Traditional Approach before Node.js

Client — Request → Web server — → Thread 1, Thread 2, Thread 3

After Node.js

Single thread

Client, Client, Client → Web Server → Threads Pool
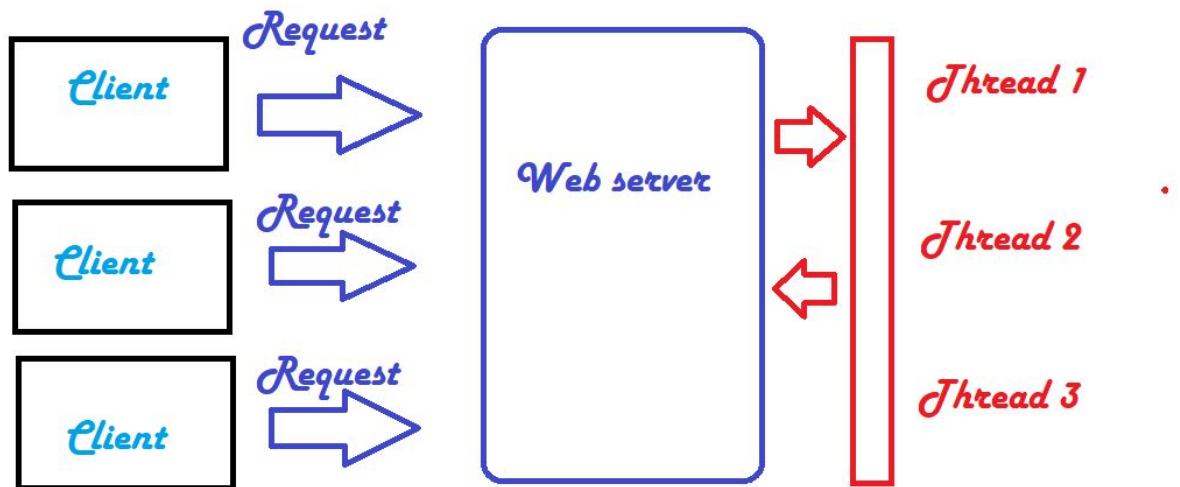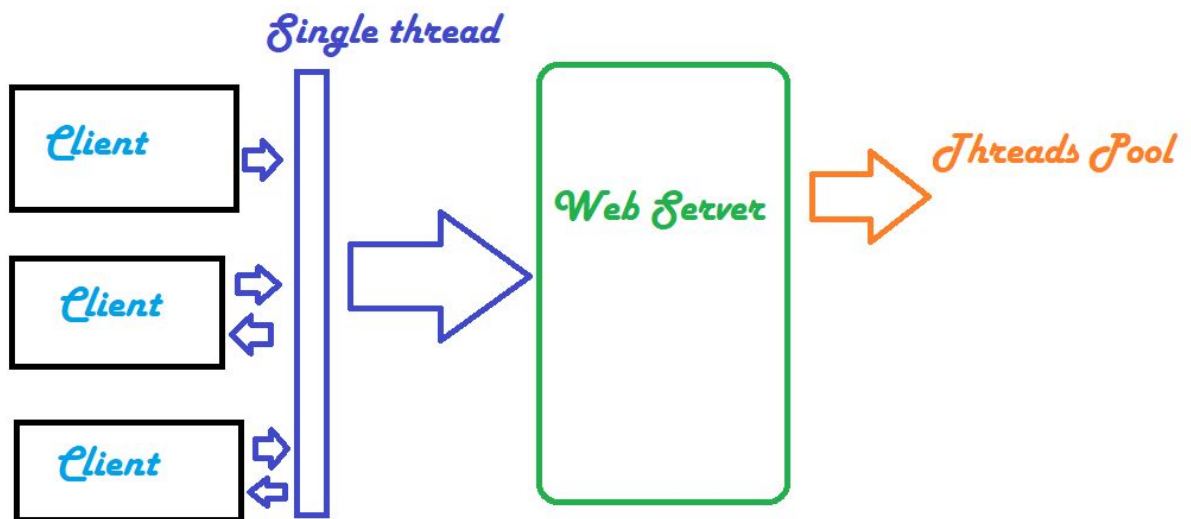
Node.js Modules

1. Core Modules
2. Local Modules
3. Third Party Modules

1. **Core Modules**
   a. http : includes all the classes and methods and events of http
   b. url
   c. querystring
   d. path
   e. fs
   f. util

2. **Local Modules**

   Create the Local Module with name as "test.js" and place this code

   ```
   test - Notepad
   File  Edit  Format  View  Help

   exports.myDateTime = function(){
           return Date();
   };
   ```

   Create the new file with the name as "LocalModule.js" and place the below code
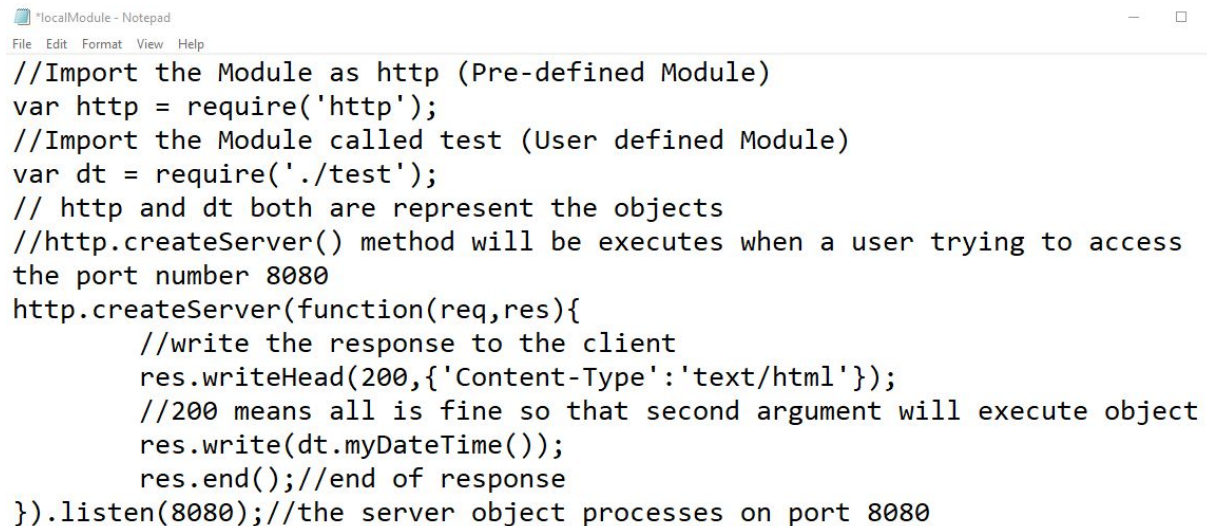
   ```
   localModule - Notepad
   File  Edit  Format  View  Help

   var http = require('http');
   var dt = require('./test');
   http.createServer(function(req,res){
           res.writeHead(200,{'Content-Type':'text/html'});
           res.write(dt.myDateTime());
           res.end();
   }).listen(8080);
   ```

   Go to command prompt and then type node localModule.js and then go to browser ping for localhost:8080

```
C:\Users\Anil Kumar-DSPS>node localModule.js
```

Explanation of the above programs:

```
*localModule - Notepad                                          —    □    ×
File  Edit  Format  View  Help
//Import the Module as http (Pre-defined Module)
var http = require('http');
//Import the Module called test (User defined Module)
var dt = require('./test');
// http and dt both are represent the objects
//http.createServer() method will be executes when a user trying to access
the port number 8080
http.createServer(function(req,res){
        //write the response to the client
        res.writeHead(200,{'Content-Type':'text/html'});
        //200 means all is fine so that second argument will execute object
        res.write(dt.myDateTime());
        res.end();//end of response
}).listen(8080);//the server object processes on port 8080
```

Node.js as a File Server

1. Read file
2. Create file
3. Update file
4. Delete file
5. Rename file

First create the HTML Doc

File Edit Format View Help

```html
<html>
        <body>
                <h1>Dhyanahita</h1>
        </body>
</html>
```

Create the node.js file

File Edit Format View Help

```javascript
var http = require('http');//http Module
var fs = require('fs');//fs - File Server Module
http.createServer(function(req,res){
        fs.readFile('Demo1.html',function(err,data){
                res.writeHead(200,{'Content-Type':'text/html'});
                res.write(data);
                return res.end();
        });
}).listen(8080);
```

```
C:\Users\Anil Kumar-DSPS>node fsDemo.js
```

Task : Create Html with some basic CSS with one or two classes, then load that file to node.js

fs.appendFile() : It's for updating the existing file, it means let's take an example of a text file and try to write some new data then it will update the existing file.

fs.open() :

fs.writeFile() :