1.append.js

const fs = require('fs');

// Reading a file asynchronously

fs.appendFile('newfile1.txt', 'Hi Hello', (err, data) => {

  if (err) {

    console.error(err);

    return;

  }

  console.log("success");

});

2.assert.js

const assert = require('assert');

// Example function to test

function add(a, b) {

  return a + b;

}

// Test case using assert module

try {

  assert.strictEqual(add(1, 2), 3); // Passes, 1 + 2 equals 3

  assert.deepEqual(add(-1, 1), 0); // Passes, -1 + 1 equals 0

  assert.deepStrictEqual(add(0, 0), 0); // Passes, 0 + 0 equals 0

  assert.notStrictEqual(add(2, 3), 5); // Passes, 2 + 3 equals 5

  assert.notDeepEqual(add(5, -3), 2); // Passes, 5 + (-3) equals 2

  assert.ok(1);

  assert.ok(0);

  console.log('All assertions passed successfully!');

} catch (error) {

  console.error('Assertion failed:', error.message);

}

3.event\_once.js

// Import the events module

const EventEmitter = require('events');

// Create an instance of EventEmitter

const myEmitter = new EventEmitter();

// Register an event listener for the 'once' event

myEmitter.once('once', () => {

  console.log('This event will only be handled once.');

});

// Emit the 'once' event multiple times

myEmitter.emit('once');

myEmitter.emit('once'); // This won't trigger the listener again

4.events1.js

// Import the events module

const EventEmitter = require('events');

// Create an instance of EventEmitter

const myEmitter = new EventEmitter();

// Register an event listener for the 'greet' event

myEmitter.on('greet', (name) => {

  console.log(`Hello, ${name}!`);

});

// Emit the 'greet' event with an argument

myEmitter.emit('greet', 'John');

5.http\_get.js

const http = require('http');

http.get('http://example.com', (res) => {

  let data = '';

  // A chunk of data has been received.

  res.on('data', (chunk) => {

    data += chunk;

  });

  // The whole response has been received.

  res.on('end', () => {

    console.log(data);

  });

}).on('error', (err) => {

  console.log('Error: ' + err.message);

});

6.http\_request.js

const http = require('http');

const options = {

  hostname: 'example.com',

  port: 80,

  path: '/',

  method: 'POST',

};

const req = http.request(options, (res) => {

  let data = '';

  // A chunk of data has been received.

  res.on('data', (chunk) => {

    data += chunk;

  });

  // The whole response has been received.

  res.on('end', () => {

    console.log(data);

  });

});

req.on('error', (err) => {

  console.log('Error: ' + err.message);

});

req.end();

7.http1.js

var http = require('http');

http.createServer(function (req, res) {

  res.writeHead(200, {'Content-Type': 'text/plain'});

  res.write('Hello World!');

  res.end();

}).listen(8089,  'localhost', () => {

    console.log('Server running at http://localhost:8080/');

});

8.http2.js

var http = require('http');

http.createServer(function (req, res) {

  res.writeHead(200, {'Content-Type': 'text/html'});

  res.write(req.url);

  res.end();

}).listen(8998,  'localhost', () => {

    console.log('Server running at http://localhost:8998/summer');

});

9.httpex3.js

const http = require('http');

const fs = require('fs');

http.createServer((req, res) => {

  const readStream = fs.createReadStream('largeFile.txt');

  res.writeHead(200, { 'Content-Type': 'text/plain' });

  readStream.on('data', (chunk) => {

    res.write(chunk);  // Sending data in chunks

  });

  readStream.on('end', () => {

    res.end();  // Signal that the response is complete

  });

}).listen(9089,  'localhost', () => {

    console.log('Server running at http://localhost:9089/');

});

10.largefile.txt

This is a sample text file.

It contains multiple lines of text.

11.myrenamedfile.txt

HELLO WORLD WELCOME TO NODE JS CLASS

12.newfile1.txt

Hi HELLO WORLD WELCOME TO NODE JS CLASSHi HelloHi Hello

13.newfile3.txt

Hello, Node.js!

14.openfile.js

/\*var fs = require('fs');

fs.open('newfile12.txt', 'w', function (err, file) {

  if (err) throw err;

  console.log('Saved!');

});\*/

const fs = require('fs');

const filePath = 'newfile1.txt';

fs.open(filePath, 'w', (err, fd) => {

    if (err) {

        console.error('Error opening file:', err);

        return;

    }

    console.log(`File ${filePath} opened successfully for reading`);

});

15.parse.js

const queryString = require('querystring');

const parsedQuery = queryString.parse('This%20is%20my%20query%20string=s');

console.log(parsedQuery);

16.path.js

const path = require('path');

const filePath = '/path/to/some/file.txt';

// Extracting the directory name

const dirname = path.dirname(filePath);

console.log('Directory:', dirname);

// Extracting the file name

const basename = path.basename(filePath);

console.log('File Name:', basename);

// Joining paths

const fullPath = path.join(\_\_dirname, 'files', 'example.txt');

console.log('Full Path:', fullPath);

17.qsdecode.js

const encodedString = 'This%20is%20my%20query%20string';

const decodedString = decodeURIComponent(encodedString);//instead of unescape()

console.log(decodedString); // Output: 'This is my query string'

18.quencode.js

const originalString = 'This is my query string';

const encodedString = encodeURIComponent(originalString);//instead of escape encodeURIComponent()

console.log(encodedString); // Output: 'This%20is%20my%20query%20string'

19.readfile.js

const fs = require('fs');

// Reading a file asynchronously

fs.readFile('newfile1.txt', 'utf8', (err, data) => {

  if (err) {

    console.error(err);

    return;

  }

  console.log(data);

});

20.rename.js

var fs = require('fs');

fs.rename('newfile1.txt', 'myrenamedfile.txt', function (err) {

  if (err) throw err;

  console.log('File Renamed!');

});

21.stringify.js

const queryString = require('querystring');

const obj = { q: 'nodejs', page: 2 };

const stringifiedQuery = queryString.stringify(obj);

console.log(stringifiedQuery); // Output: q=nodejs&page=2

22.unlik-delete.js

var fs = require('fs');

fs.unlink('newfile2.txt', function (err) {

  if (err) throw err;

  console.log('File deleted!');

});

23.url\_format.js

const url = require('url');

const parsedUrl = {

  protocol: 'https:',

  hostname: 'example.com',

  pathname: '/path/to/resource',

  query: { key1: 'value1', key2: 'value2' }

};

const formattedUrl = url.format(parsedUrl);

console.log(formattedUrl); // Output: 'https://example.com/path/to/resource?key1=value1&key2=value2'

24.url\_parse.js

const url = require('url');

const urlString = 'https://www.w3schools.com/nodejs/ref\_url.asp';

const parsedUrl = url.parse(urlString, true);

console.log(parsedUrl.hostname);

console.log(parsedUrl.pathname);

console.log("query: ",  parsedUrl.query);

25.url\_resolve.js

const url = require('url');

const baseUrl = 'https://example.com/';

const relativeUrl = '/about';

const aurl = '/abc';

const resolvedUrl = url.resolve(baseUrl,  aurl, relativeUrl);

console.log(resolvedUrl);

26.writefile.js

const fs=require('fs');

fs.writeFile('newfile3.txt', 'Hello, Node.js!', (err) => {

    if (err) {

      console.error(err);

      return;

    }

    console.log('File written successfully.');

  });