/\*\*

\* The `string\_decoder` module provides an API for decoding `Buffer` objects into

\* strings in a manner that preserves encoded multi-byte UTF-8 and UTF-16

\* characters. It can be accessed using:

\*

\* ```js

\* const { StringDecoder } = require('string\_decoder');

\* ```

\*

\* The following example shows the basic use of the `StringDecoder` class.

\*

\* ```js

\* const { StringDecoder } = require('string\_decoder');

\* const decoder = new StringDecoder('utf8');

\*

\* const cent = Buffer.from([0xC2, 0xA2]);

\* console.log(decoder.write(cent));

\*

\* const euro = Buffer.from([0xE2, 0x82, 0xAC]);

\* console.log(decoder.write(euro));

\* ```

\*

\* When a `Buffer` instance is written to the `StringDecoder` instance, an

\* internal buffer is used to ensure that the decoded string does not contain

\* any incomplete multibyte characters. These are held in the buffer until the

\* next call to `stringDecoder.write()` or until `stringDecoder.end()` is called.

\*

\* In the following example, the three UTF-8 encoded bytes of the European Euro

\* symbol (`€`) are written over three separate operations:

\*

\* ```js

\* const { StringDecoder } = require('string\_decoder');

\* const decoder = new StringDecoder('utf8');

\*

\* decoder.write(Buffer.from([0xE2]));

\* decoder.write(Buffer.from([0x82]));

\* console.log(decoder.end(Buffer.from([0xAC])));

\* ```

\* @see [source](https://github.com/nodejs/node/blob/v17.0.0/lib/string\_decoder.js)

\*/

declare module 'string\_decoder' {

class StringDecoder {

constructor(encoding?: BufferEncoding);

/\*\*

\* Returns a decoded string, ensuring that any incomplete multibyte characters at

\* the end of the `Buffer`, or `TypedArray`, or `DataView` are omitted from the

\* returned string and stored in an internal buffer for the next call to`stringDecoder.write()` or `stringDecoder.end()`.

\* @since v0.1.99

\* @param buffer A `Buffer`, or `TypedArray`, or `DataView` containing the bytes to decode.

\*/

write(buffer: Buffer): string;

/\*\*

\* Returns any remaining input stored in the internal buffer as a string. Bytes

\* representing incomplete UTF-8 and UTF-16 characters will be replaced with

\* substitution characters appropriate for the character encoding.

\*

\* If the `buffer` argument is provided, one final call to `stringDecoder.write()`is performed before returning the remaining input.

\* After `end()` is called, the `stringDecoder` object can be reused for new input.

\* @since v0.9.3

\* @param buffer A `Buffer`, or `TypedArray`, or `DataView` containing the bytes to decode.

\*/

end(buffer?: Buffer): string;

}

}

declare module 'node:string\_decoder' {

export \* from 'string\_decoder';

}