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

/*~ This is the global-modifying module template file. You should rename it to
 *~ 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'
 */

/*~ Note: If your global-modifying module is callable or constructable, you'll
 *~ need to combine the patterns here with those in the module-class or module-
 *~ template files
 */
declare global {
    /*~ Here, declare things that go in the global namespace, or augment
     *~ existing declarations in the global namespace
     */
    interface String {
        fancyFormat(opts: StringFormatOptions): string;
    }
}

/*~ If your module exports types or values, write them as usual */
export interface StringFormatOptions {
    fancinessLevel: number;
}

/*~ For example, declaring a method on the module (in addition to its global si
export function doSomething(): void;

/*~ If your module exports nothing, you'll need this line. Otherwise, delete it
export { };
```