

# COMP6080

# Web Front-End Programming

Javascript - NodeJS  
Demonstration

# Let's demo together!

Let's write a simple NodeJS script that:

- 
- Reads in up to 20 dates in the format "YYYY/MM/DD" from argv
  - For each date, calculates how many days since that date have passed
  - Creates a list of keys:value pairs that store date:days-since-date
  - Outputs this as JSON to a file on the file system

- Reads in up to 20 dates in the format "YYYY/MM/DD" from argv

```
10  const allDates = process.argv.slice(2);
11  for (const date of allDates) {
12      console.log(date);
13  }
14  allDates.forEach(date => {
15      console.log(date);
16  });
17  allDates.map(date => {
18      console.log(date);
19  });
20
```

- For each date, calculates how many days since that date have passed

```
10  const allDates = process.argv.slice(2);
11  for (const date of allDates) {
12      const previousTime = Date.parse(date) / 1000;
13      const nowTime = new Date().getTime() / 1000;
14      const secondsDifference = nowTime - previousTime;
15      console.log(Math.floor(secondsDifference / 60 / 60 / 24))
16  }
17 /*allDates.forEach(date => {
18     console.log(date);
19 });
20 allDates.map(date => {
21     console.log(date);
22});*/
```

```
11 const ParseDate = require('date-fns/parse');
12 const differenceInBusinessDays = require('date-fns/differenceInBusinessDays')
13
14 const allDates = process.argv.slice(2);
15 for (const date of allDates) {
16     const interval = differenceInBusinessDays(
17         new Date(),
18         ParseDate(date, 'yyyy/MM/dd', new Date()),
19     );
20     console.log(interval);
21 }
```

```
11 const ParseDate = require('date-fns/parse');
12 const intervalToDuration = require('date-fns/intervalToDuration')
13
14
15 const allDates = process.argv.slice(2);
16 for (const date of allDates) {
17     const interval = intervalToDuration({
18         start: ParseDate(date, 'yyyy/MM/dd', new Date()),
19         end: new Date(),
20     });
21     console.log(interval.days);
22 }
```

- Creates a list of keys:value pairs that store date:days-since-date

```
11 const ParseDate = require('date-fns/parse');
12 const differenceInBusinessDays = require('date-fns/differenceInBusinessDays')
13
14 const allDates = process.argv.slice(2);
15 const pairs = {};
16 for (const date of allDates) {
17     const interval = differenceInBusinessDays(
18         new Date(),
19         ParseDate(date, 'yyyy/MM/dd', new Date()),
20     );
21     pairs[date] = interval;
22 }
23 console.log(pairs);
```

- Outputs this as JSON to a file on the file

```
11 const ParseDate = require('date-fns/parse');
12 const differenceInDays = require('date-fns/differenceInDays');
13 const fs = require('fs');
14
15 const allDates = process.argv.slice(2);
16 const pairs = {};
17 for (const date of allDates) {
18     const interval = differenceInDays(
19         new Date(),
20         ParseDate(date, 'yyyy/MM/dd', new Date()),
21     );
22     pairs[date] = interval;
23 }
24 const jsonString = JSON.stringify(pairs);
25 console.log(jsonString);
26
27 fs.writeFile('./output.json', jsonString, err => {
28     if (err) {
29         console.error(err)
30         return
31     }
32 })
```

# Let's demo together!

Let's write a simple NodeJS script that:

- Starts up a simple HTTP server with one GET route called "/scrape"
  - This route takes in a URL and a particular HTML tag to look for and count
  - This route returns a simple object with the number of times that tag appears on a webpage

```
1 const express = require('express');
2 const fs = require('fs');
3
4 const app = express();
5 const port = 3001;
6
7 app.get('/scrape', (req, res) => {
8     const data = fs.readFileSync('./output.json', 'utf8')
9     const unpacked = JSON.parse(data);
10    let returnString = 'Dates:<br />\n';
11    for (const dateKey of Object.keys(unpacked)) {
12        returnString += `${dateKey} => ${unpacked[dateKey]}<br />';
13    }
14    res.send(returnString);
15 })
16
17 app.get('/hayden', (req, res) => {
18     res.send('Hi Hayden!')
19 })
20
21 const handler2 = () => {
22     console.log(`Example app listening on port ${port}`)
23 };
24
25 app.listen(port, handler2);
```

# Feedback

