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

/*~ This is the module template file. You should rename it to index.d.ts
 *~ and place it in a folder with the same name as the module.
 *~ For example, if you were writing a file for "super-greeter", this
 *~ file should be 'super-greeter/index.d.ts'
 */

/*~ If this module is a UMD module that exposes a global variable 'myLib' when
 *~ loaded outside a module loader environment, declare that global here.
 *~ Otherwise, delete this declaration.
 */
export as namespace myLib;

/*~ If this module has methods, declare them as functions like so.
 */
export function myMethod(a: string): string;
export function myOtherMethod(a: number): number;

/*~ You can declare types that are available via importing the module */
export interface someType {
    name: string;
    length: number;
    extras?: string[];
}

/*~ You can declare properties of the module using const, let, or var */
export const myField: number;

/*~ If there are types, properties, or methods inside dotted names
 *~ of the module, declare them inside a 'namespace'.
 */
export namespace subProp {
    /*~ For example, given this definition, someone could write:
     *~   import { subProp } from 'yourModule';
     *~   subProp.foo();
     *~ or
     *~   import * as yourMod from 'yourModule';
     *~   yourMod.subProp.foo();
     */
    export function foo(): void;
}
```