

ESMAD | TSIW | PWII  
Exercise Sheet nº1 – Introduction to Node.js

1. Implement a Node module to export color codes and corresponding HTML name. In your module, add the following array of objects with some colors:

```
const allColors = [  
  { name: 'brightred', code: '#E74C3C' },  
  { name: 'soothingpurple', code: '#9B59B6' },  
  { name: 'skyblue', code: '#5DADE2' },  
  { name: 'leafygreen', code: '#48C9B0' },  
  { name: 'sunkissedyellow', code: '#F4D03F' },  
  { name: 'groovygray', code: '#D7DBDD' }  
];
```

Complete the module, by exporting the following functionalities:

- 1) **getRandomColor**: randomly exports one color from the array
- 3) **getAllColors**: exports all colors

Import the created module into a Node application so that the following information is displayed into the command prompt: all available colors, one random color for a website, a favorite color.

```
Available colors:  
  brightred: #E74C3C  
  soothingpurple: #9B59B6  
  skyblue: #5DADE2  
  leafygreen: #48C9B0  
  sunkissedyellow: #F4D03F  
  groovygray: #D7DBDD  
  
You should use brightred on your website. It's HTML code is #E74C3C  
  
My favorite color is skyblue/#5DADE2
```

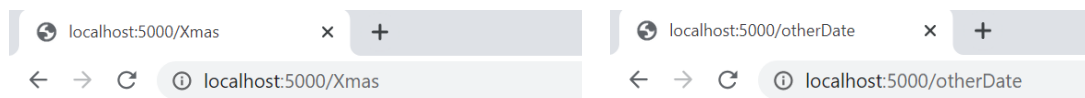
2. Implement a Node module to export the 4 basic mathematical operations between 2 numbers: addition, subtraction, multiplication, and division. The module must provide an error message if the divider of a division operation is zero.

Create a Node application that reads the command-line arguments with the format **operation number1 number2** and outputs the result of the operation:

```
> node ex2 + 1 2  
1 + 2 = 3  
> node ex2 - 1 2  
1 - 2 = -1  
> node ex2 . 1 2  
Sorry, could not recognize the operation  
> node ex2 +
```

Sorry, must provide three arguments

3. Implement a Node module that calculates the number of days until a special date (Christmas, Mother's Day, Children's Day,...). Choose at least 3 special dates, that have a fixed day in a year: Christmas – 25th december, Mother's day – 8th may, Valentine's day – 14th february. Build a web server accessed through port 8080 so that, according to the URL of the request, an HTML page is displayed, with the number of days calculated inside an `<h1>` header. If the URL corresponds to a calculation not predicted by the module, the server should return a `NOT_FOUND.html` page (add one to your project directory), with error code 404.

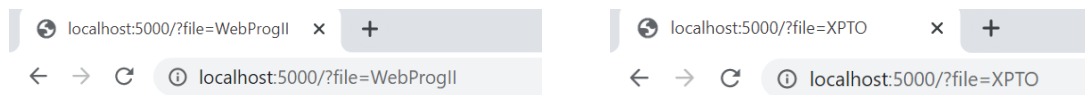


## 241 days till next Christmas!      Page NOT FOUND

HINT: consider the following code that returns the number of days until next Christmas

```
let today = new Date();
let cmas = new Date(today.getFullYear(), 11, 25); // months start at 0
let daysTill = Math.ceil((cmas.getTime() - today.getTime()) / (1000*60*60*24));
```

4. Using only native modules of Node.js, build a web server accessed through port 5000. It must serve static HTML pages by obtaining the name of the HTML file in the query string of the HTTP request. If the request corresponds to a non-existent file, the server must return a `NOT_FOUND.html` page, with error code 404. Add some sample HTML files into your application directory (including one `NOT_FOUND.html` page) and test it.



## Sample page for WebProgII      Page NOT FOUND