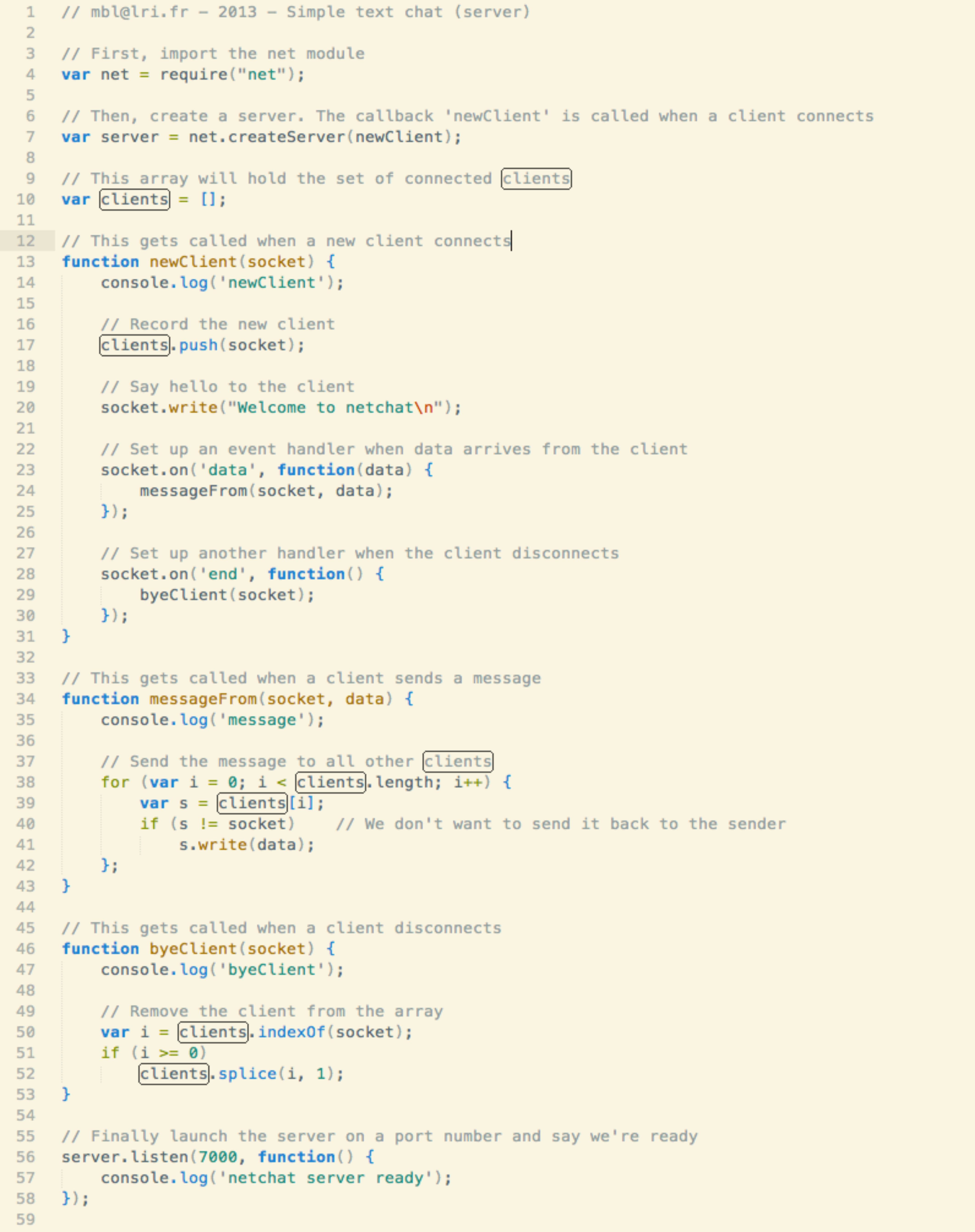
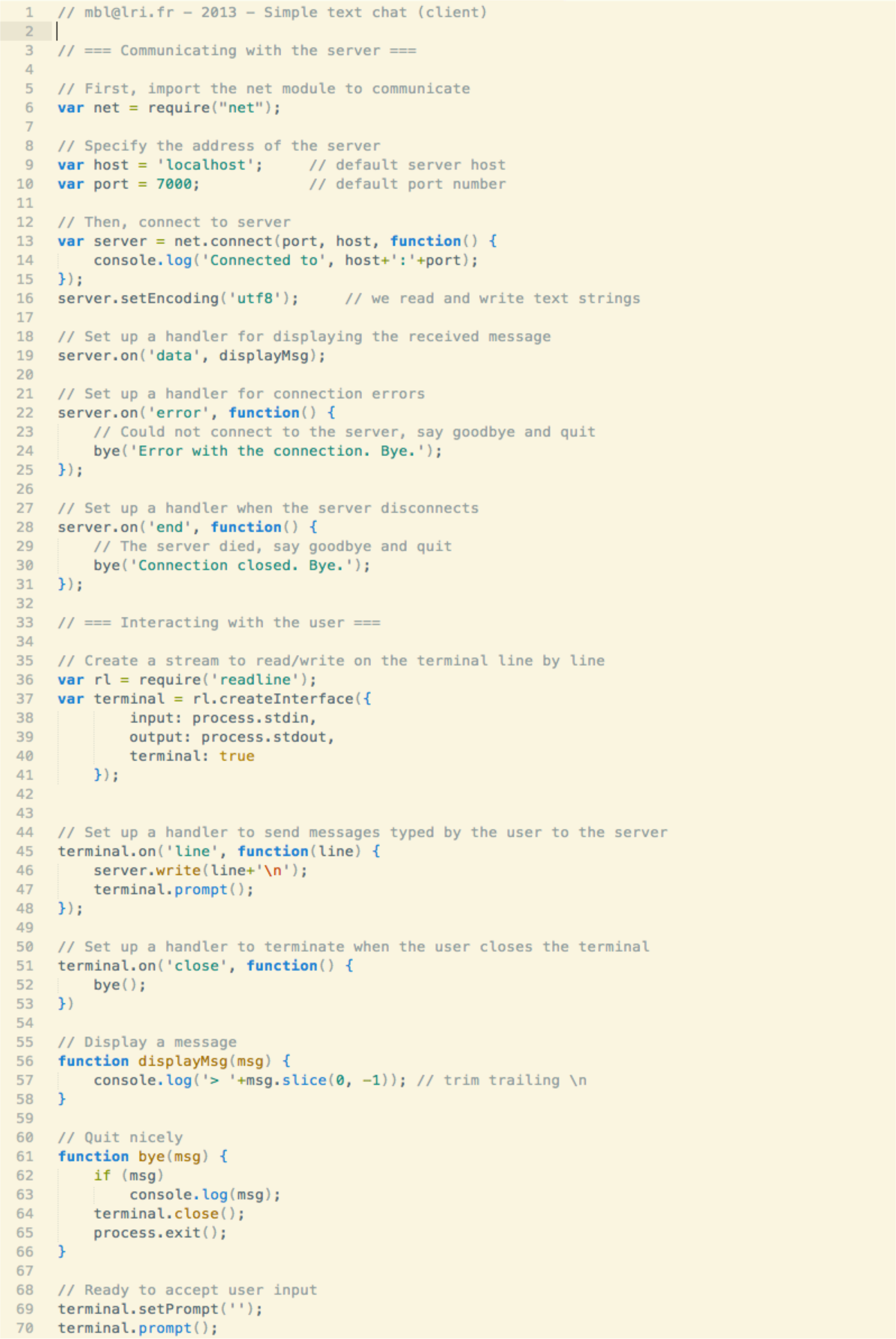
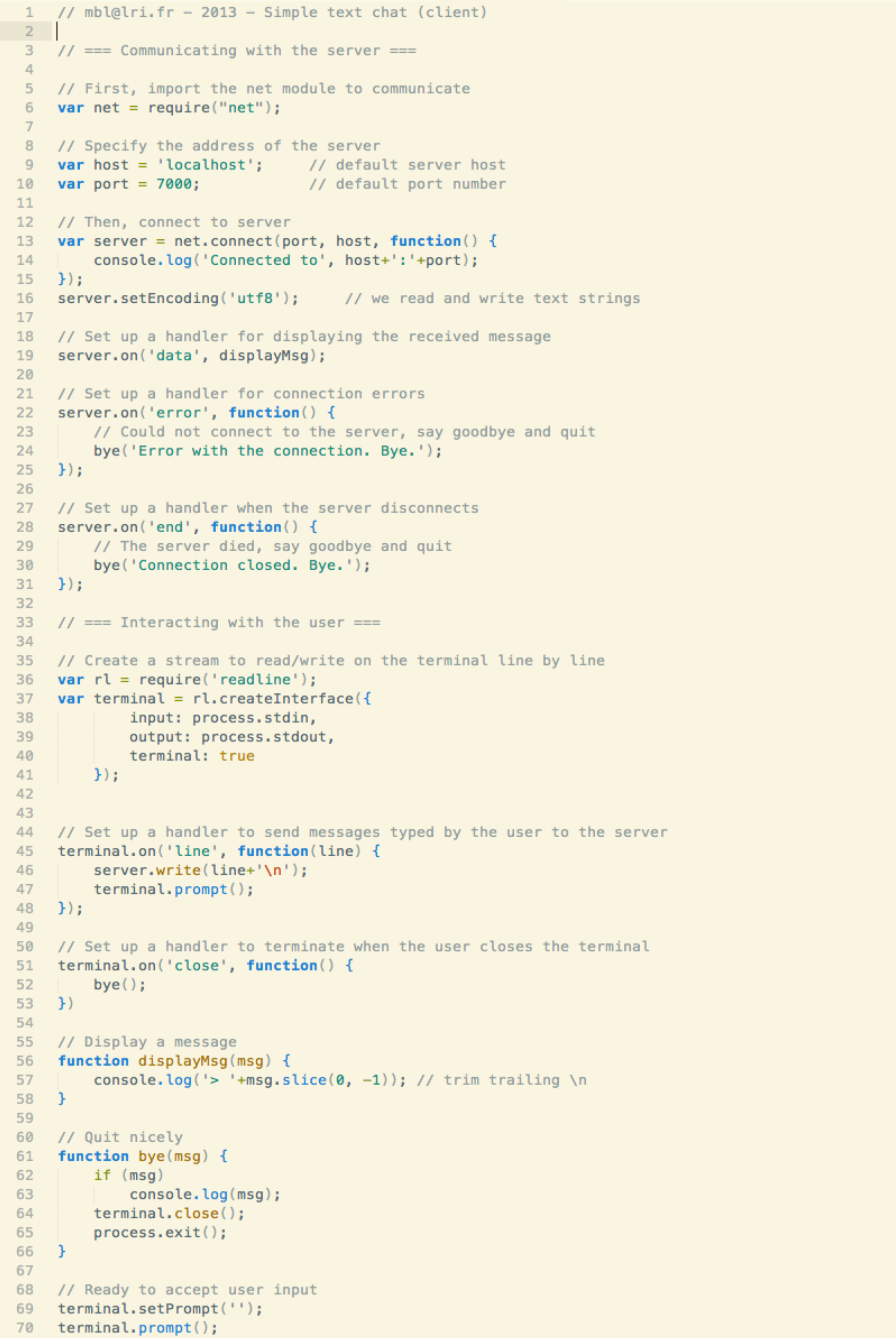


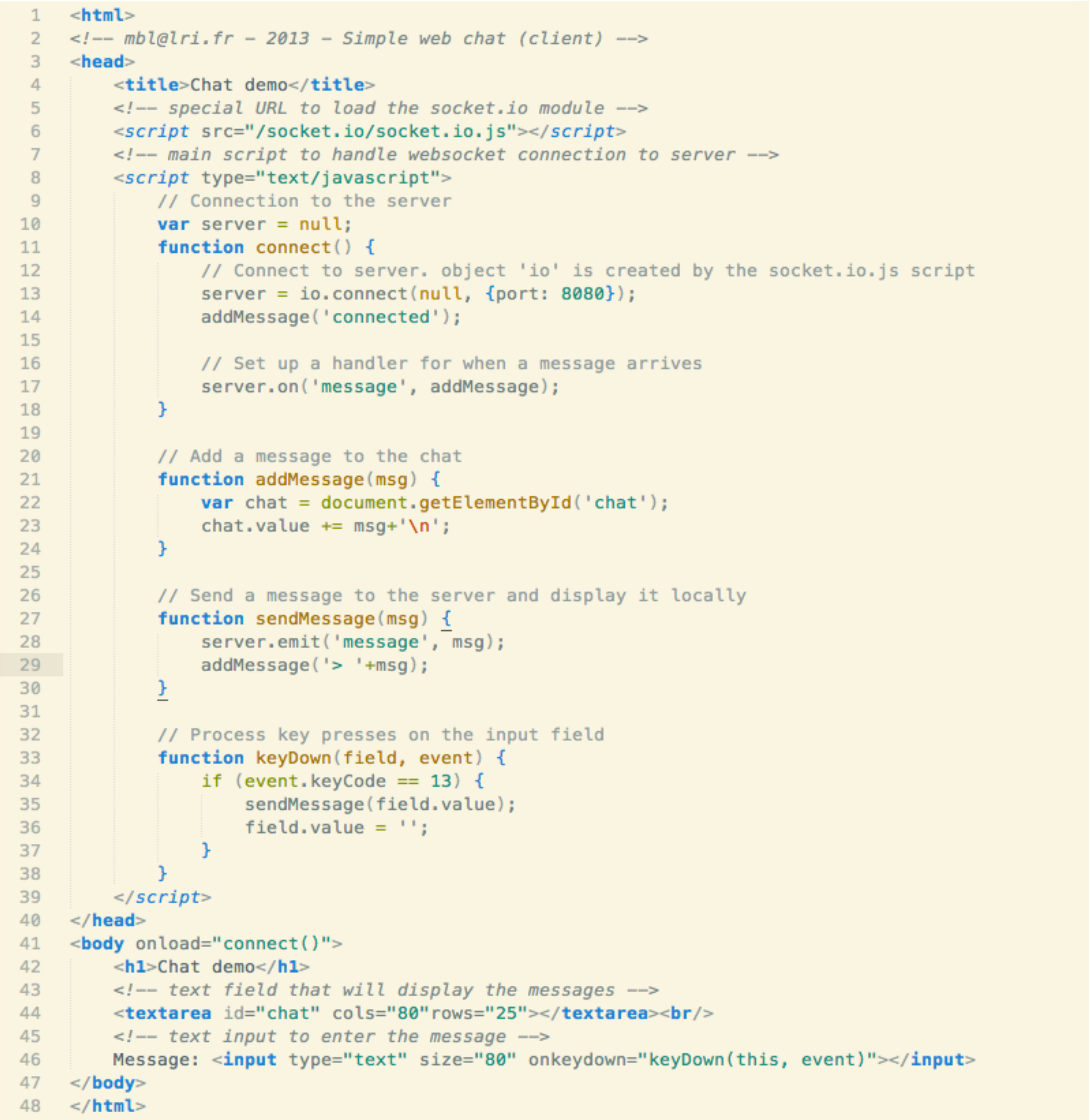
```javascript
// mbl@lri.fr - 2013 - Simple text chat (server)

// First, import the net module
var net = require("net");

// Then, create a server. The callback 'newClient' is called when a client connects
var server = net.createServer(newClient);

// This array will hold the set of connected clients
var clients = [];

// This gets called when a new client connects
function newClient(socket) {
    console.log('newClient');

    // Record the new client
    clients.push(socket);

    // Say hello to the client
    socket.write("Welcome to netchat\n");

    // Set up an event handler when data arrives from the client
    socket.on('data', function(data) {
        messageFrom(socket, data);
    });

    // Set up another handler when the client disconnects
    socket.on('end', function() {
        byeClient(socket);
    });
}

// This gets called when a client sends a message
function messageFrom(socket, data) {
    console.log('message');

    // Send the message to all other clients
    for (var i = 0; i < clients.length; i++) {
        var s = clients[i];
        if (s != socket)    // We don't want to send it back to the sender
            s.write(data);
    };
}

// This gets called when a client disconnects
function byeClient(socket) {
    console.log('byeClient');

    // Remove the client from the array
    var i = clients.indexOf(socket);
    if (i >= 0)
        clients.splice(i, 1);
}

// Finally launch the server on a port number and say we're ready
server.listen(7000, function() {
    console.log('netchat server ready');
});
```



```
// mbl@lri.fr - 2013 - Simple text chat (client)

// === Communicating with the server ===

// First, import the net module to communicate
var net = require("net");

// Specify the address of the server
var host = 'localhost';     // default server host
var port = 7000;            // default port number

// Then, connect to server
var server = net.connect(port, host, function() {
    console.log('Connected to', host+':'+port);
});
server.setEncoding('utf8');     // we read and write text strings

// Set up a handler for displaying the received message
server.on('data', displayMsg);

// Set up a handler for connection errors
server.on('error', function() {
    // Could not connect to the server, say goodbye and quit
    bye('Error with the connection. Bye.');
});

// Set up a handler when the server disconnects
server.on('end', function() {
    // The server died, say goodbye and quit
    bye('Connection closed. Bye.');
});

// === Interacting with the user ===

// Create a stream to read/write on the terminal line by line
var rl = require('readline');
var terminal = rl.createInterface({
        input: process.stdin,
        output: process.stdout,
        terminal: true
    });


// Set up a handler to send messages typed by the user to the server
terminal.on('line', function(line) {
    server.write(line+'\n');
    terminal.prompt();
});

// Set up a handler to terminate when the user closes the terminal
terminal.on('close', function() {
    bye();
})

// Display a message
function displayMsg(msg) {
    console.log('> '+msg.slice(0, -1)); // trim trailing \n
}

// Quit nicely
function bye(msg) {
    if (msg)
        console.log(msg);
    terminal.close();
    process.exit();
}

// Ready to accept user input
terminal.setPrompt('');
terminal.prompt();
```



```javascript
// mbl@lri.fr - 2013 - Simple text chat (client)

// === Communicating with the server ===

// First, import the net module to communicate
var net = require("net");

// Specify the address of the server
var host = 'localhost';     // default server host
var port = 7000;            // default port number

// Then, connect to server
var server = net.connect(port, host, function() {
    console.log('Connected to', host+':'+port);
});
server.setEncoding('utf8');     // we read and write text strings

// Set up a handler for displaying the received message
server.on('data', displayMsg);

// Set up a handler for connection errors
server.on('error', function() {
    // Could not connect to the server, say goodbye and quit
    bye('Error with the connection. Bye.');
});

// Set up a handler when the server disconnects
server.on('end', function() {
    // The server died, say goodbye and quit
    bye('Connection closed. Bye.');
});

// === Interacting with the user ===

// Create a stream to read/write on the terminal line by line
var rl = require('readline');
var terminal = rl.createInterface({
        input: process.stdin,
        output: process.stdout,
        terminal: true
    });


// Set up a handler to send messages typed by the user to the server
terminal.on('line', function(line) {
    server.write(line+'\n');
    terminal.prompt();
});

// Set up a handler to terminate when the user closes the terminal
terminal.on('close', function() {
    bye();
})

// Display a message
function displayMsg(msg) {
    console.log('> '+msg.slice(0, -1)); // trim trailing \n
}

// Quit nicely
function bye(msg) {
    if (msg)
        console.log(msg);
    terminal.close();
    process.exit();
}

// Ready to accept user input
terminal.setPrompt('');
terminal.prompt();
```



```
<html>
<!-- mbl@lri.fr - 2013 - Simple web chat (client) -->
<head>
    <title>Chat demo</title>
    <!-- special URL to load the socket.io module -->
    <script src="/socket.io/socket.io.js"></script>
    <!-- main script to handle websocket connection to server -->
    <script type="text/javascript">
        // Connection to the server
        var server = null;
        function connect() {
            // Connect to server. object 'io' is created by the socket.io.js script
            server = io.connect(null, {port: 8080});
            addMessage('connected');

            // Set up a handler for when a message arrives
            server.on('message', addMessage);
        }

        // Add a message to the chat
        function addMessage(msg) {
            var chat = document.getElementById('chat');
            chat.value += msg+'\n';
        }

        // Send a message to the server and display it locally
        function sendMessage(msg) {
            server.emit('message', msg);
            addMessage('> '+msg);
        }

        // Process key presses on the input field
        function keyDown(field, event) {
            if (event.keyCode == 13) {
                sendMessage(field.value);
                field.value = '';
            }
        }
    </script>
</head>
<body onload="connect()">
    <h1>Chat demo</h1>
    <!-- text field that will display the messages -->
    <textarea id="chat" cols="80"rows="25"></textarea><br/>
    <!-- text input to enter the message -->
    Message: <input type="text" size="80" onkeydown="keyDown(this, event)"></input>
</body>
</html>
```