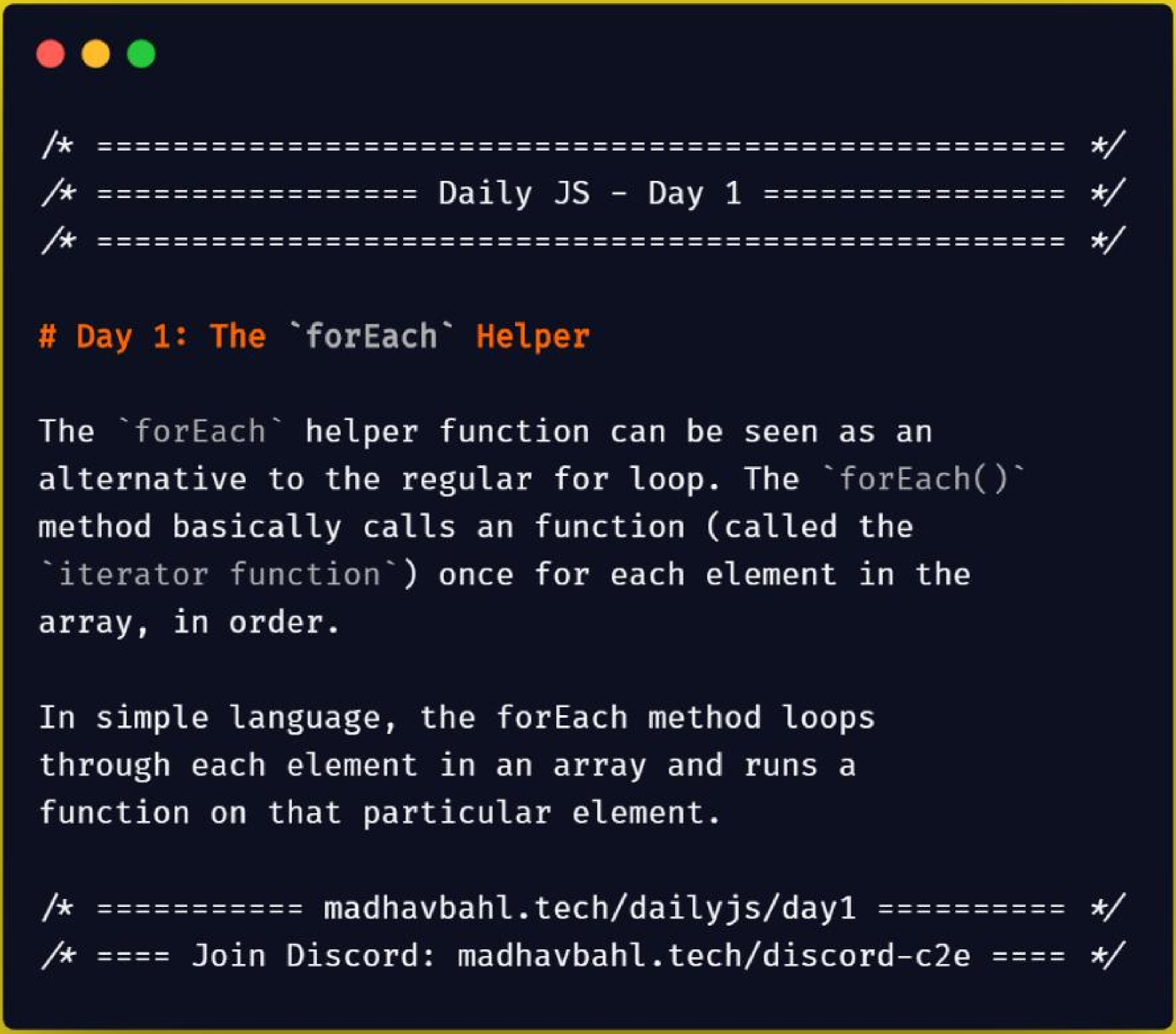
THE LEAN PROGRAMMER | CodeTo Express

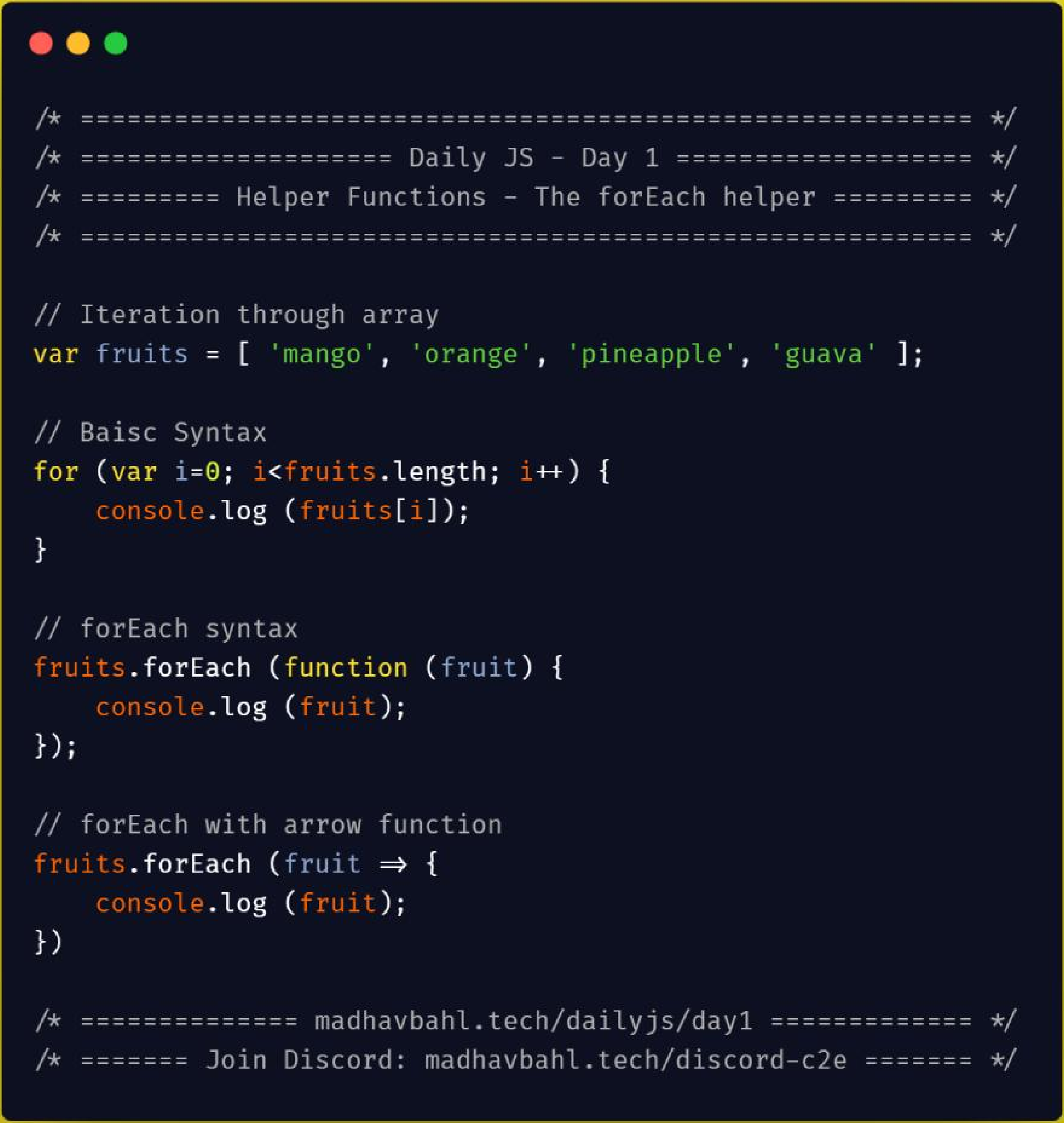
Learn and understand

ES6 HELPERS

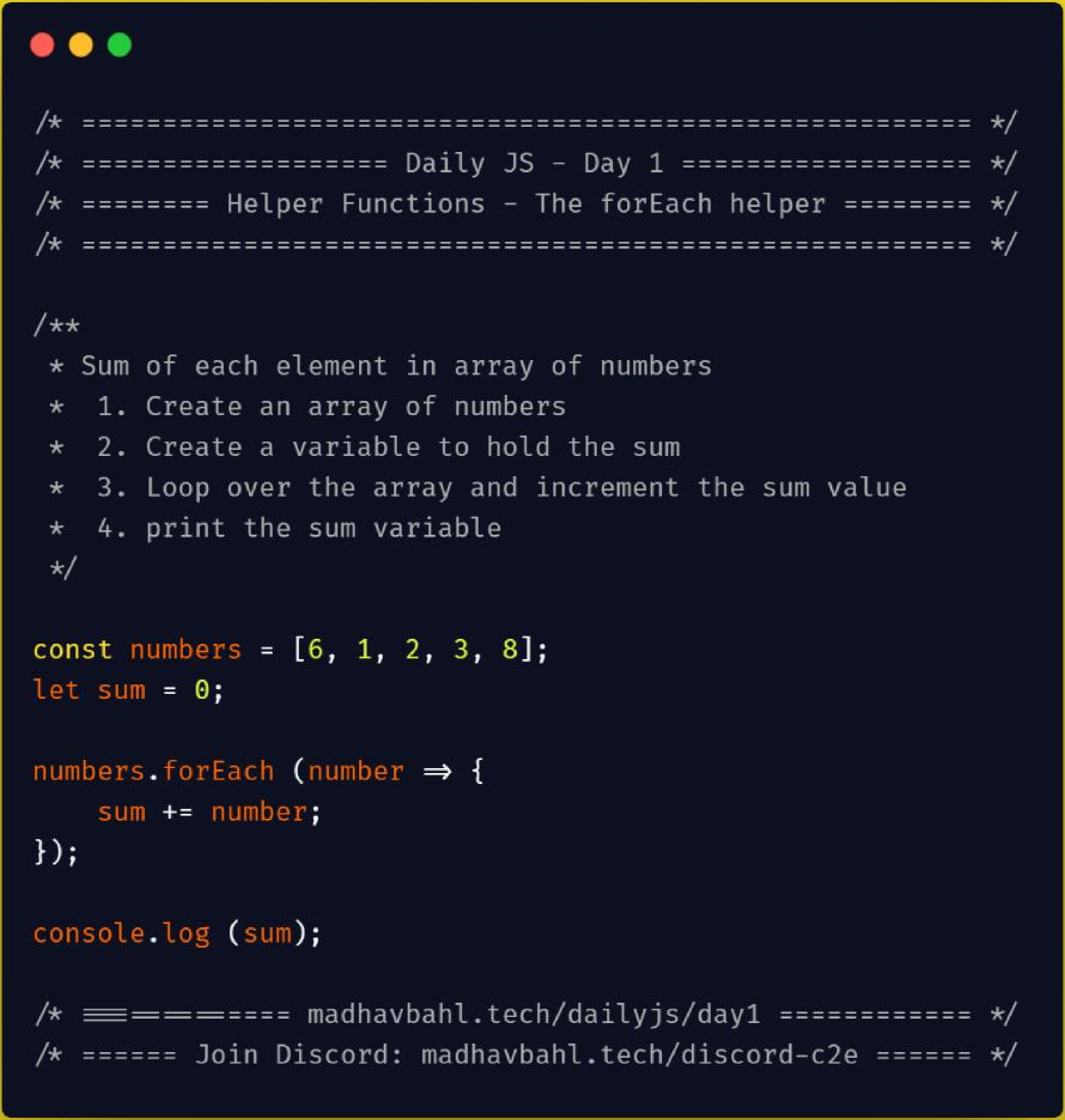
DAILY JS



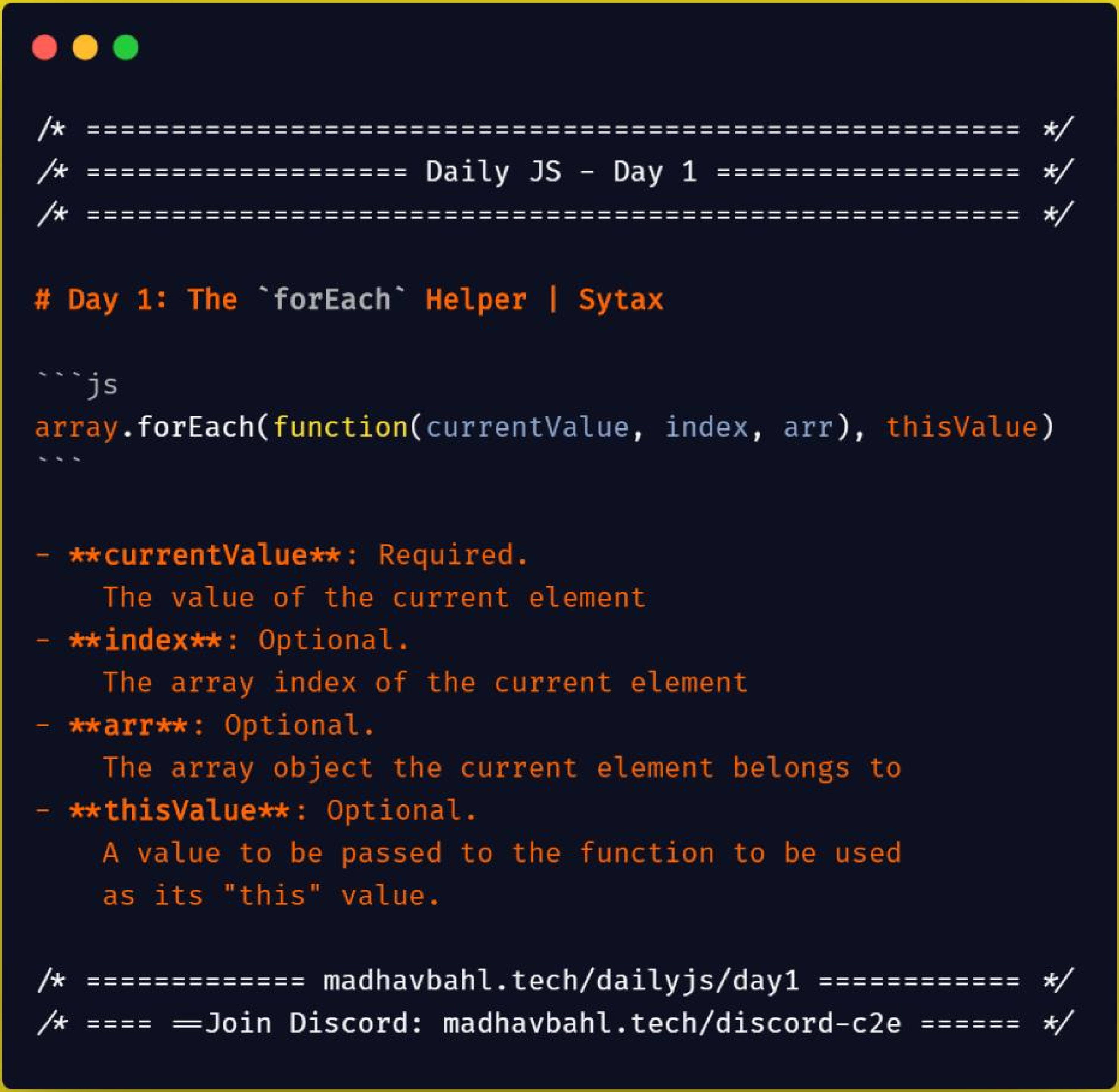
```
/* =================================================== */
/* ================= Daily JS - Day 1 ================ */
/* =================================================== */

# Day 1: The `forEach` Helper

The `forEach` helper function can be seen as an
alternative to the regular for loop. The `forEach()`
method basically calls an function (called the
`iterator function`) once for each element in the
array, in order.

In simple language, the forEach method loops
through each element in an array and runs a
function on that particular element.

/* =========== madhavbahl.tech/dailyjs/day1 ========== */
/* ==== Join Discord: madhavbahl.tech/discord-c2e ==== */
```



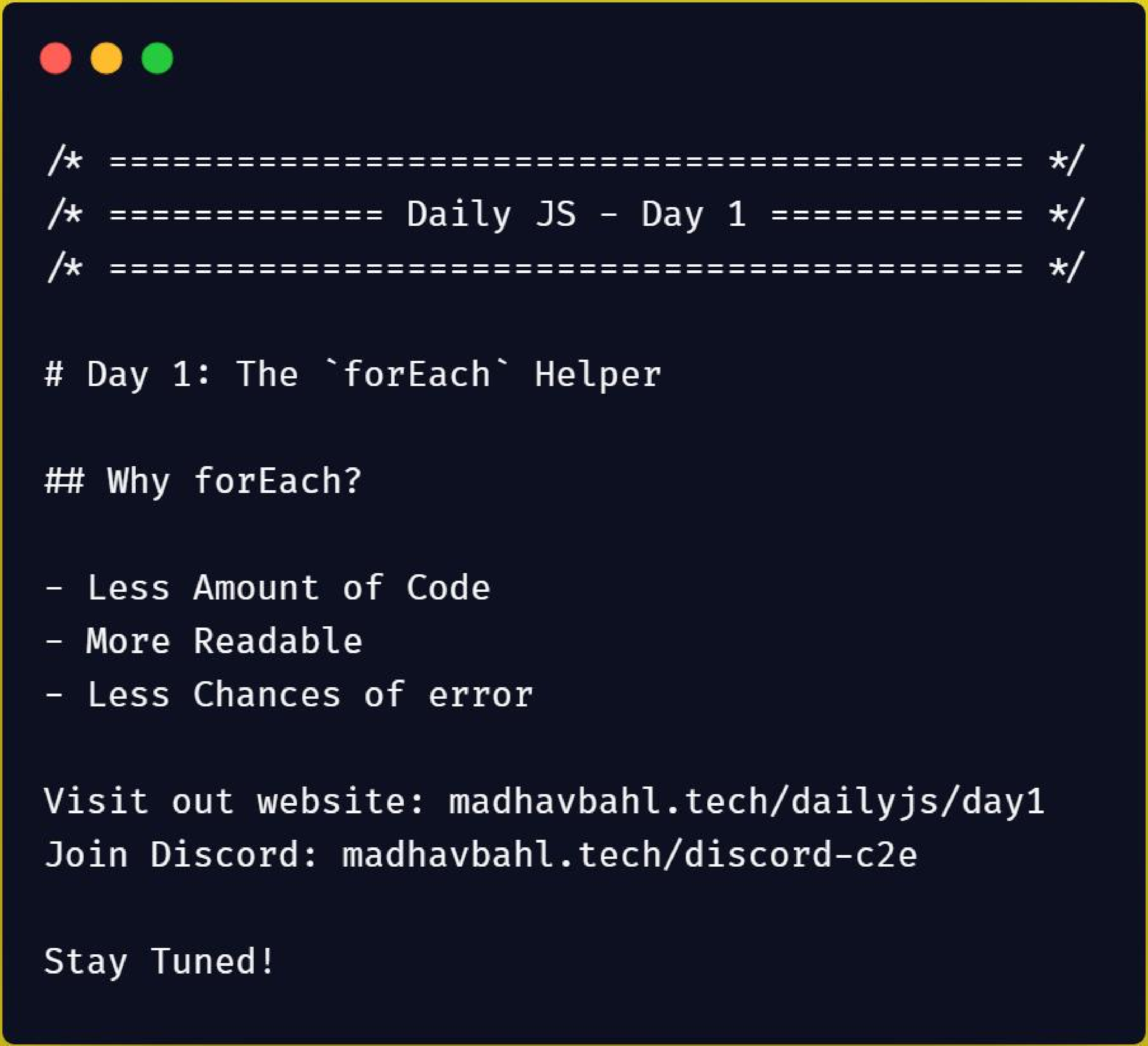
```
/* ========================================================= */
/* ==================== Daily JS - Day 1 =================== */
/* ========= Helper Functions - The forEach helper ========= */
/* ========================================================= */

// Iteration through array
var fruits = [ 'mango', 'orange', 'pineapple', 'guava' ];

// Baisc Syntax
for (var i=0; i<fruits.length; i++) {
    console.log (fruits[i]);
}

// forEach syntax
fruits.forEach (function (fruit) {
    console.log (fruit);
});

// forEach with arrow function
fruits.forEach (fruit => {
    console.log (fruit);
})

/* ============== madhavbahl.tech/dailyjs/day1 ============= */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ======= */
```



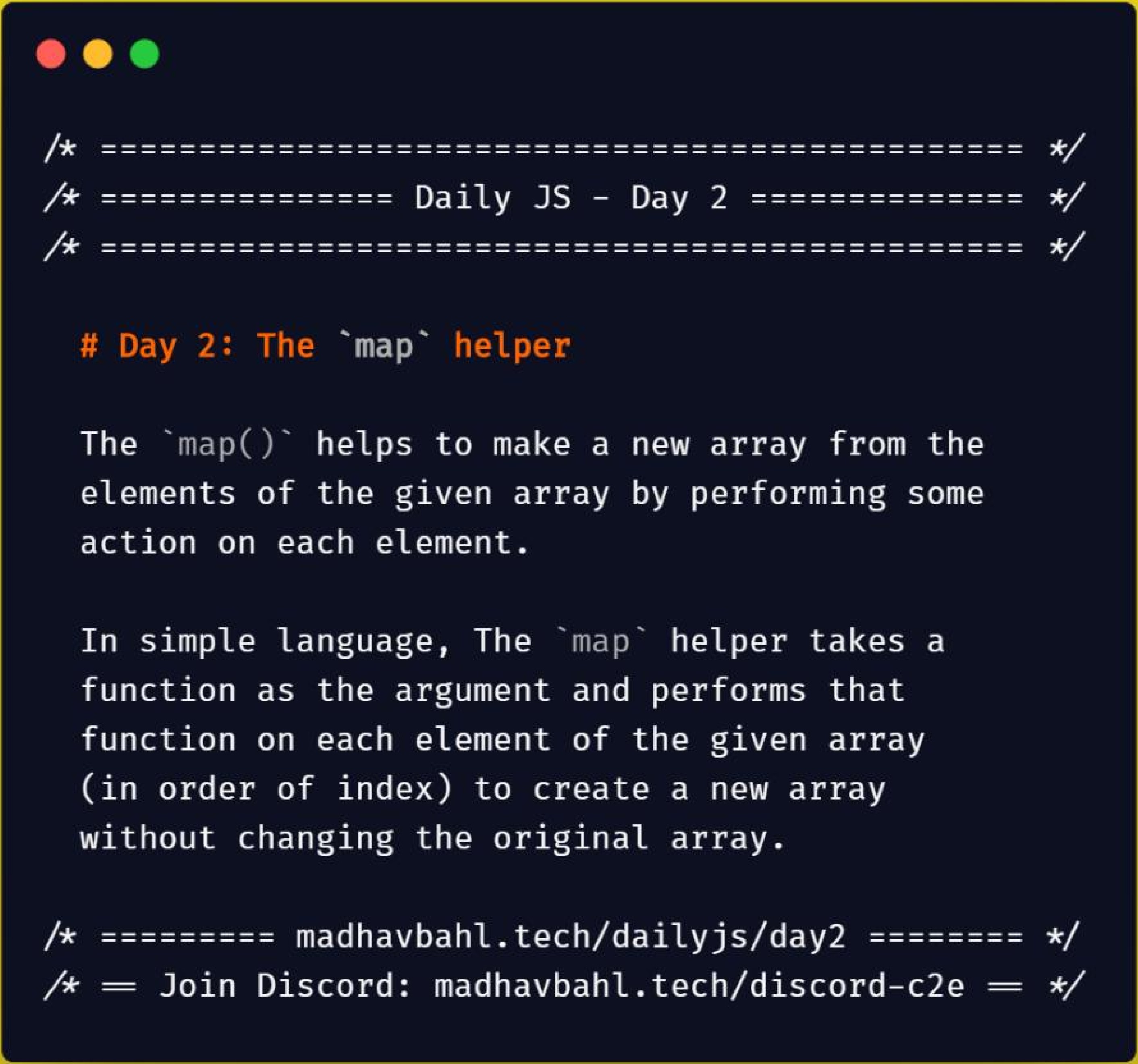
```
/* ======================================================= */
/* =================== Daily JS - Day 1 ================== */
/* ======== Helper Functions - The forEach helper ======== */
/* ======================================================= */

/**
 * Sum of each element in array of numbers
 *  1. Create an array of numbers
 *  2. Create a variable to hold the sum
 *  3. Loop over the array and increment the sum value
 *  4. print the sum variable
 */

const numbers = [6, 1, 2, 3, 8];
let sum = 0;

numbers.forEach (number => {
    sum += number;
});

console.log (sum);

/* ============= madhavbahl.tech/dailyjs/day1 ============ */
/* ====== Join Discord: madhavbahl.tech/discord-c2e ====== */
```



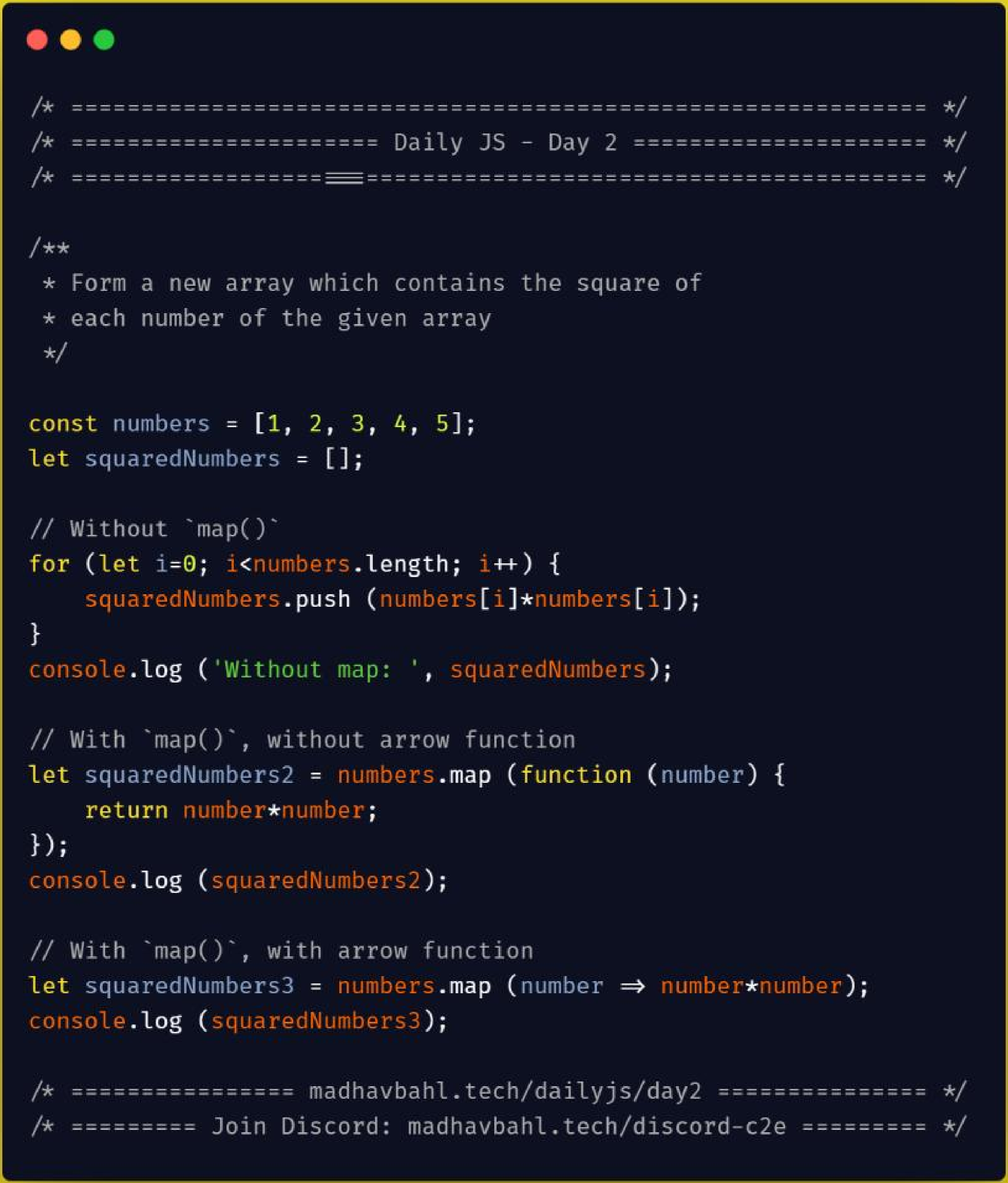
~~~
/* ======================================================= */
/* =================== Daily JS - Day 1 ================== */
/* ======================================================= */

# Day 1: The `forEach` Helper | Sytax

```js
array.forEach(function(currentValue, index, arr), thisValue)
```


- **currentValue**: Required.
    The value of the current element
- **index**: Optional.
    The array index of the current element
- **arr**: Optional.
    The array object the current element belongs to
- **thisValue**: Optional.
    A value to be passed to the function to be used
    as its "this" value.

/* ============= madhavbahl.tech/dailyjs/day1 ============ */
/* ==== ==Join Discord: madhavbahl.tech/discord-c2e ====== */
~~~



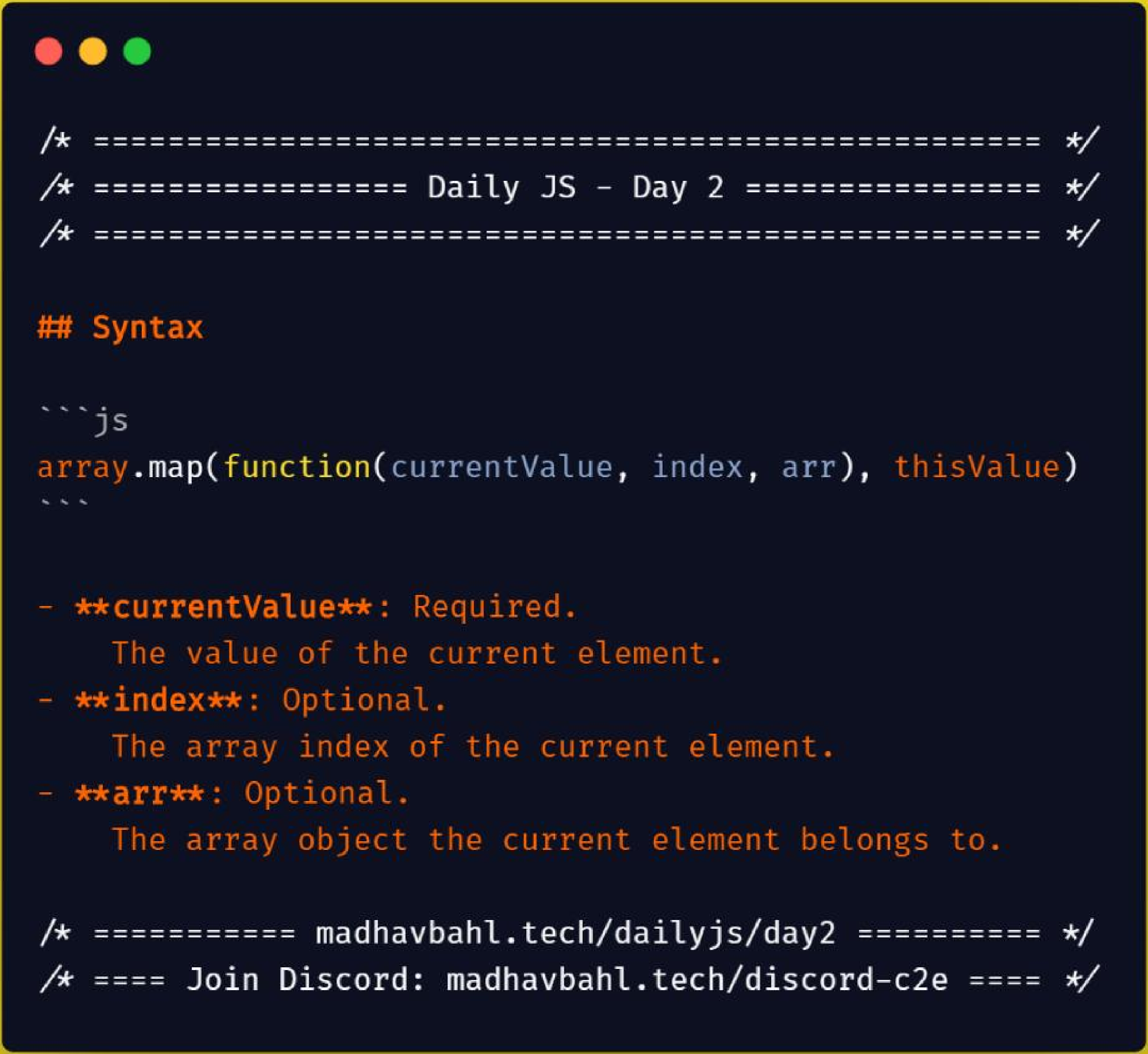
```
/* =========================================== */
/* ============= Daily JS - Day 1 ============ */
/* =========================================== */

# Day 1: The `forEach` Helper

## Why forEach?

- Less Amount of Code
- More Readable
- Less Chances of error

Visit out website: madhavbahl.tech/dailyjs/day1
Join Discord: madhavbahl.tech/discord-c2e

Stay Tuned!
```



```
/* =============================================== */
/* =============== Daily JS - Day 2 ============== */
/* =============================================== */

  # Day 2: The `map` helper

  The `map()` helps to make a new array from the
  elements of the given array by performing some
  action on each element.

  In simple language, The `map` helper takes a
  function as the argument and performs that
  function on each element of the given array
  (in order of index) to create a new array
  without changing the original array.

/* ========= madhavbahl.tech/dailyjs/day2 ======== */
/* == Join Discord: madhavbahl.tech/discord-c2e == */
```



```
/* ============================================================= */
/* ====================== Daily JS - Day 2 ===================== */
/* ============================================================= */

/**
 * Form a new array which contains the square of
 * each number of the given array
 */

const numbers = [1, 2, 3, 4, 5];
let squaredNumbers = [];

// Without `map()`
for (let i=0; i<numbers.length; i++) {
    squaredNumbers.push (numbers[i]*numbers[i]);
}
console.log ('Without map: ', squaredNumbers);

// With `map()`, without arrow function
let squaredNumbers2 = numbers.map (function (number) {
    return number*number;
});
console.log (squaredNumbers2);

// With `map()`, with arrow function
let squaredNumbers3 = numbers.map (number => number*number);
console.log (squaredNumbers3);

/* ================ madhavbahl.tech/dailyjs/day2 =============== */
/* ========= Join Discord: madhavbahl.tech/discord-c2e ========= */
```

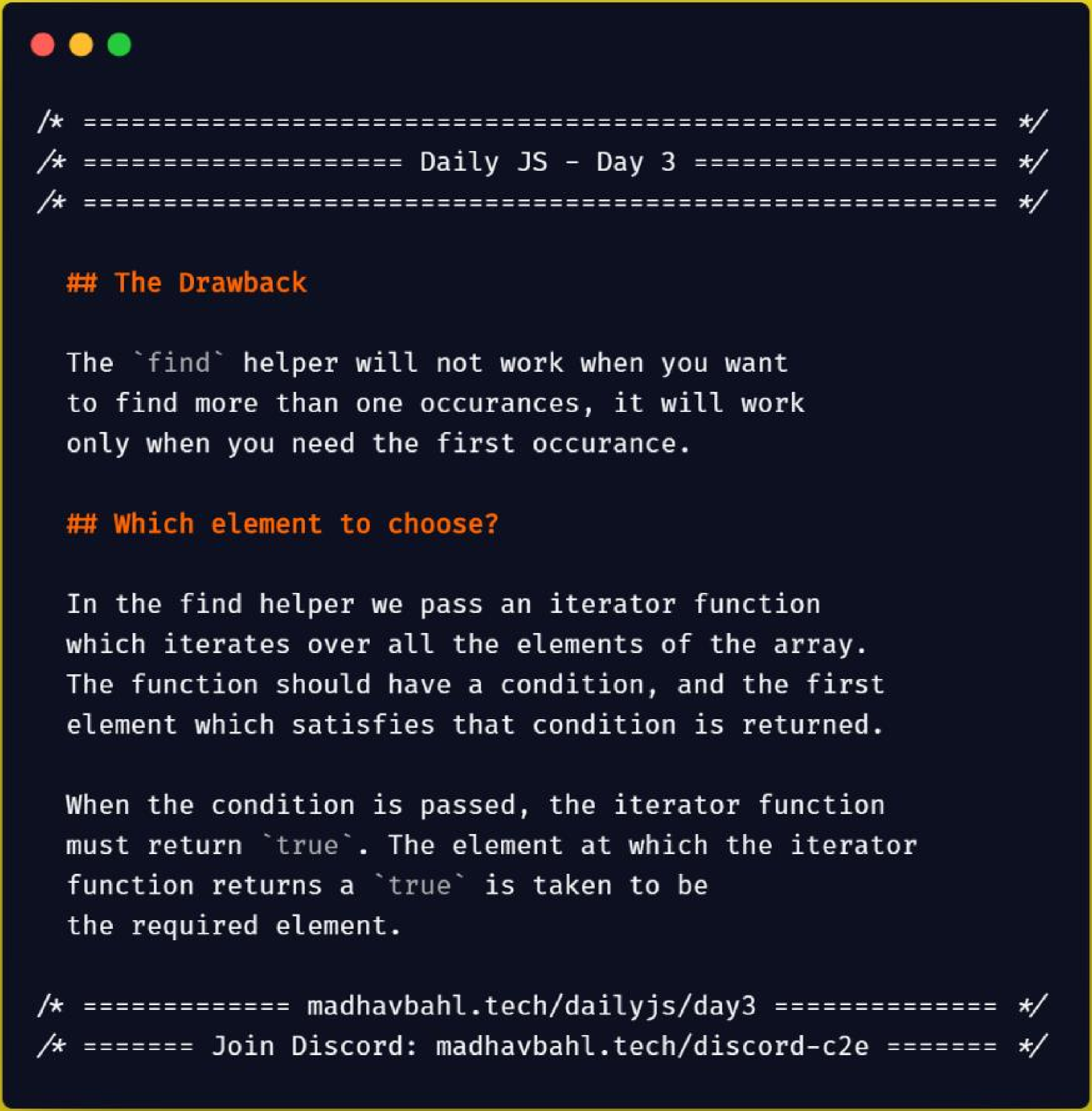


~~~
/* =================================================== */
/* ================= Daily JS - Day 2 ================ */
/* =================================================== */

## Syntax

```js
array.map(function(currentValue, index, arr), thisValue)
```


- **currentValue**: Required.
    The value of the current element.
- **index**: Optional.
    The array index of the current element.
- **arr**: Optional.
    The array object the current element belongs to.

/* =========== madhavbahl.tech/dailyjs/day2 ========== */
/* ==== Join Discord: madhavbahl.tech/discord-c2e ==== */
~~~

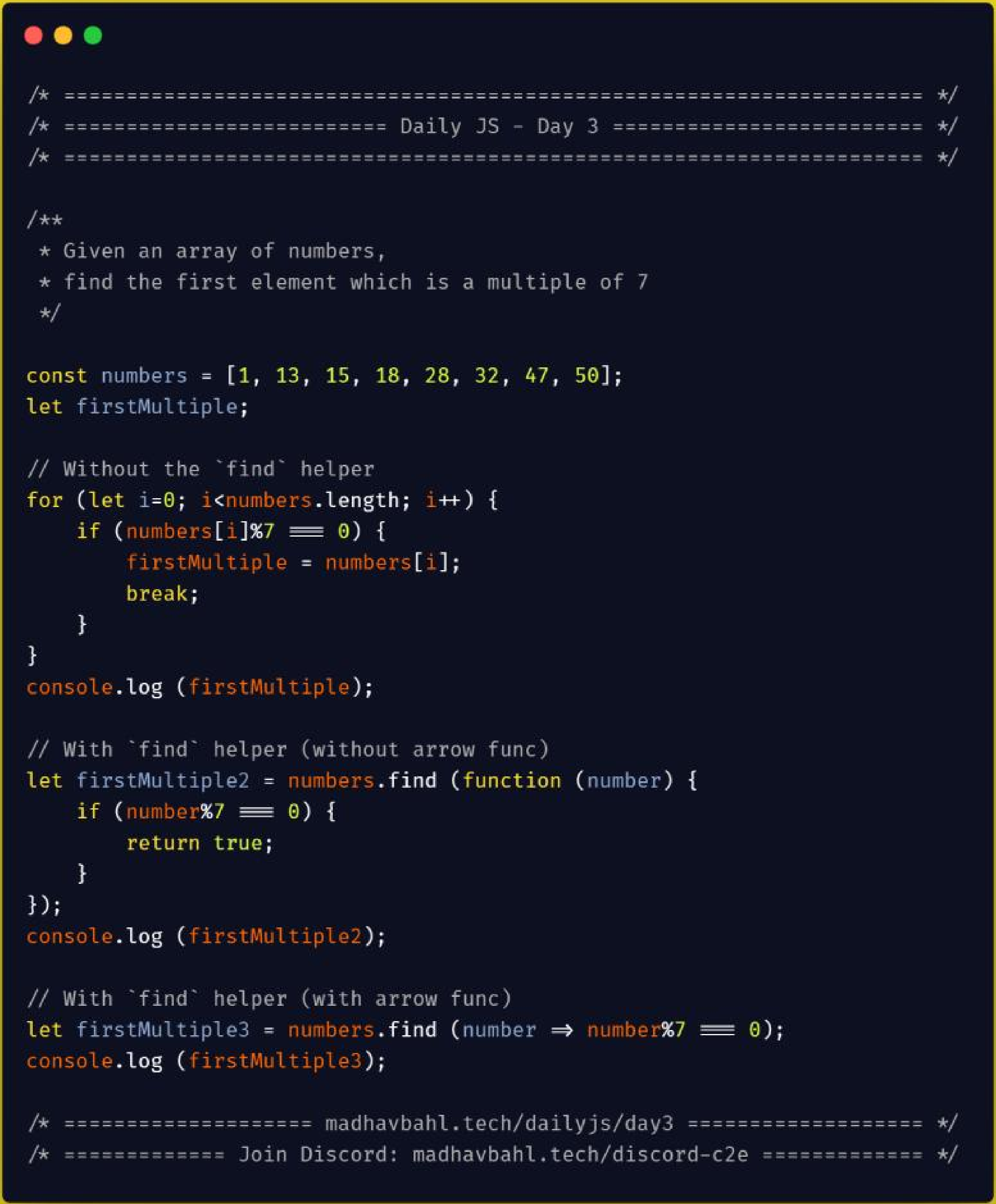
```
/* ============================================================ */
/* ==================== Daily JS - Day 2 ==================== */
/* ============================================================ */

  ## Why creating a new array is a good practise

  When the code base is small, we tend to perform all
  the actions on the same data structure and feel free
  to change it easily because we know wherever it is
  being used, but when the code base becomes huge and
  we are not sure whether a given data structure is
  being used at other place or not, it is not good to
  change it directly as it might lead to some
  inconsistencies at the other places where it is being used

  Generally, we try not to mutate any given data structure,
  because it's possible that the given data structure could
  have been used in some other place as well and changing
  it might break our code.

  The `map` helper comes to the rescue at such places,
  it applies a particular function to each element in
  an array and return a new array without changing
  the original one.

/* ============== madhavbahl.tech/dailyjs/day2 ============= */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ======= */
```

```
/* ========================================================== */
/* ==================== Daily JS - Day 2 ==================== */
/* ========================================================== */

  ## Homework for you (some more questions)

  1. Make an array of numbers that are doubles
     of the first array

  2. Take an array of numbers and convert each element
     into a string.

  3. Capitalize the first letter of each name in
     the names array

  4. Given an of news headlines (take some random data),
     and enclose the headlines in `<h1>` tag and append them
     to DOM to display the news headlines

/* ============== madhavbahl.tech/dailyjs/day2 ============== */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ======= */
```

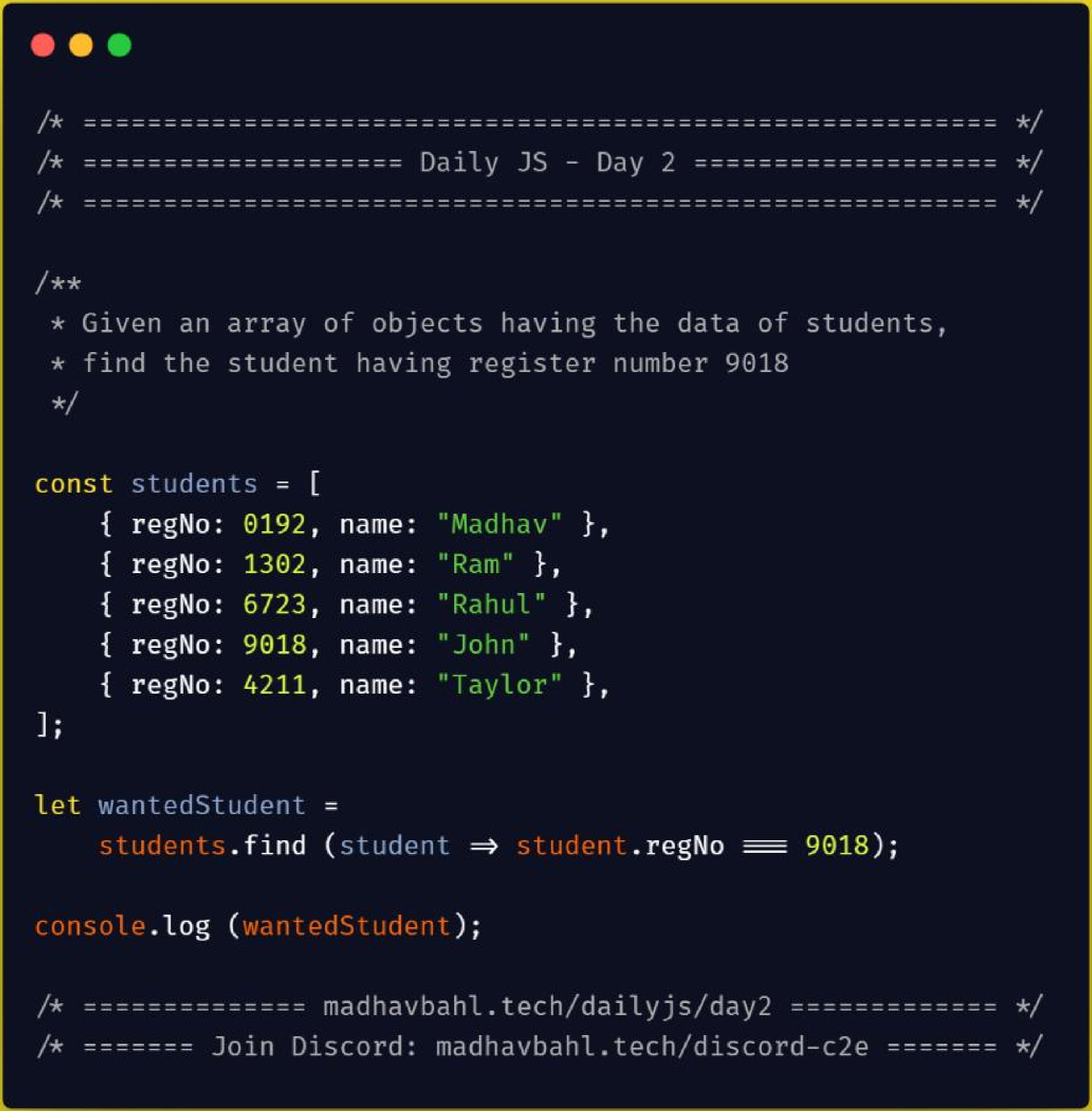
```
/* =================================================== */
/* ================= Daily JS - Day 3 ================ */
/* =================================================== */

  # Day 3: The `find` helper

  The `find` helper loops through an array to
  look for a particular element (first occurance),
  which passes the given criteria.

  It runs the provided function over each element,
  and if the particular element passes the criteria,
  it is returned. If no element found, `undefined`
  is returned.

  This helper method does not change the
  original array.

/* ========== madhavbahl.tech/dailyjs/day3 =========== */
/* ==== Join Discord: madhavbahl.tech/discord-c2e ==== */
```



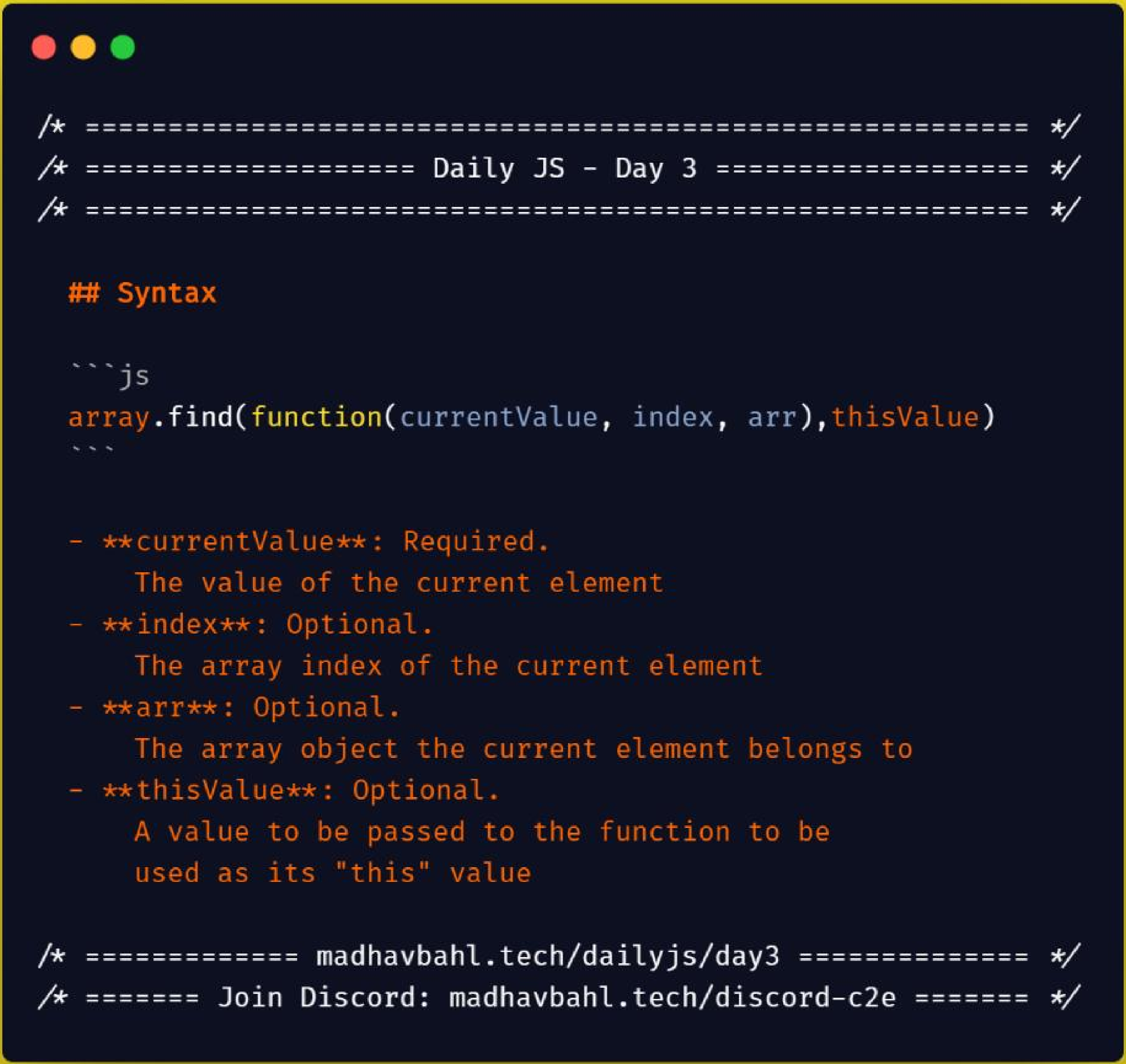
```
/* ========================================================= */
/* ==================== Daily JS - Day 3 =================== */
/* ========================================================= */

  ## The Drawback

  The `find` helper will not work when you want
  to find more than one occurances, it will work
  only when you need the first occurance.

  ## Which element to choose?

  In the find helper we pass an iterator function
  which iterates over all the elements of the array.
  The function should have a condition, and the first
  element which satisfies that condition is returned.

  When the condition is passed, the iterator function
  must return `true`. The element at which the iterator
  function returns a `true` is taken to be
  the required element.

/* ============= madhavbahl.tech/dailyjs/day3 ============== */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ======= */
```



```javascript
/* ===================================================================== */
/* ========================== Daily JS - Day 3 ========================= */
/* ===================================================================== */

/**
 * Given an array of numbers,
 * find the first element which is a multiple of 7
 */

const numbers = [1, 13, 15, 18, 28, 32, 47, 50];
let firstMultiple;

// Without the `find` helper
for (let i=0; i<numbers.length; i++) {
    if (numbers[i]%7 === 0) {
        firstMultiple = numbers[i];
        break;
    }
}
console.log (firstMultiple);

// With `find` helper (without arrow func)
let firstMultiple2 = numbers.find (function (number) {
    if (number%7 === 0) {
        return true;
    }
});
console.log (firstMultiple2);

// With `find` helper (with arrow func)
let firstMultiple3 = numbers.find (number => number%7 === 0);
console.log (firstMultiple3);

/* ==================== madhavbahl.tech/dailyjs/day3 =================== */
/* ============= Join Discord: madhavbahl.tech/discord-c2e ============= */
```

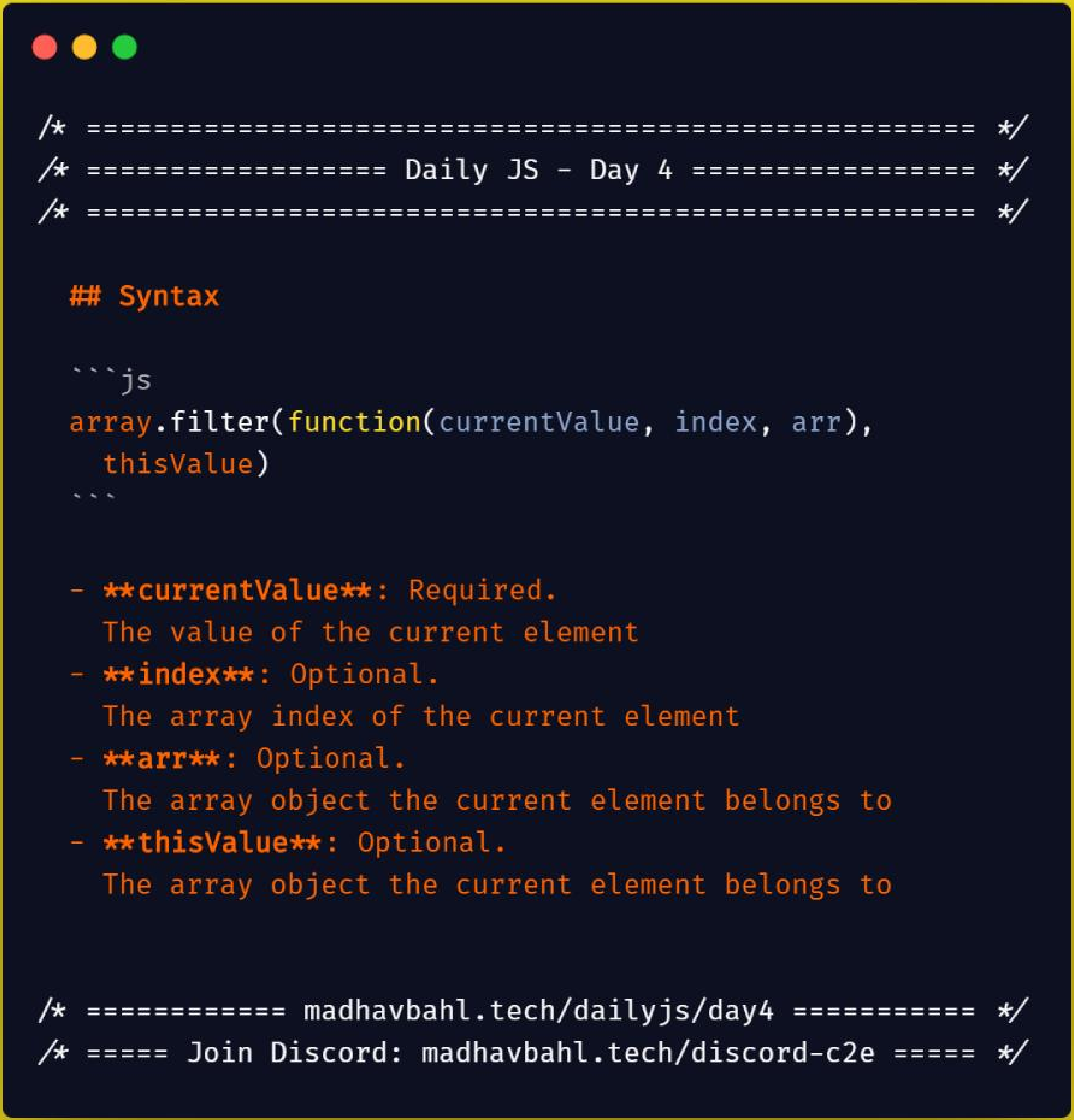


```javascript
/* ========================================================= */
/* ==================== Daily JS - Day 2 =================== */
/* ========================================================= */

/**
 * Given an array of objects having the data of students,
 * find the student having register number 9018
 */

const students = [
    { regNo: 0192, name: "Madhav" },
    { regNo: 1302, name: "Ram" },
    { regNo: 6723, name: "Rahul" },
    { regNo: 9018, name: "John" },
    { regNo: 4211, name: "Taylor" },
];

let wantedStudent =
    students.find (student => student.regNo === 9018);

console.log (wantedStudent);

/* ============== madhavbahl.tech/dailyjs/day2 ============= */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ======= */
```

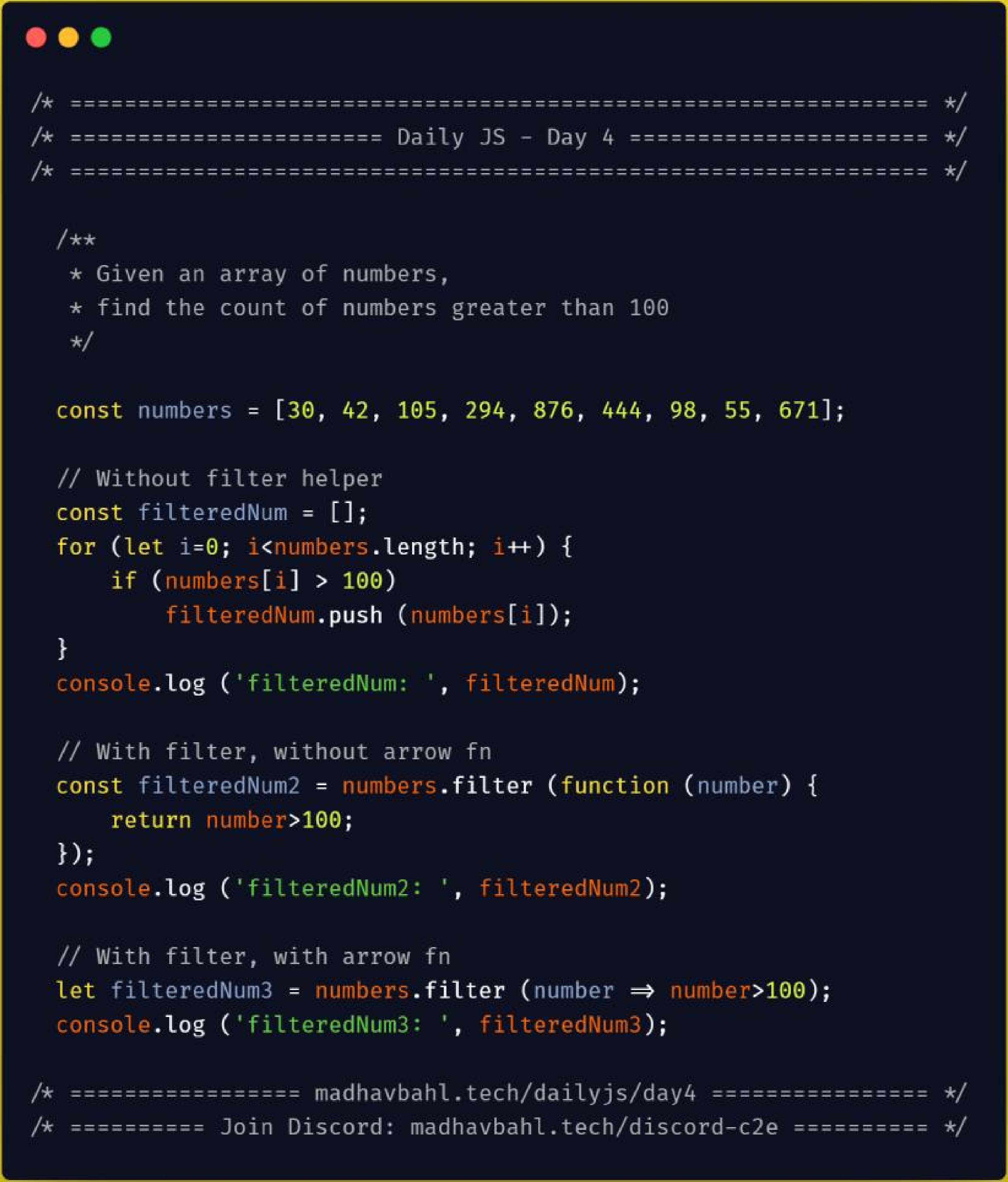


~~~
/* ========================================================= */
/* ==================== Daily JS - Day 3 =================== */
/* ========================================================= */

  ## Syntax

  ```js
  array.find(function(currentValue, index, arr),thisValue)
  ```


  - **currentValue**: Required.
      The value of the current element
  - **index**: Optional.
      The array index of the current element
  - **arr**: Optional.
      The array object the current element belongs to
  - **thisValue**: Optional.
      A value to be passed to the function to be
      used as its "this" value

/* ============= madhavbahl.tech/dailyjs/day3 ============== */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ======= */
~~~

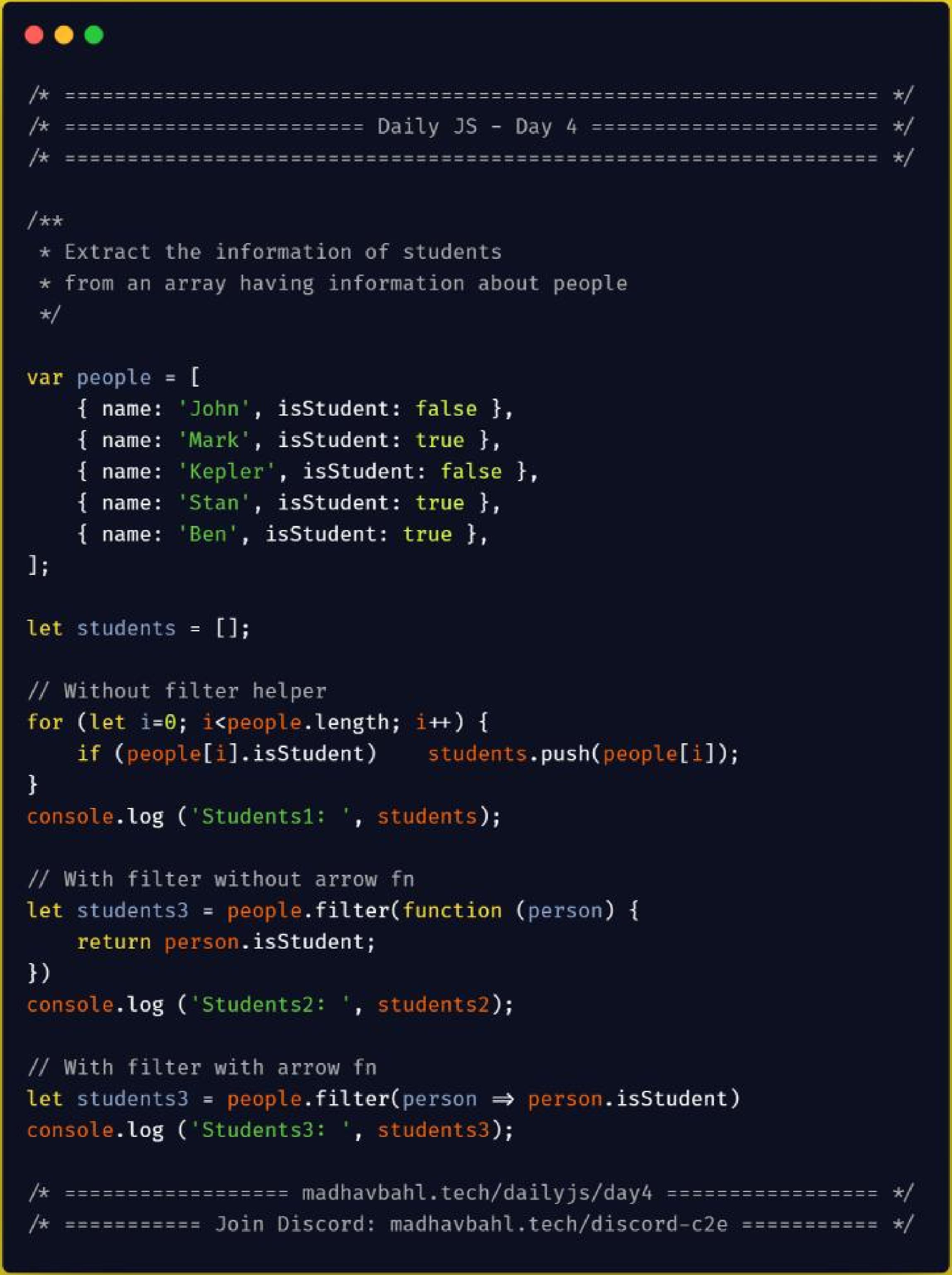
```
/* =================================================== */
/* ================= Daily JS - Day 4 ================ */
/* =================================================== */

  # Day 4: The `filter` helper

  As the name suggests, the `filter` helper is
  used to filter out some specific elements from
  an array of elements.

  It creates a new array from the elements of the
  original array which pass the given criteria.

  The criteria is generally given as a condition in
  the iterator function.

  Unlike `find` helper, `filter` returns an arrow
  of elements which pass the criteria.

/* =========== madhavbahl.tech/dailyjs/day4 ========== */
/* ==== Join Discord: madhavbahl.tech/discord-c2e ==== */
```



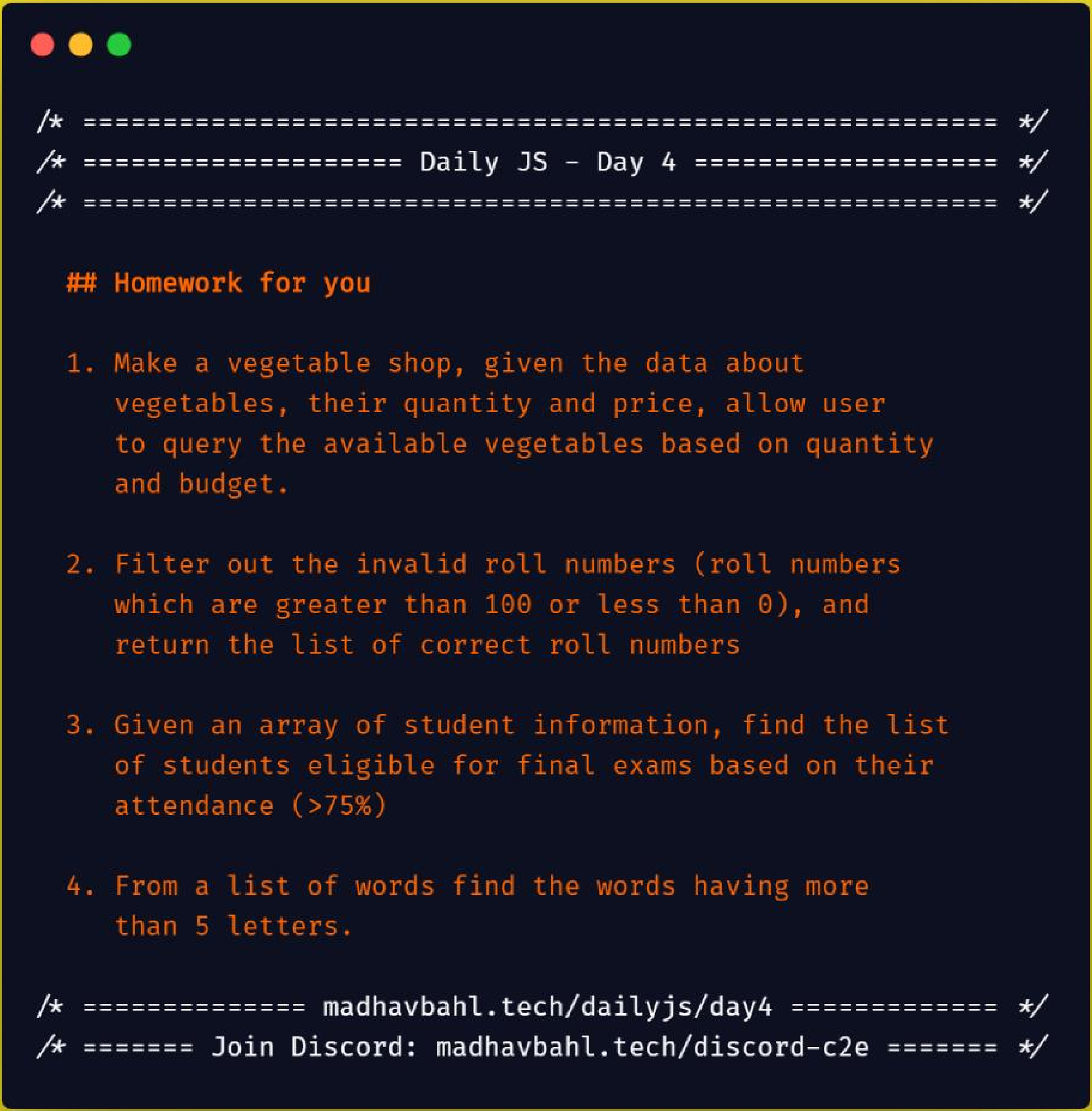
~~~
/* ===================================================== */
/* ================== Daily JS - Day 4 ================= */
/* ===================================================== */

  ## Syntax

  ```js
  array.filter(function(currentValue, index, arr),
    thisValue)
  ```


  - **currentValue**: Required.
    The value of the current element
  - **index**: Optional.
    The array index of the current element
  - **arr**: Optional.
    The array object the current element belongs to
  - **thisValue**: Optional.
    The array object the current element belongs to



/* ============ madhavbahl.tech/dailyjs/day4 =========== */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ===== */
~~~



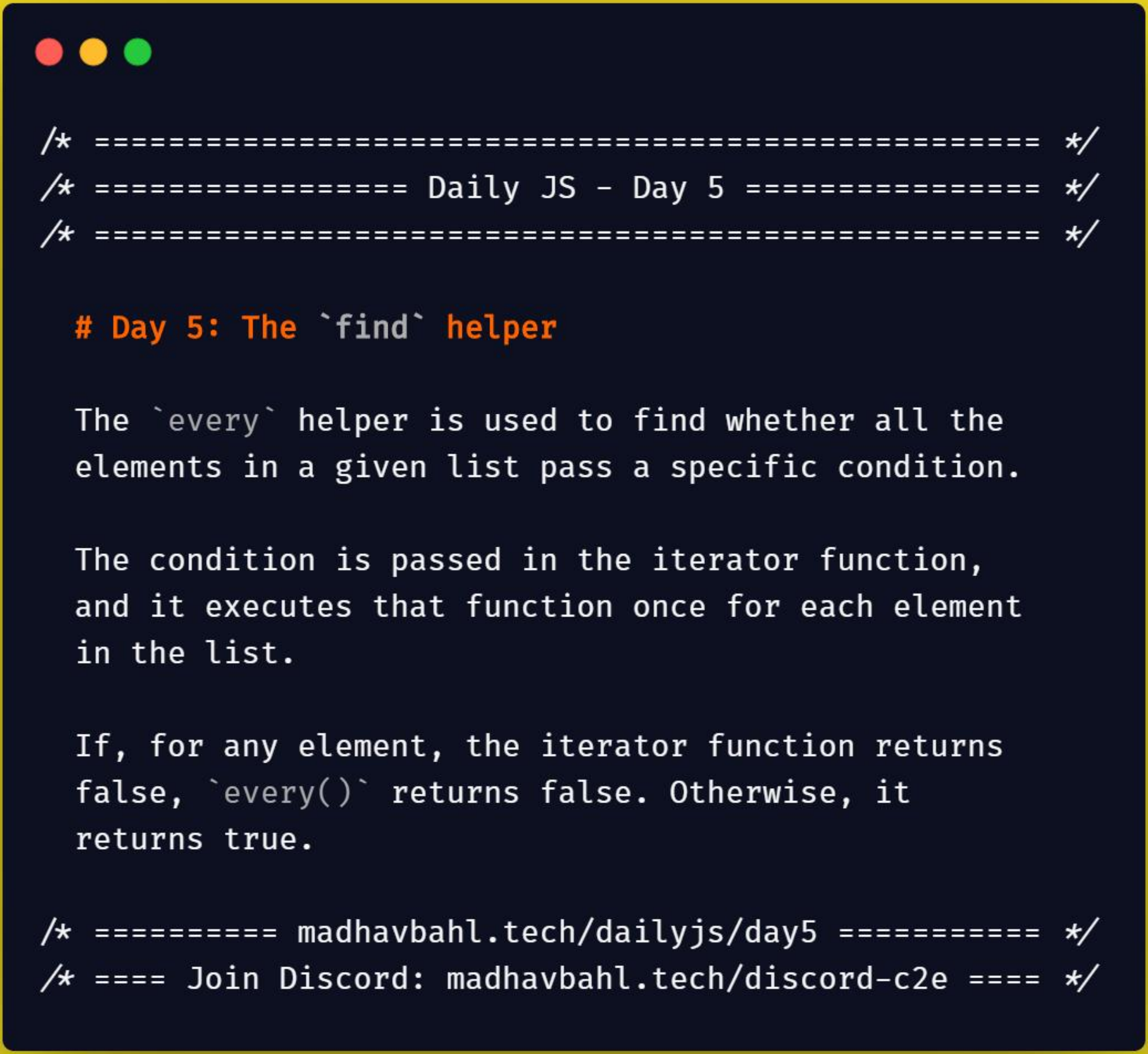
```
/* ================================================================ */
/* ======================= Daily JS - Day 4 ======================= */
/* ================================================================ */

  /**
   * Given an array of numbers,
   * find the count of numbers greater than 100
   */

  const numbers = [30, 42, 105, 294, 876, 444, 98, 55, 671];

  // Without filter helper
  const filteredNum = [];
  for (let i=0; i<numbers.length; i++) {
      if (numbers[i] > 100)
          filteredNum.push (numbers[i]);
  }
  console.log ('filteredNum: ', filteredNum);

  // With filter, without arrow fn
  const filteredNum2 = numbers.filter (function (number) {
      return number>100;
  });
  console.log ('filteredNum2: ', filteredNum2);

  // With filter, with arrow fn
  let filteredNum3 = numbers.filter (number => number>100);
  console.log ('filteredNum3: ', filteredNum3);

/* ================= madhavbahl.tech/dailyjs/day4 ================= */
/* ========== Join Discord: madhavbahl.tech/discord-c2e ========== */
```



```javascript
/* ================================================================= */
/* ======================== Daily JS - Day 4 ======================= */
/* ================================================================= */

/**
 * Extract the information of students
 * from an array having information about people
 */

var people = [
    { name: 'John', isStudent: false },
    { name: 'Mark', isStudent: true },
    { name: 'Kepler', isStudent: false },
    { name: 'Stan', isStudent: true },
    { name: 'Ben', isStudent: true },
];

let students = [];

// Without filter helper
for (let i=0; i<people.length; i++) {
    if (people[i].isStudent)    students.push(people[i]);
}
console.log ('Students1: ', students);

// With filter without arrow fn
let students3 = people.filter(function (person) {
    return person.isStudent;
})
console.log ('Students2: ', students2);

// With filter with arrow fn
let students3 = people.filter(person => person.isStudent)
console.log ('Students3: ', students3);

/* ================== madhavbahl.tech/dailyjs/day4 ================= */
/* =========== Join Discord: madhavbahl.tech/discord-c2e =========== */
```



```
/* ========================================================= */
/* ==================== Daily JS - Day 4 =================== */
/* ========================================================= */

  ## Homework for you

  1. Make a vegetable shop, given the data about
     vegetables, their quantity and price, allow user
     to query the available vegetables based on quantity
     and budget.

  2. Filter out the invalid roll numbers (roll numbers
     which are greater than 100 or less than 0), and
     return the list of correct roll numbers

  3. Given an array of student information, find the list
     of students eligible for final exams based on their
     attendance (>75%)

  4. From a list of words find the words having more
     than 5 letters.

/* ============== madhavbahl.tech/dailyjs/day4 ============= */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ======= */
```



```
/* ============================================== */
/* ================= Daily JS - Day 5 ================= */
/* ============================================== */
```

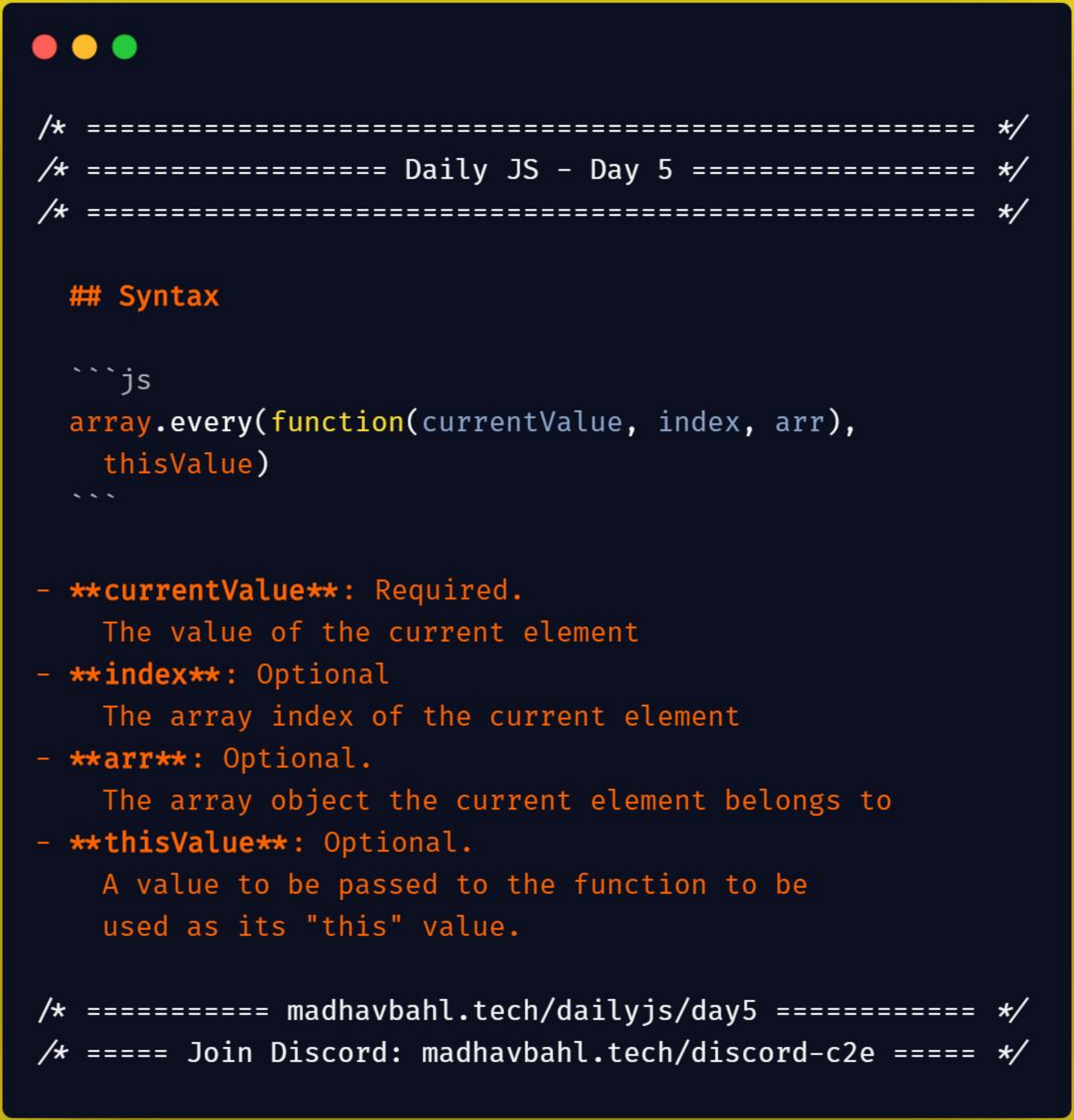
Day 5: The `find` helper

The `every` helper is used to find whether all the elements in a given list pass a specific condition.

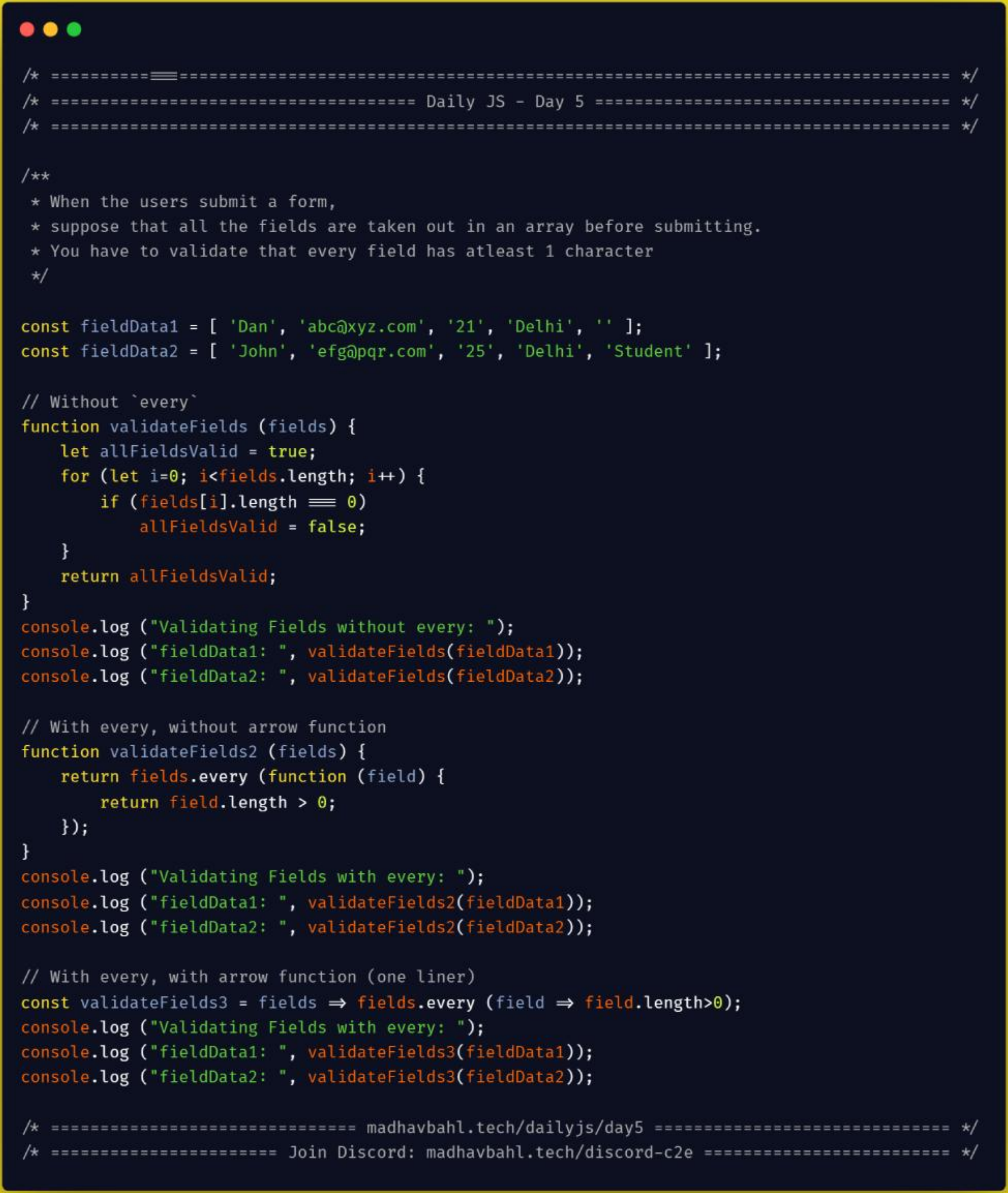
The condition is passed in the iterator function, and it executes that function once for each element in the list.

If, for any element, the iterator function returns false, `every()` returns false. Otherwise, it returns true.

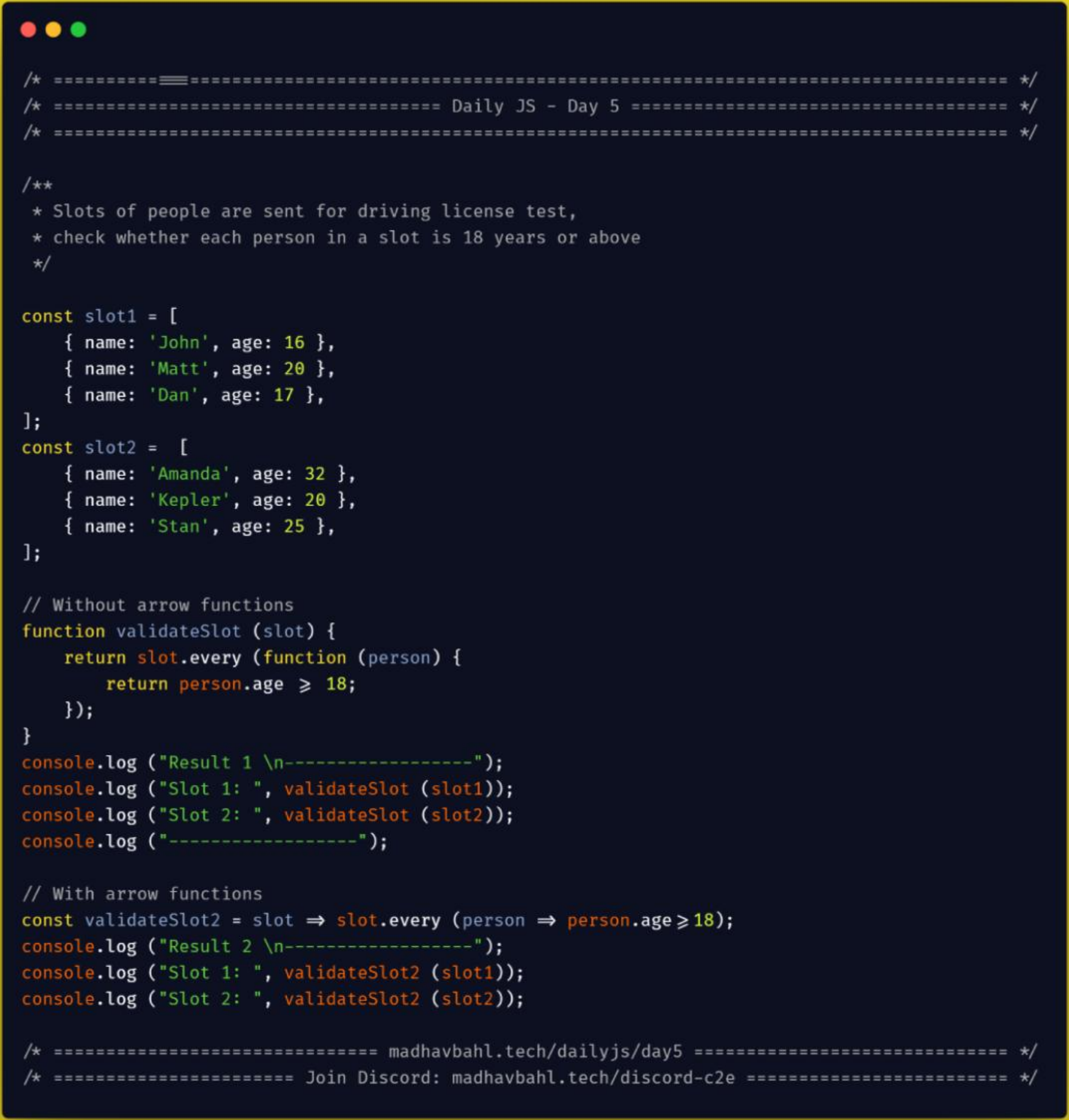
```
/* ========== madhavbahl.tech/dailyjs/day5 =========== */
/* ==== Join Discord: madhavbahl.tech/discord-c2e ==== */
```



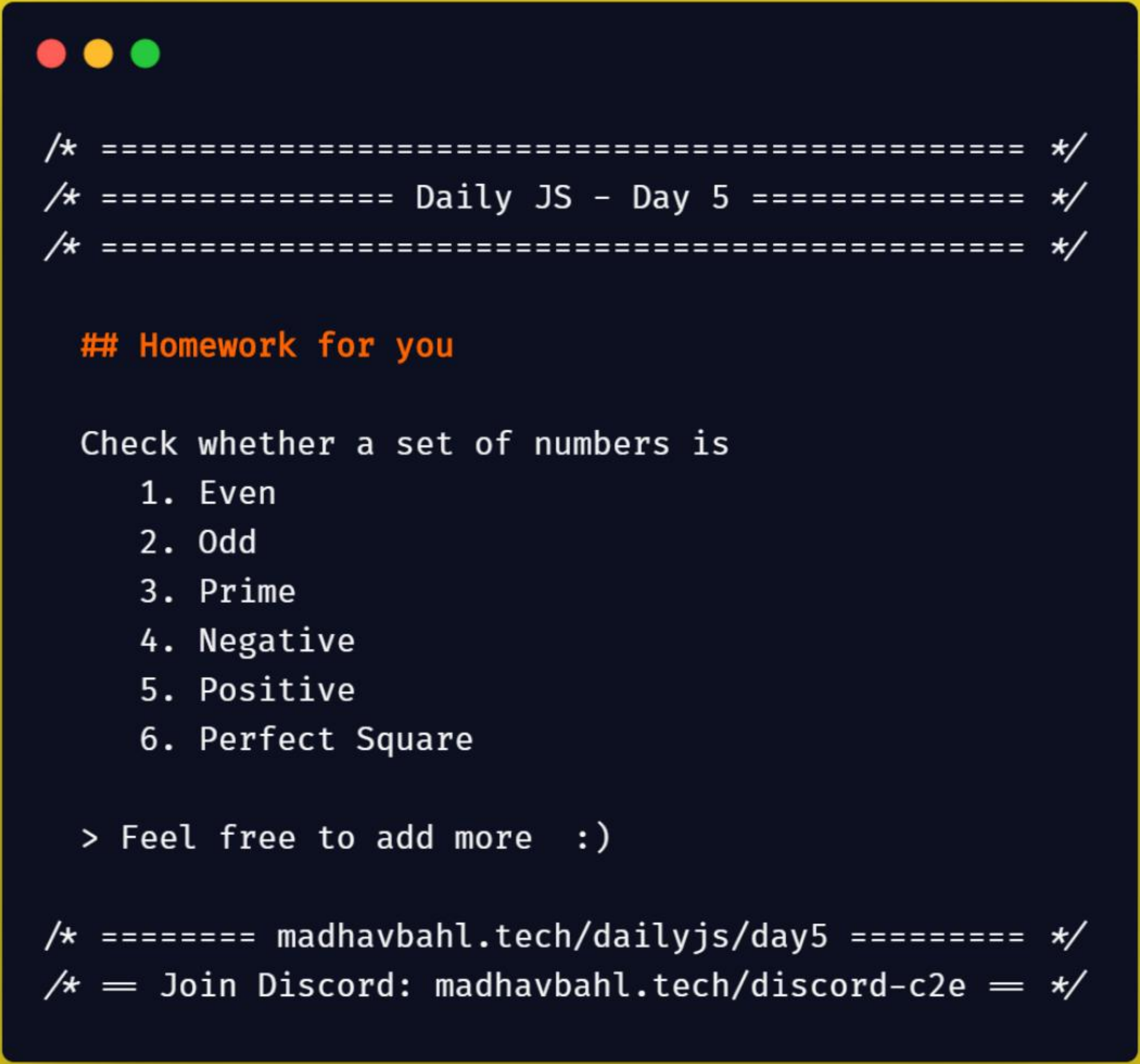
~~~
/* ====================================================== */
/* ================== Daily JS - Day 5 ================== */
/* ====================================================== */

  ## Syntax

  ```js
  array.every(function(currentValue, index, arr),
    thisValue)
  ```


- **currentValue**: Required.
    The value of the current element
- **index**: Optional
    The array index of the current element
- **arr**: Optional.
    The array object the current element belongs to
- **thisValue**: Optional.
    A value to be passed to the function to be
    used as its "this" value.

/* =========== madhavbahl.tech/dailyjs/day5 ============= */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ====== */
~~~



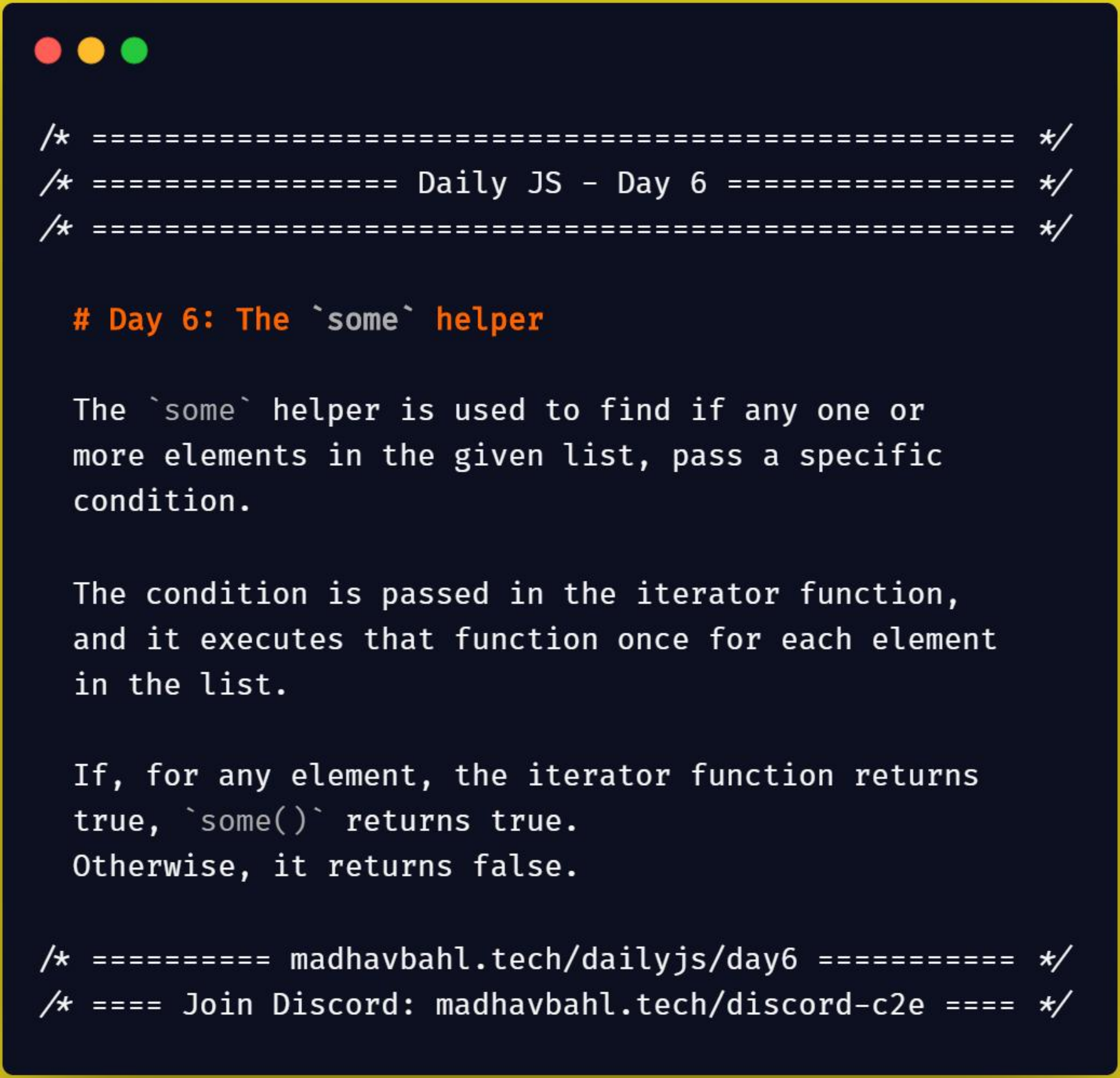
```
/* =========================================================================================== */
/* ===================================== Daily JS - Day 5 ==================================== */
/* =========================================================================================== */

/**
 * When the users submit a form,
 * suppose that all the fields are taken out in an array before submitting.
 * You have to validate that every field has atleast 1 character
 */

const fieldData1 = [ 'Dan', 'abc@xyz.com', '21', 'Delhi', '' ];
const fieldData2 = [ 'John', 'efg@pqr.com', '25', 'Delhi', 'Student' ];

// Without `every`
function validateFields (fields) {
    let allFieldsValid = true;
    for (let i=0; i<fields.length; i++) {
        if (fields[i].length === 0)
            allFieldsValid = false;
    }
    return allFieldsValid;
}
console.log ("Validating Fields without every: ");
console.log ("fieldData1: ", validateFields(fieldData1));
console.log ("fieldData2: ", validateFields(fieldData2));

// With every, without arrow function
function validateFields2 (fields) {
    return fields.every (function (field) {
        return field.length > 0;
    });
}
console.log ("Validating Fields with every: ");
console.log ("fieldData1: ", validateFields2(fieldData1));
console.log ("fieldData2: ", validateFields2(fieldData2));

// With every, with arrow function (one liner)
const validateFields3 = fields => fields.every (field => field.length>0);
console.log ("Validating Fields with every: ");
console.log ("fieldData1: ", validateFields3(fieldData1));
console.log ("fieldData2: ", validateFields3(fieldData2));

/* =============================== madhavbahl.tech/dailyjs/day5 ============================== */
/* ======================= Join Discord: madhavbahl.tech/discord-c2e ========================= */
```



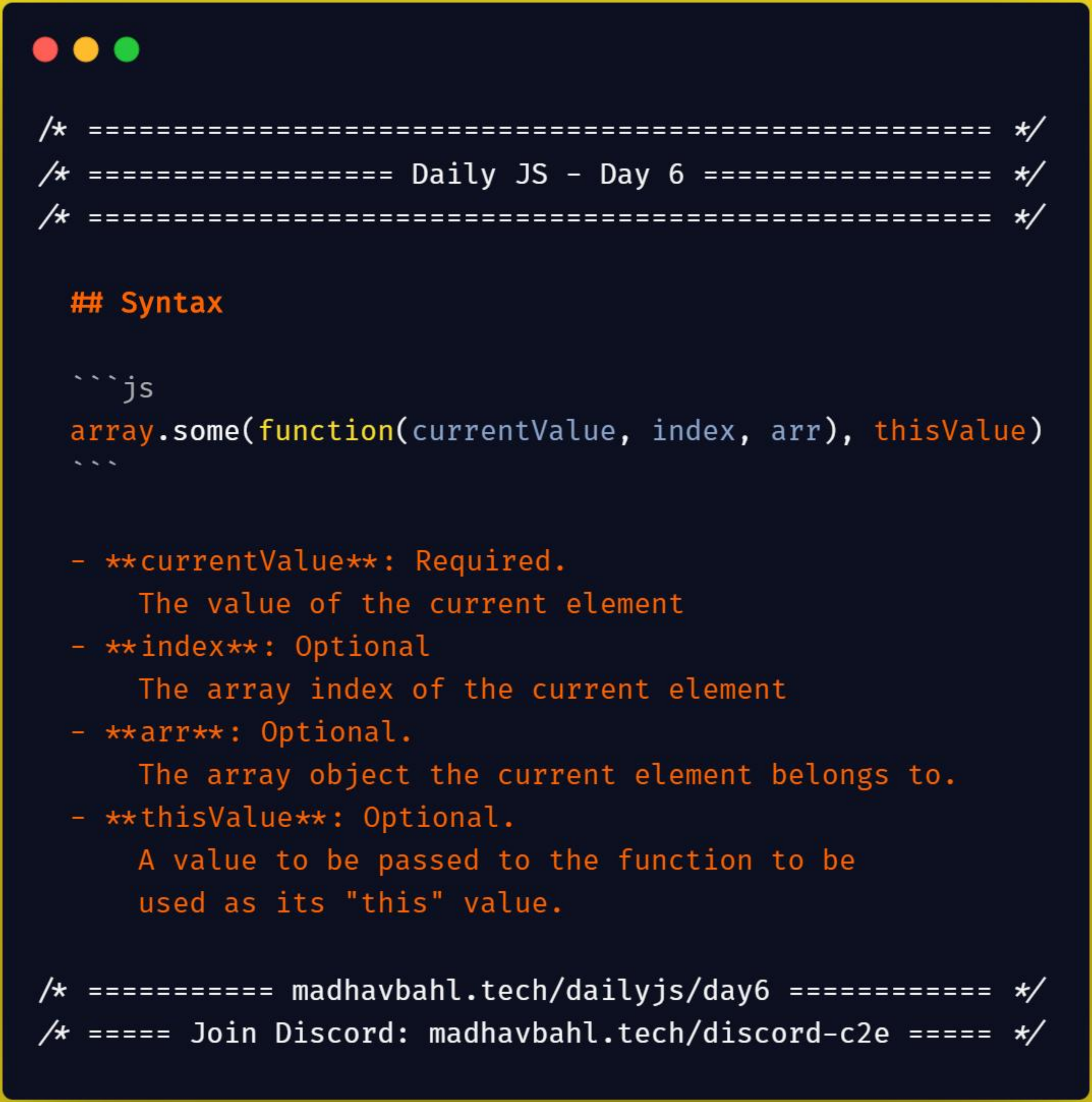
```
/* ============================================================================================ */
/* ==================================== Daily JS - Day 5 ==================================== */
/* ============================================================================================ */

/**
 * Slots of people are sent for driving license test,
 * check whether each person in a slot is 18 years or above
 */

const slot1 = [
    { name: 'John', age: 16 },
    { name: 'Matt', age: 20 },
    { name: 'Dan', age: 17 },
];
const slot2 =  [
    { name: 'Amanda', age: 32 },
    { name: 'Kepler', age: 20 },
    { name: 'Stan', age: 25 },
];

// Without arrow functions
function validateSlot (slot) {
    return slot.every (function (person) {
        return person.age >= 18;
    });
}
console.log ("Result 1 \n-----------------");
console.log ("Slot 1: ", validateSlot (slot1));
console.log ("Slot 2: ", validateSlot (slot2));
console.log ("-----------------");

// With arrow functions
const validateSlot2 = slot => slot.every (person => person.age>=18);
console.log ("Result 2 \n-----------------");
console.log ("Slot 1: ", validateSlot2 (slot1));
console.log ("Slot 2: ", validateSlot2 (slot2));

/* ============================== madhavbahl.tech/dailyjs/day5 ============================== */
/* ====================== Join Discord: madhavbahl.tech/discord-c2e ====================== */
```



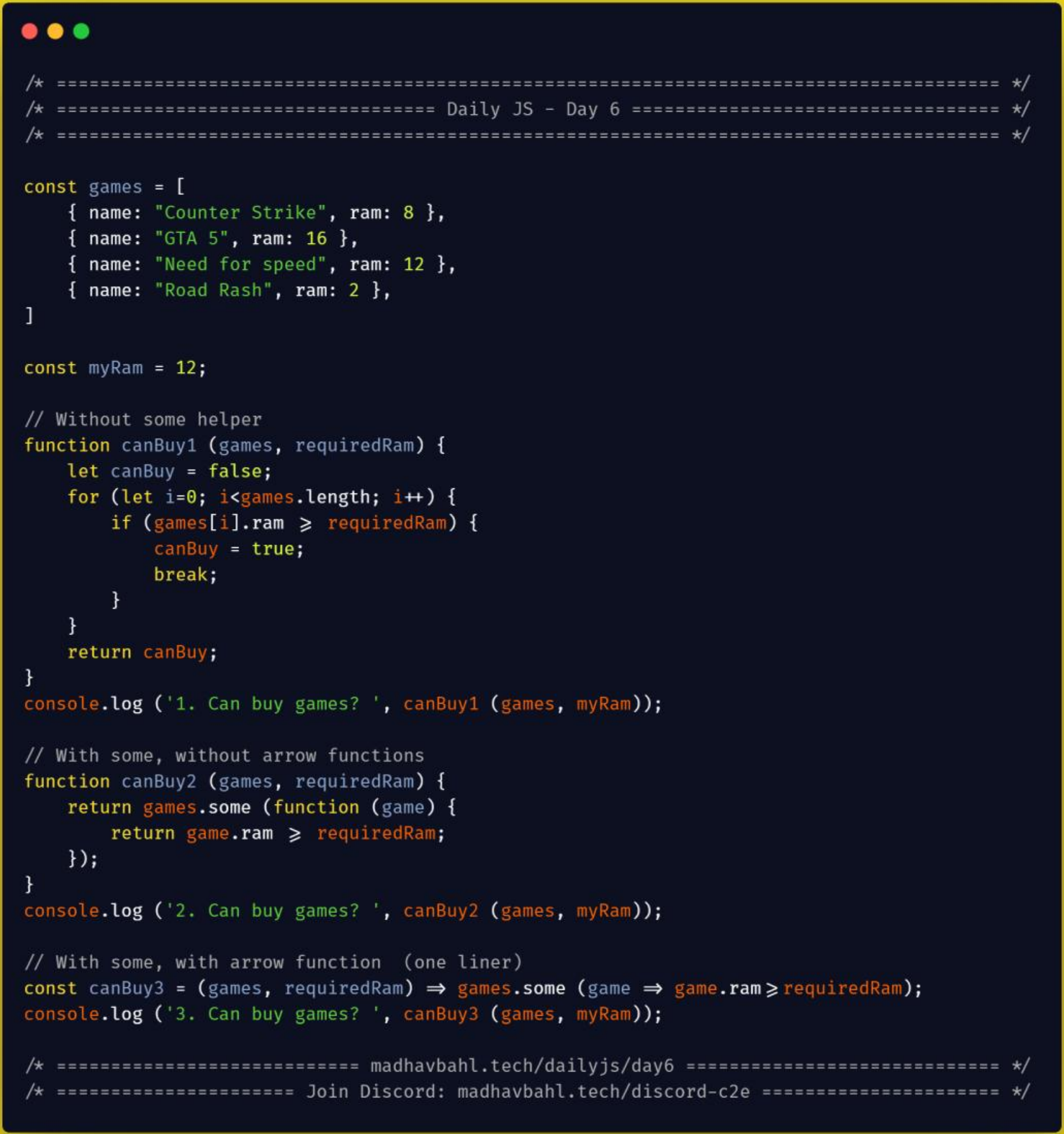
```
/* =============================================== */
/* =============== Daily JS - Day 5 ============== */
/* =============================================== */

  ## Homework for you

  Check whether a set of numbers is
     1. Even
     2. Odd
     3. Prime
     4. Negative
     5. Positive
     6. Perfect Square

  > Feel free to add more  :)

/* ======== madhavbahl.tech/dailyjs/day5 ========= */
/* == Join Discord: madhavbahl.tech/discord-c2e == */
```



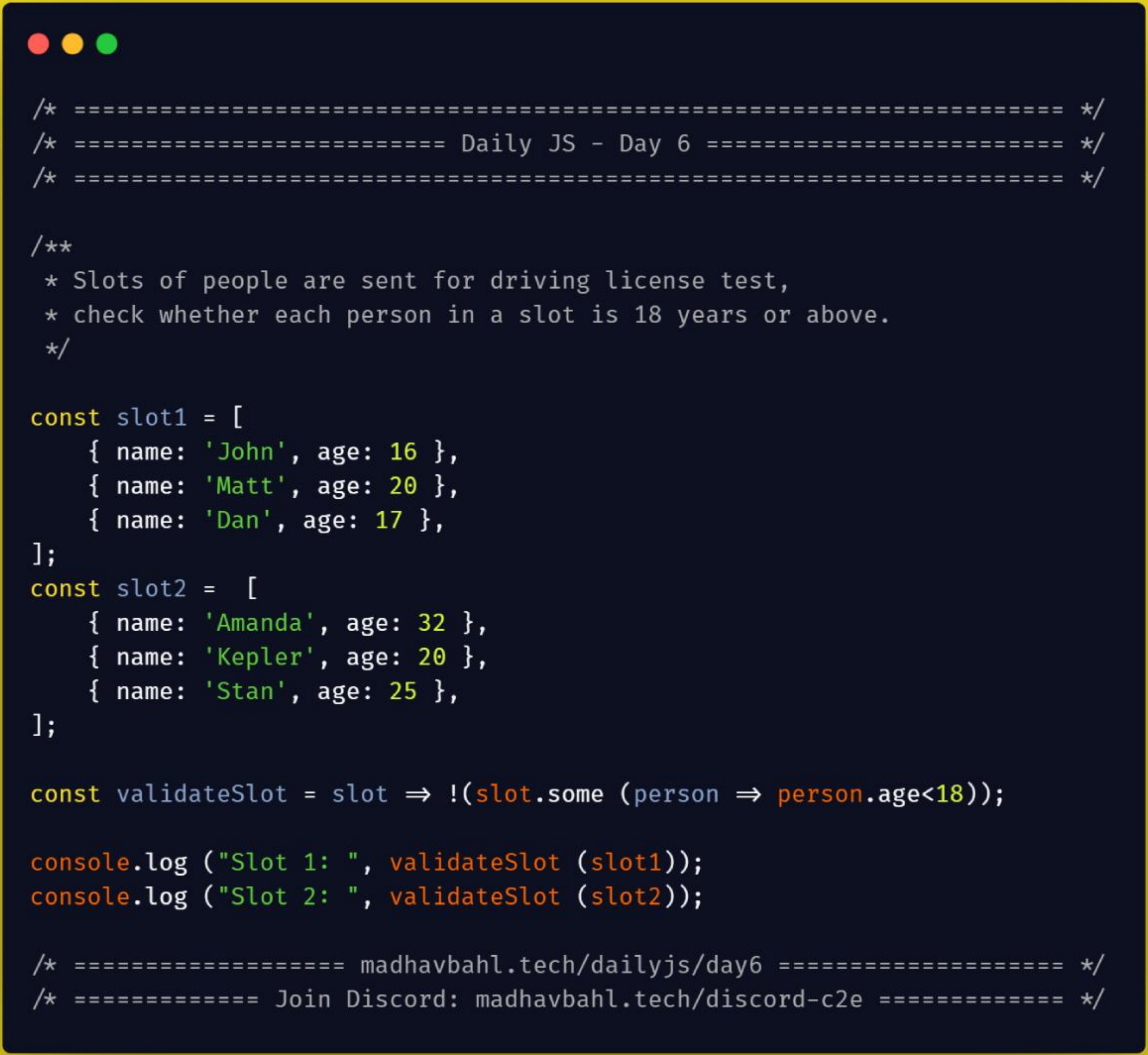
```
/* =================================================== */
/* ================= Daily JS - Day 6 ================ */
/* =================================================== */

  # Day 6: The `some` helper

  The `some` helper is used to find if any one or
  more elements in the given list, pass a specific
  condition.

  The condition is passed in the iterator function,
  and it executes that function once for each element
  in the list.

  If, for any element, the iterator function returns
  true, `some()` returns true.
  Otherwise, it returns false.

/* ========== madhavbahl.tech/dailyjs/day6 =========== */
/* ==== Join Discord: madhavbahl.tech/discord-c2e ==== */
```



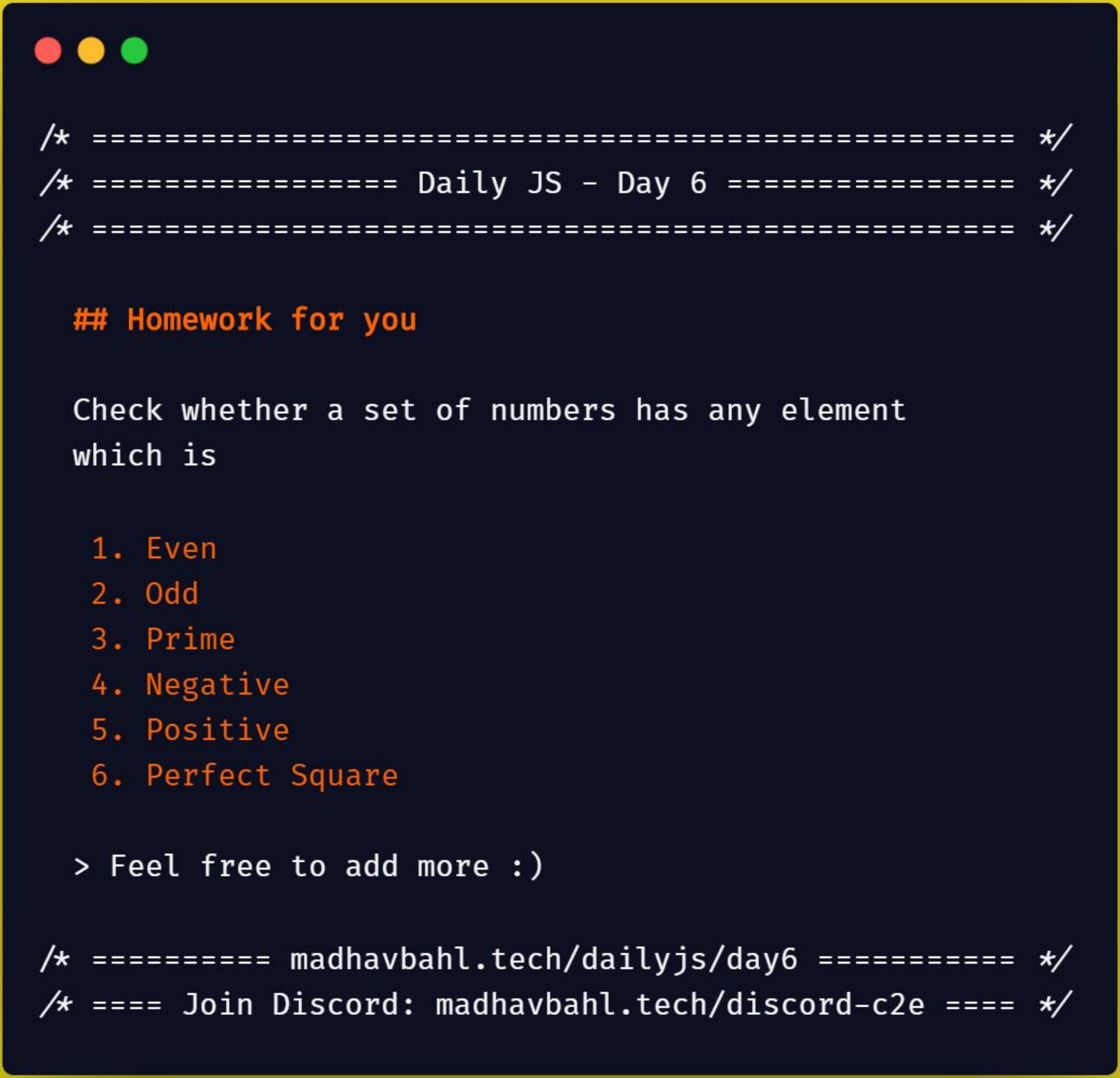
~~~
/* ===================================================== */
/* ================== Daily JS - Day 6 ================= */
/* ===================================================== */

  ## Syntax

  ```js
  array.some(function(currentValue, index, arr), thisValue)
  ```


  - **currentValue**: Required.
      The value of the current element
  - **index**: Optional
      The array index of the current element
  - **arr**: Optional.
      The array object the current element belongs to.
  - **thisValue**: Optional.
      A value to be passed to the function to be
      used as its "this" value.

/* =========== madhavbahl.tech/dailyjs/day6 ============ */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ===== */
~~~



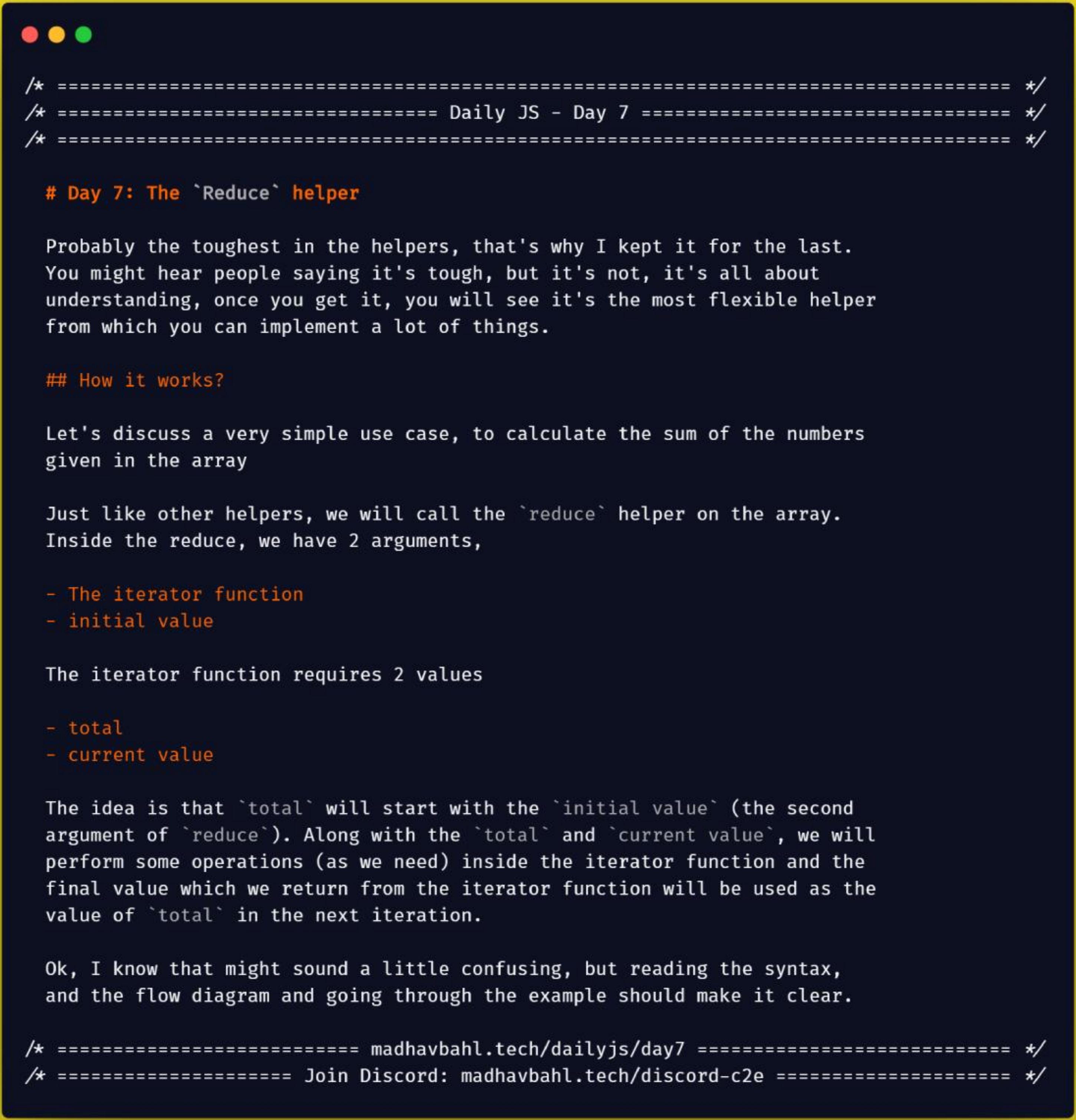
```
/* ======================================================================================== */
/* =================================== Daily JS - Day 6 ================================== */
/* ======================================================================================== */

const games = [
    { name: "Counter Strike", ram: 8 },
    { name: "GTA 5", ram: 16 },
    { name: "Need for speed", ram: 12 },
    { name: "Road Rash", ram: 2 },
]

const myRam = 12;

// Without some helper
function canBuy1 (games, requiredRam) {
    let canBuy = false;
    for (let i=0; i<games.length; i++) {
        if (games[i].ram >= requiredRam) {
            canBuy = true;
            break;
        }
    }
    return canBuy;
}
console.log ('1. Can buy games? ', canBuy1 (games, myRam));

// With some, without arrow functions
function canBuy2 (games, requiredRam) {
    return games.some (function (game) {
        return game.ram >= requiredRam;
    });
}
console.log ('2. Can buy games? ', canBuy2 (games, myRam));

// With some, with arrow function  (one liner)
const canBuy3 = (games, requiredRam) => games.some (game => game.ram>=requiredRam);
console.log ('3. Can buy games? ', canBuy3 (games, myRam));

/* ============================ madhavbahl.tech/dailyjs/day6 ============================= */
/* ====================== Join Discord: madhavbahl.tech/discord-c2e ====================== */
```



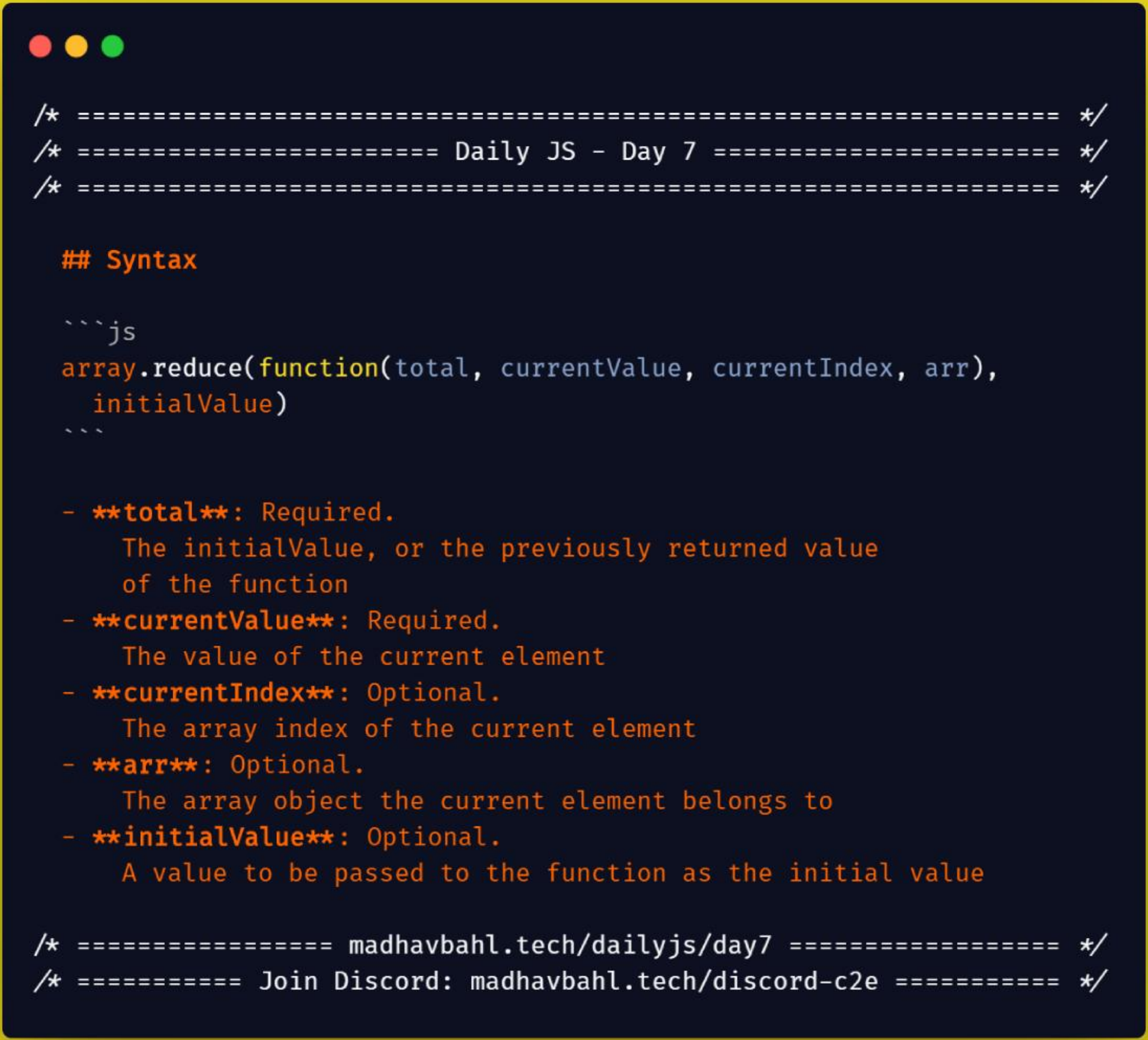
```javascript
/* ===================================================================== */
/* ========================== Daily JS - Day 6 ========================= */
/* ===================================================================== */

/**
 * Slots of people are sent for driving license test,
 * check whether each person in a slot is 18 years or above.
 */

const slot1 = [
    { name: 'John', age: 16 },
    { name: 'Matt', age: 20 },
    { name: 'Dan', age: 17 },
];
const slot2 =  [
    { name: 'Amanda', age: 32 },
    { name: 'Kepler', age: 20 },
    { name: 'Stan', age: 25 },
];

const validateSlot = slot => !(slot.some (person => person.age<18));

console.log ("Slot 1: ", validateSlot (slot1));
console.log ("Slot 2: ", validateSlot (slot2));

/* =================== madhavbahl.tech/dailyjs/day6 ==================== */
/* ============= Join Discord: madhavbahl.tech/discord-c2e ============= */
```



```
/* =================================================== */
/* ================= Daily JS - Day 6 ================ */
/* =================================================== */

  ## Homework for you

  Check whether a set of numbers has any element
  which is

   1. Even
   2. Odd
   3. Prime
   4. Negative
   5. Positive
   6. Perfect Square

  > Feel free to add more :)

/* ========== madhavbahl.tech/dailyjs/day6 =========== */
/* ==== Join Discord: madhavbahl.tech/discord-c2e ==== */
```

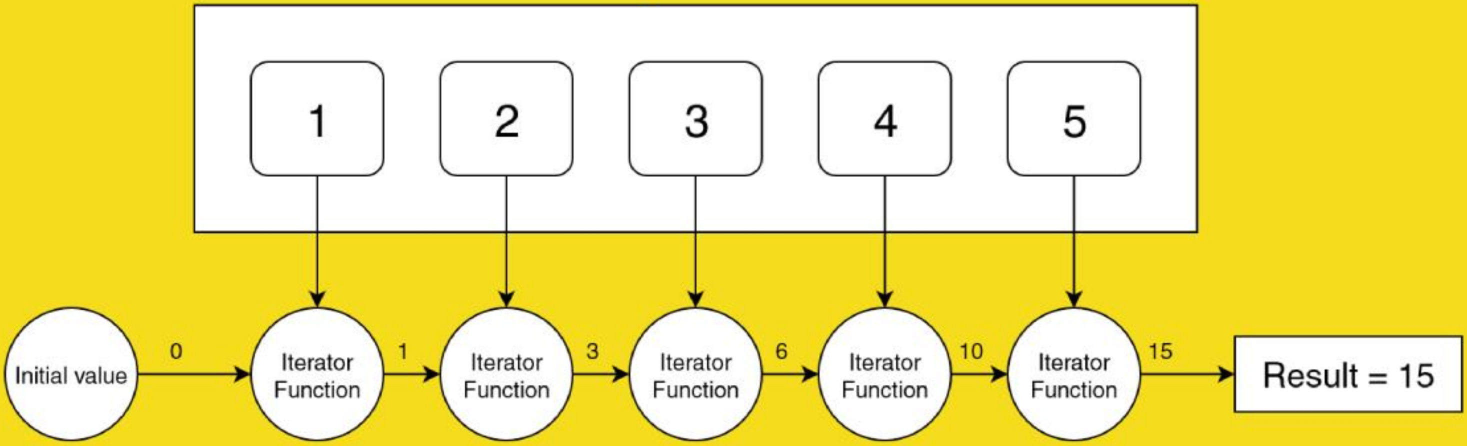


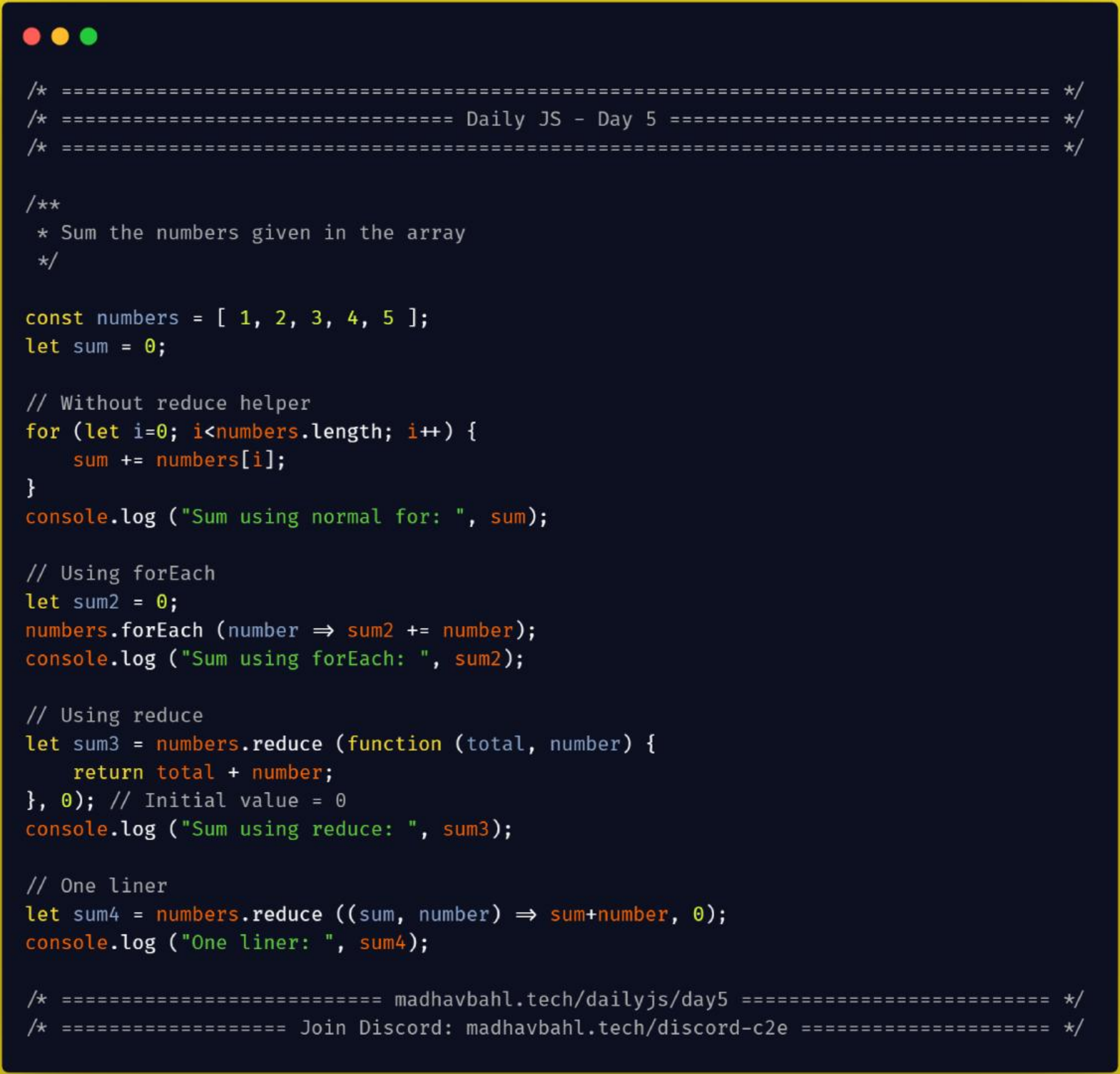
```
/* ======================================================================================== */
/* ================================== Daily JS - Day 7 ================================== */
/* ======================================================================================== */

  # Day 7: The `Reduce` helper

  Probably the toughest in the helpers, that's why I kept it for the last.
  You might hear people saying it's tough, but it's not, it's all about
  understanding, once you get it, you will see it's the most flexible helper
  from which you can implement a lot of things.

  ## How it works?

  Let's discuss a very simple use case, to calculate the sum of the numbers
  given in the array

  Just like other helpers, we will call the `reduce` helper on the array.
  Inside the reduce, we have 2 arguments,

  - The iterator function
  - initial value

  The iterator function requires 2 values

  - total
  - current value

  The idea is that `total` will start with the `initial value` (the second
  argument of `reduce`). Along with the `total` and `current value`, we will
  perform some operations (as we need) inside the iterator function and the
  final value which we return from the iterator function will be used as the
  value of `total` in the next iteration.

  Ok, I know that might sound a little confusing, but reading the syntax,
  and the flow diagram and going through the example should make it clear.

/* =========================== madhavbahl.tech/dailyjs/day7 ============================= */
/* ===================== Join Discord: madhavbahl.tech/discord-c2e ====================== */
```



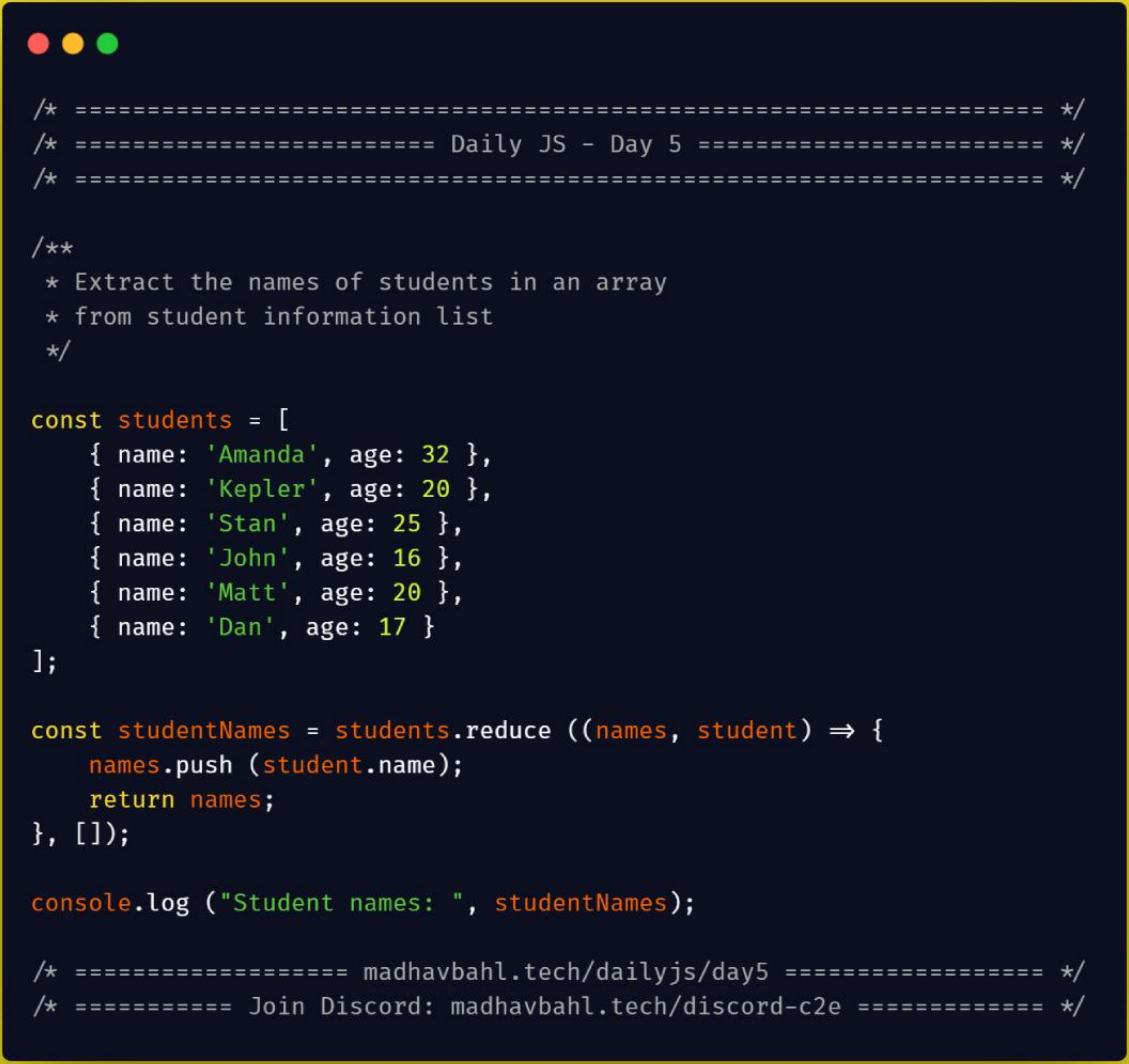
~~~
/* ================================================================= */
/* ======================== Daily JS - Day 7 ======================= */
/* ================================================================= */

  ## Syntax

  ```js
  array.reduce(function(total, currentValue, currentIndex, arr),
    initialValue)
  ```


  - **total**: Required.
      The initialValue, or the previously returned value
      of the function
  - **currentValue**: Required.
      The value of the current element
  - **currentIndex**: Optional.
      The array index of the current element
  - **arr**: Optional.
      The array object the current element belongs to
  - **initialValue**: Optional.
      A value to be passed to the function as the initial value

/* ================= madhavbahl.tech/dailyjs/day7 ================== */
/* =========== Join Discord: madhavbahl.tech/discord-c2e =========== */
~~~

The reduce helper

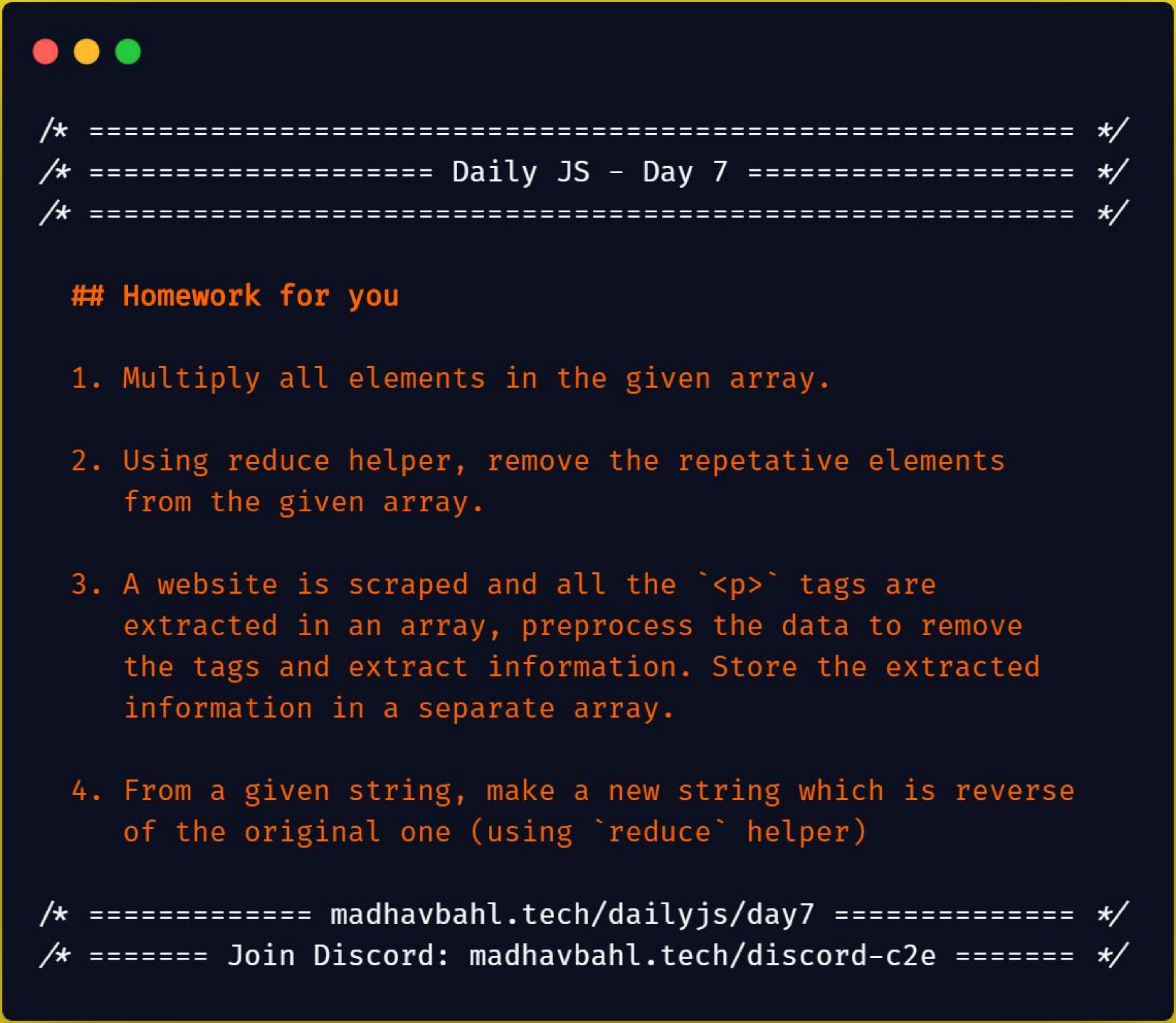




```
/* =================================================================================== */
/* ================================= Daily JS - Day 5 ================================ */
/* =================================================================================== */

/**
 * Sum the numbers given in the array
 */

const numbers = [ 1, 2, 3, 4, 5 ];
let sum = 0;

// Without reduce helper
for (let i=0; i<numbers.length; i++) {
    sum += numbers[i];
}
console.log ("Sum using normal for: ", sum);

// Using forEach
let sum2 = 0;
numbers.forEach (number => sum2 += number);
console.log ("Sum using forEach: ", sum2);

// Using reduce
let sum3 = numbers.reduce (function (total, number) {
    return total + number;
}, 0); // Initial value = 0
console.log ("Sum using reduce: ", sum3);

// One liner
let sum4 = numbers.reduce ((sum, number) => sum+number, 0);
console.log ("One liner: ", sum4);

/* =========================== madhavbahl.tech/dailyjs/day5 ========================== */
/* =================== Join Discord: madhavbahl.tech/discord-c2e ===================== */
```



```javascript
/* =================================================================== */
/* ========================= Daily JS - Day 5 ======================== */
/* =================================================================== */

/**
 * Extract the names of students in an array
 * from student information list
 */

const students = [
    { name: 'Amanda', age: 32 },
    { name: 'Kepler', age: 20 },
    { name: 'Stan', age: 25 },
    { name: 'John', age: 16 },
    { name: 'Matt', age: 20 },
    { name: 'Dan', age: 17 }
];

const studentNames = students.reduce ((names, student) => {
    names.push (student.name);
    return names;
}, []);

console.log ("Student names: ", studentNames);

/* =================== madhavbahl.tech/dailyjs/day5 ================== */
/* =========== Join Discord: madhavbahl.tech/discord-c2e ============= */
```



```
/* ========================================================= */
/* ==================== Daily JS - Day 7 =================== */
/* ========================================================= */

  ## Homework for you

  1. Multiply all elements in the given array.

  2. Using reduce helper, remove the repetative elements
     from the given array.

  3. A website is scraped and all the `<p>` tags are
     extracted in an array, preprocess the data to remove
     the tags and extract information. Store the extracted
     information in a separate array.

  4. From a given string, make a new string which is reverse
     of the original one (using `reduce` helper)

/* ============= madhavbahl.tech/dailyjs/day7 ============== */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ======= */
```

THE LEAN PROGRAMMER | CodeTo Express

Thank you!

Feel free to reach out...

Email: theleanprogrammer@gmail.com
Web: madhavbahl.tech/
Github: github.com/MadhavBahlMD
LinkedIn: linkedin.com/in/madhavbahl/
Insta: instagram.com/theleanprogrammer/

DAILY JS