and Opera (since IE11, they require a polyfill).