



*Front-end Advanced*

# Training Assignment

Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.1
Effective Date	7/1/2019

Hanoi, mm/yyyy

**RECORD OF CHANGES**

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	30/May/2019	Create a new assignment	Create new	DieuNT1	VinhNV
2	07/Jun/2019	Update Fsoft Template	Update	DieuNT1	VinhNV

## Contents

Day 2. Unit 2-5: Async Programming .....	4
Objectives: .....	4
Assumptions: .....	4
Problem 01: .....	5
Problem 02: .....	5
Problem 03: .....	5
Problem 04: .....	5



CODE:	Async-JS.M.A103 (Async 03)
TYPE:	Medium
LOC:	300
DURATION:	180

## Day 2. Unit 2-5: Async Programming

### Objectives:

- Understand how to abstract asynchronous task in JavaScript (Async Programming)
- Understand the core concept of Async in JavaScript: the Event Loop
- Understand the concept of Parallel Programming and how JavaScript implement that concept
- Able to working with Async API with ease (Callback, Promise, Generators, async/await)

### Assumptions:

You are given a file called **numfiles.zip**

Unzip it and you will have a folder contains many small files like below:

```
numfiles — lyly@lyly — ..peak/numfiles — -zsh — 80x24
157 236 32232 381 453 5265 60047 66710 7421 818 8757
1571 23633 3224 3810 4532 52722 60126 66768 74243 8180 876
1574 237 323 382 4535 52735 6037 668 7425 81801 87613
15781 2384 3237 38234 454 5274 6048 66802 74263 8181 87708
1583 2387 32373 383 455 52834 605 6681 743 81837 8774
15888 240 3238 3832 4551 5285 6052 670 7438 820 87804
160 24024 324 38363 4558 5286 606 6701 744 82032 87854
16047 24042 32404 3838 45585 530 60678 67016 74525 821 8805
16057 24068 32518 384 45681 5300 607 6707 74544 82105 8808
16060 24114 3252 3846 457 531 608 671 7457 82113 88188
161 2416 3257 385 45743 5311 6088 6711 74576 8214 882
1610 242 32575 38505 4576 53116 6101 672 747 82158 8828
1611 2421 3258 38663 4588 532 61024 6720 74725 8217 883
1614 2425 326 387 460 533 6104 67246 7481 822 88377
16184 2428 3260 388 46068 534 61075 6737 74834 823 88414
162 243 32714 3887 4607 5346 61100 6738 74837 824 885
16214 244 3272 400 46143 535 6116 674 7507 8241 8850
1624 2442 328 40153 46203 53544 6128 6745 751 8242 88515
16254 2443 330 402 4628 5362 613 6762 7512 825 88642
163 24448 3304 40288 463 53628 6131 67630 7513 82582 887
16340 2448 33076 403 4630 5363 61388 6767 7515 826 88870
164 24516 331 4035 4632 5366 6144 67754 752 8263 readme
1640 24622 33167 40364 464 5370 6147 678 75216 82772
→ numfiles git:(dev) x
```

You need to do the following:

- Start from file 1024.
- Collect the binary digit enclosed in [].
- Proceed to get the next file and the next binary digit.

- Continue till the last file.

To read a file use the following API from NodeJS, which will read the file **1024** asynchronously

```
1. const fs = require('fs');
2.
3. fs.readFile('1024', 'utf-8', function(err, data) {
4.   if (err) {
5.     throw err;
6.   }
7.
8.   console.log(data);
9. });
```

### **Problem 01:**

You have to do the requirements using **Callback** first then **Promise** and lastly use **async/await**

You need to setup your project folder in order to do so.

Expected output:

```
010101000110100001100101001000000111001101101111011011000111010101110100011010010110
111101101110001000000110100101110011001000000111010001101000011001010010000001101110
01100001011011010110010100100000011011101100110001000000111010001101000011001010010
000001101100011011110110001101100001011101000110100101101111011011100010000001100001
011101000010000000110010001101010010111000110100001100000100111000100000001100110011
001000101110001100110011010101000101001000000101010001101000011001010010000001110011
011011110110110001110101011101000110100101101111011011100010000001101001011100110010
000001110100011010000110010100100000011011100110000101101101011001010010000001101111
011001100010000001110100011010000110010100100000011011000110111101100011011000010111
010001101001011011110110111000100000011000010111010000100000001100100011010100101110
001101000011000001001110001000000011001100110010001011100011001100110101010001010010
000001010100011010000110010100100000011100110110111101101100011101010111010001101001
011011110110111000100000011010010111001100100000011101000110100001100101001000000110
111001100001011011010110010100100000011011110110011000100000011101000110100001100101
001000000110110001101111011000110110000101110100011010010110111101101110001000000110
000101110100001000000011001000110101001011100011010000110000010011100010000000110011
0011001000101110001100110011010101000101
```

### **Problem 02:**

You must do the same tasks like Exercise 1, but you have to use Promise API instead of Callback Style.

### **Problem 03:**

You must do the same tasks like Exercise 1, but you have to use Generators API.

### **Problem 04:**

You must do the same tasks like Exercise 1, but you have to use async/await API.