```
// Type definitions for [~THE LIBRARY NAME~] [~OPTIONAL VERSION NUMBER~]
// Project: [~THE PROJECT NAME~]
// Definitions by: [~YOUR NAME~] <[~A URL FOR YOU~]>

/*~ This template shows how to write a global plugin. */

/*~ Write a declaration for the original type and add new members.
 *~ For example, this adds a 'toBinaryString' method with to overloads to
 *~ the built-in number type.
 */
interface Number {
    toBinaryString(opts?: MyLibrary.BinaryFormatOptions): string;
    toBinaryString(callback: MyLibrary.BinaryFormatCallback, opts?: MyLibrary.B
}

/*~ If you need to declare several types, place them inside a namespace
 *~ to avoid adding too many things to the global namespace.
 */
declare namespace MyLibrary {
    type BinaryFormatCallback = (n: number) => string;
    interface BinaryFormatOptions {
        prefix?: string;
        padding: number;
    }
}
```