Node JS-2

- **Chalk comes with an easy to use composable API where you just chain and nest the styles you want.**

```
import chalk from 'chalk';

const log = console.log;

// Combine styled and normal strings

log(chalk.blue('Hello') + ' World' + chalk.red('!'));

// Compose multiple styles using the chainable API

log(chalk.blue.bgRed.bold('Hello world!'));

// Pass in multiple arguments

log(chalk.blue('Hello', 'World!', 'Foo', 'bar', 'biz', 'baz'));

// Nest styles

log(chalk.red('Hello', chalk.underline.bgBlue('world') + '!'));

// Nest styles of the same type even (color, underline, background)

log(chalk.green(‘I am a green line ' +chalk.blue.underline.bold('with a blue substring') +

        ' that becomes green again!'));

// Use RGB colors in terminal emulators that support it.

log(chalk.rgb(123, 45, 67).underline('Underlined reddish color'));

log(chalk.hex('#DEADED').bold('Bold gray!'));
```

- **Easily define your own themes**

```
import chalk from 'chalk';

const error = chalk.bold.red;

const warning = chalk.hex('#FFA500'); // Orange color

console.log(error('Error!'));

console.log(warning('Warning!'));
```

- **Take advantage of console.log string substitution**

```
import chalk from 'chalk';

const name = 'Mam';

console.log(chalk.green('Hello %s'), name);

//=> 'Hello Mam'
```

JSON Processing

When we create a simple object in a JS with key & value pair, then it is called as JSON object. Even if we provide each key pair as string, then also it treats them without string keys like an object. It is not possible to write JSON object directly to the file. We can write after calling stringify() method. Only because string can be easily written to the file; we can write individual properties of JSON object directly.

Task: Write node js script and json to perform below tasks.

1. Write below object in txt file named s2.txt {d:{a:10,b:20,c:[30,10]}}

2. Read data from the same file and perform the below tasks.

(i) addition of a and b. (ii) subtraction of 2nd element of c and b. (Must be positive value) (iii) multiplication of elements of c.

3. Add the Output of addition, subtraction and multiplication below the object in

s2.txt file.

```
var fs=require("fs");

const data ={d:{a:10,b:20,c:[30,10]}}

fs.writeFileSync("s2.txt",JSON.stringify(data)+"\n");

data1=fs.readFileSync("S2.txt","utf-8");

console.log(data1)

data1=JSON.parse(data1);

add=data1.d.a+data1.d.b;

sub=Math.abs(data1.d.c[1]-data1.d.b);

mul=data1.d.c[1]*data1.d.c[0];

console.log(add,sub,mul)

fs.appendFileSync("s2.txt",add+"\n"+sub+"\n"+mul)
```