WEEK 1: Modern JavaScript

A. OBJECTIVE

1. Students are able to know and implement the use of a variable using const
2. Students are able to know and implement the use of a variable using let
3. Students are able to know and implement the use of template strings
4. Students are able to know and implement the use of arrow functions
5. Students are able to know and implement the use of a destructuring object
6. Students are able to know and implement the use of a destructuring array
7. Students are able to know and implement the use of spread and rest operators
8. Students are able to know and implement the use of constructor and super classes

B. MATERIAL

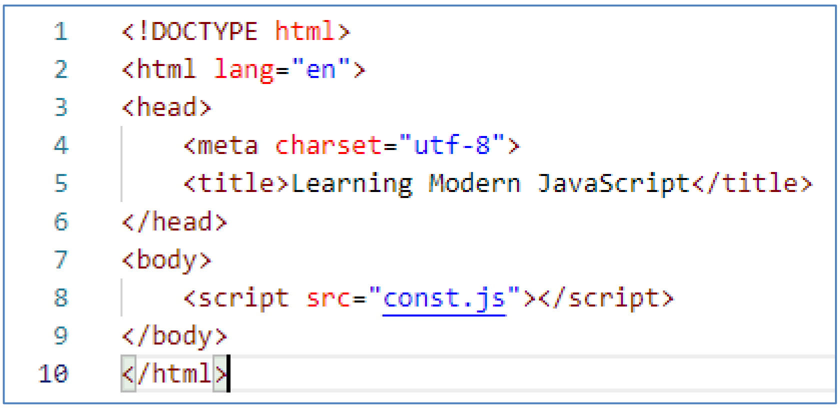
1. Reference
   https://reactjs.org/tutorial/tutorial.html
2. Modern JavaScript
   - JavaScript is based on the EcmaScript specification, which is a standard set by the European Association of Computer Manufacturers (ECMA).
   - We'll start by learning about JavaScript with the new features added by ES6. We will write ES6 with React.
   - Understanding ES6 is essential to understanding React

C. EXPERIMENT

a. Creating a variable using const

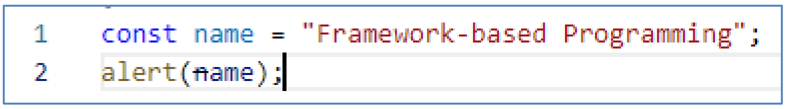
1. Create a .html file page to display the results, and a .js file to write the JavaScript code.

const.html



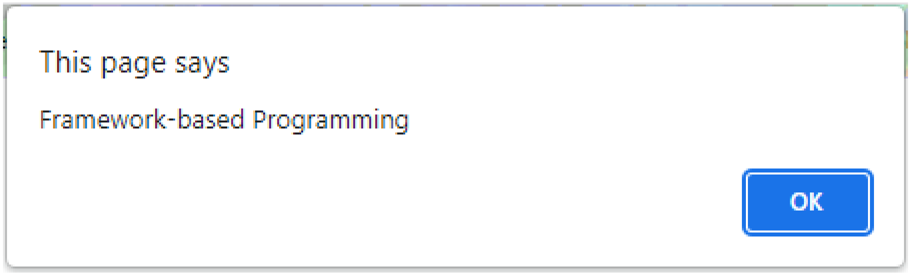
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="const.js"></script>
</body>
</html>
```

const.js



```javascript
const name = "Framework-based Programming";
alert(name);
```

2. Open the const.html file in your browser



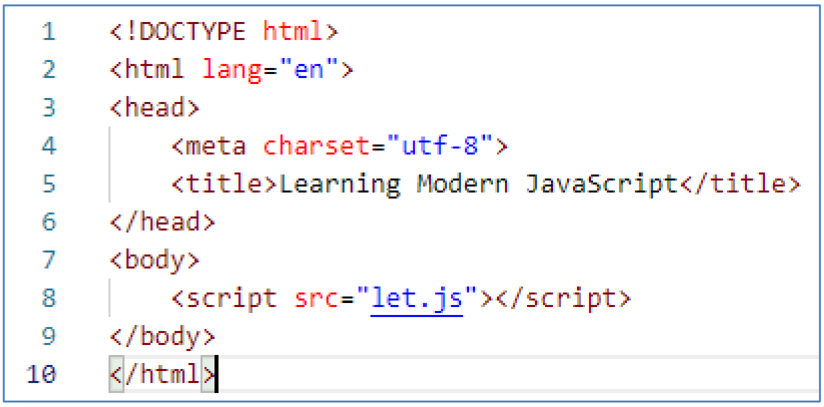
This page says

Framework-based Programming

b. Creating a variable using let

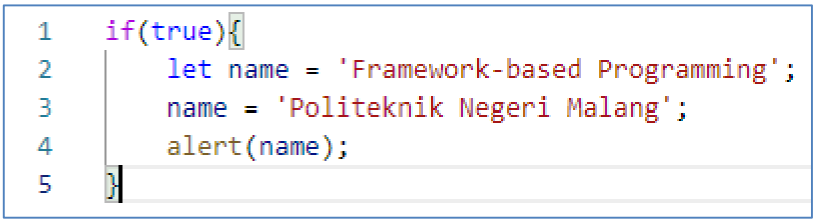
1. Create a .html file page to display the results, and a .js file to write the JavaScript code.

let.html



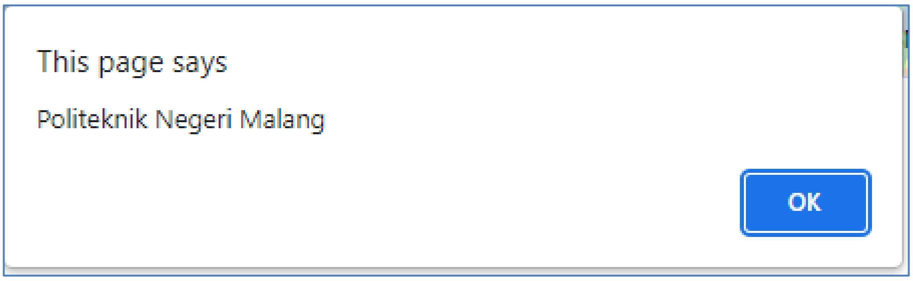
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="let.js"></script>
</body>
</html>
```

let.js



```javascript
if(true){
    let name = 'Framework-based Programming';
    name = 'Politeknik Negeri Malang';
    alert(name);
}
```

2. Open the let.html file in your browser



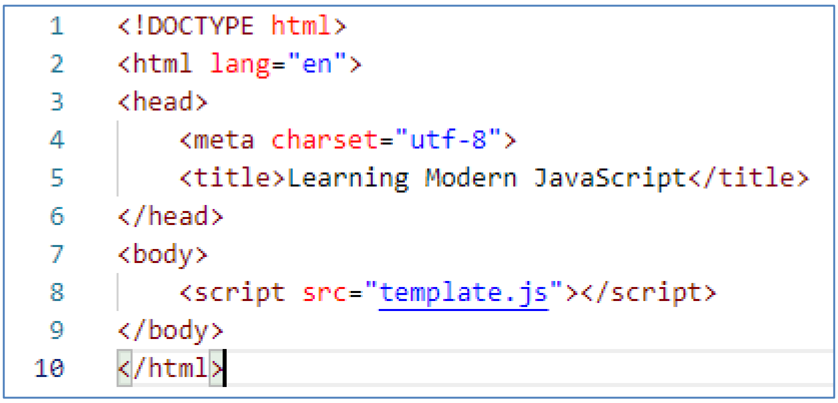
This page says

Politeknik Negeri Malang

c. Creating Template Strings

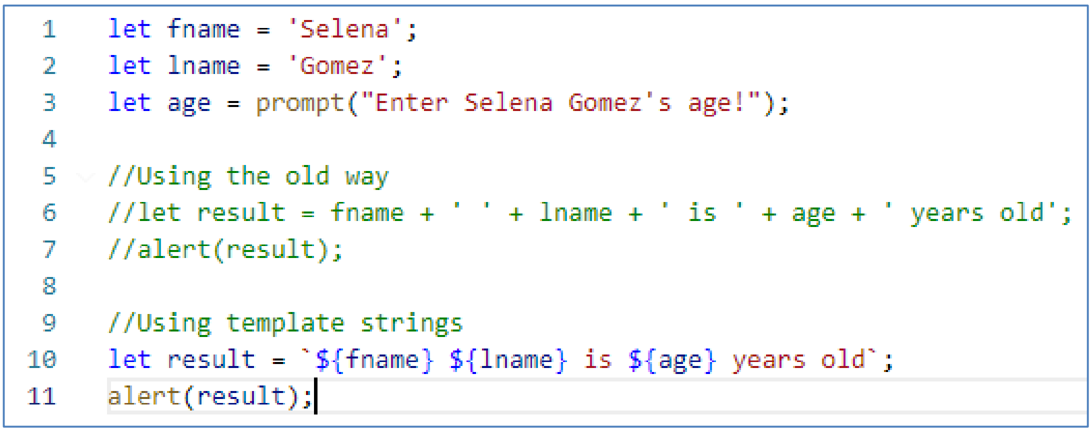
1. Create a .html file page to display the results, and a .js file to write the JavaScript code.

template.html



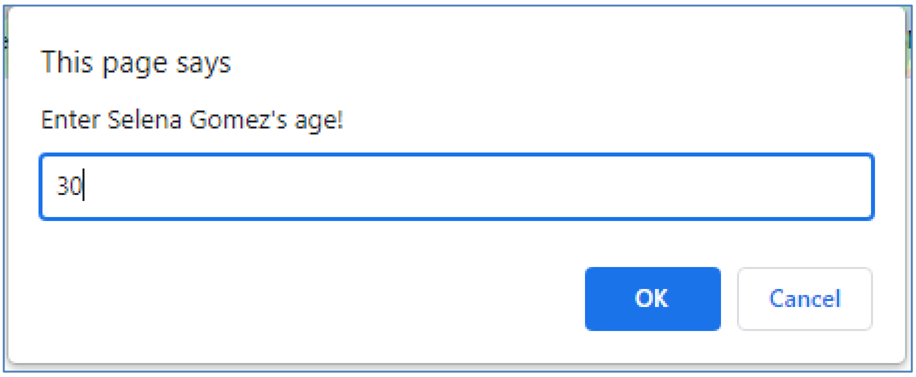
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="template.js"></script>
</body>
</html>
```

template.js



```javascript
let fname = 'Selena';
let lname = 'Gomez';
let age = prompt("Enter Selena Gomez's age!");

//Using the old way
//let result = fname + ' ' + lname + ' is ' + age + ' years old';
//alert(result);

//Using template strings
let result = `${fname} ${lname} is ${age} years old`;
alert(result);
```

2. Open the template.html file in your browser

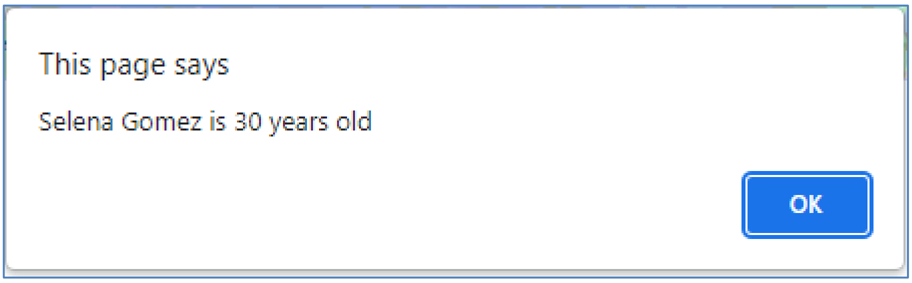


This page says

Enter Selena Gomez's age!

30

Fill in the data in the field box, then click the OK button.



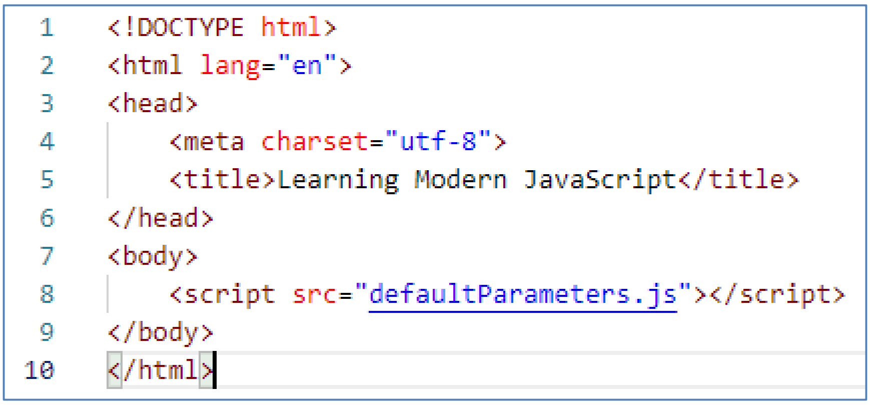
This page says

Selena Gomez is 30 years old

d. Creating default parameters

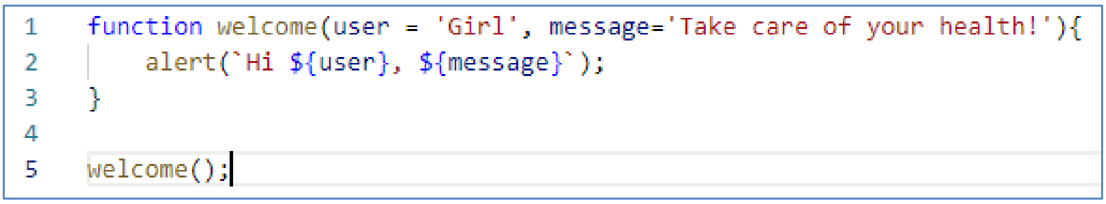
1. Create a .html file page to display the results, and a .js file to write the JavaScript code.

defaultParameters.html



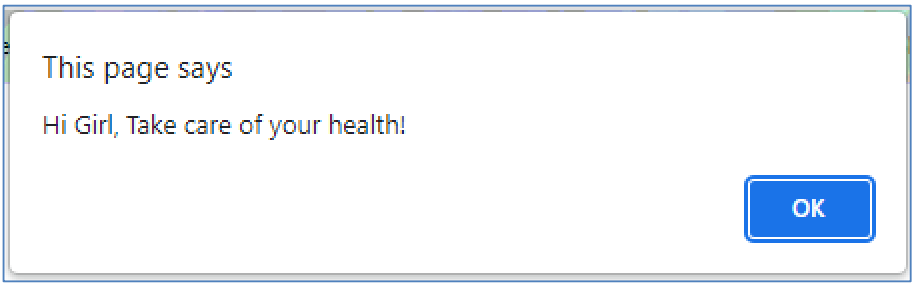
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="defaultParameters.js"></script>
</body>
</html>
```

defaultParameters.js



```javascript
function welcome(user = 'Girl', message='Take care of your health!'){
    alert(`Hi ${user}, ${message}`);
}

welcome();
```

2. Open the defaultParameters.html file in your browser



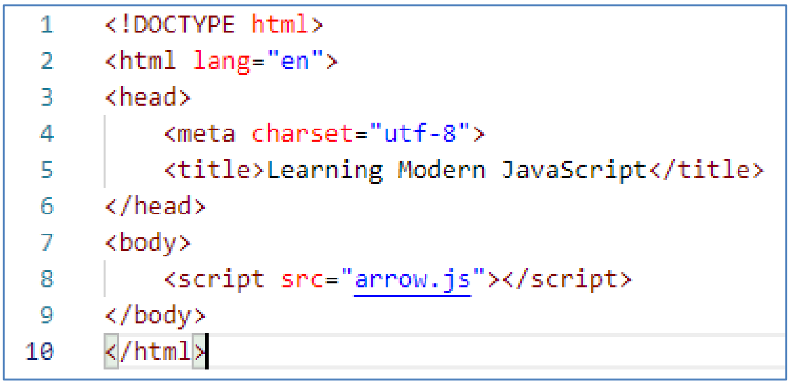
This page says

Hi Girl, Take care of your health!

e. Creating Arrow Function 1

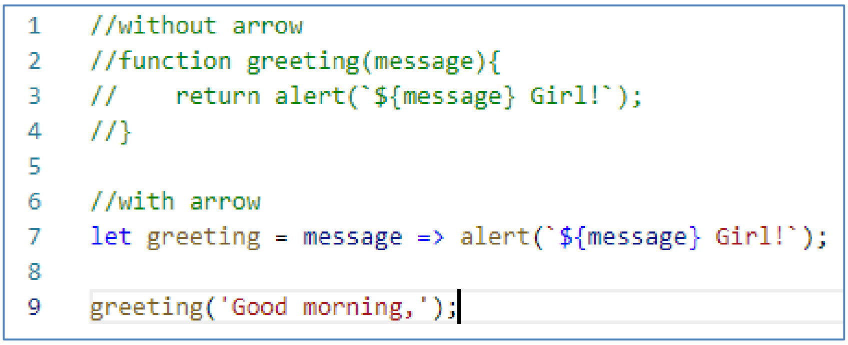
1. Create a .html file page to display the results, and a .js file to write the JavaScript code.

arrow.html



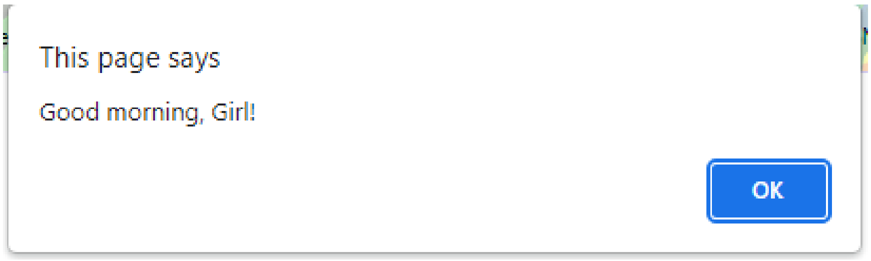
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="arrow.js"></script>
</body>
</html>
```

arrow.js



```
//without arrow
//function greeting(message){
//    return alert(`${message} Girl!`);
//}

//with arrow
let greeting = message => alert(`${message} Girl!`);

greeting('Good morning,');
```

2. Open the arrow.html file in your browser



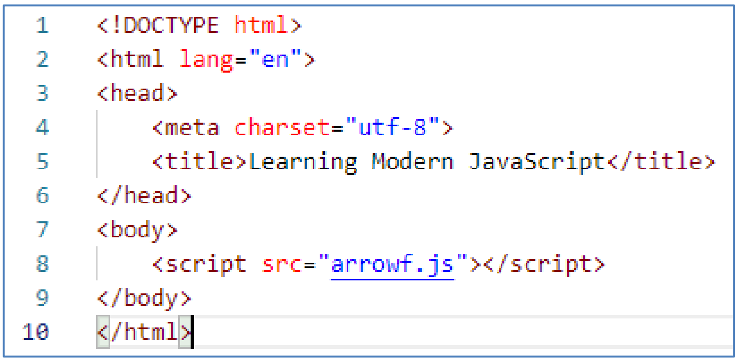
This page says

Good morning, Girl!

f. Creating Arrow Function 2

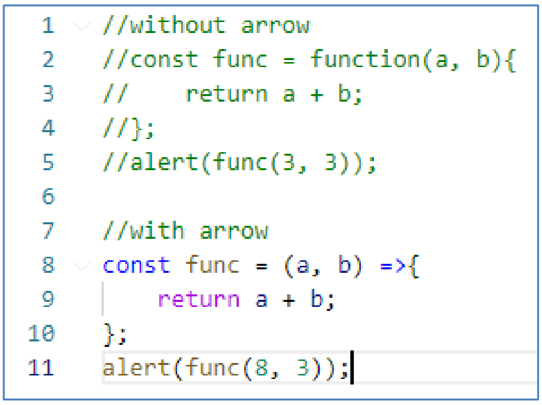
1. Create a .html file page to display the results, and a .js file to write the JavaScript code.

arrowf.html



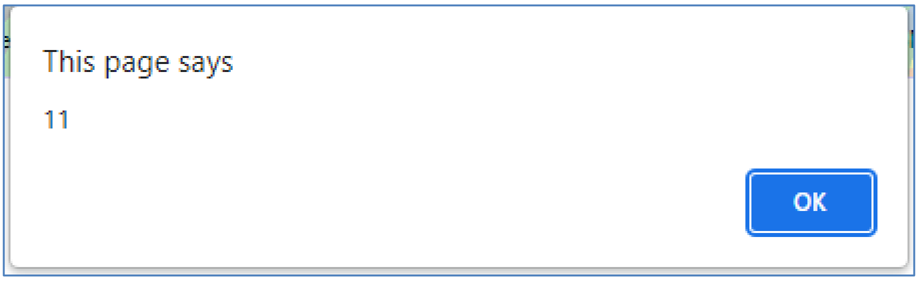
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="arrowf.js"></script>
</body>
</html>
```

arrowf.js



```
//without arrow
//const func = function(a, b){
//    return a + b;
//};
//alert(func(3, 3));

//with arrow
const func = (a, b) =>{
    return a + b;
};
alert(func(8, 3));
```

2. Open the arrowf.html file in your browser



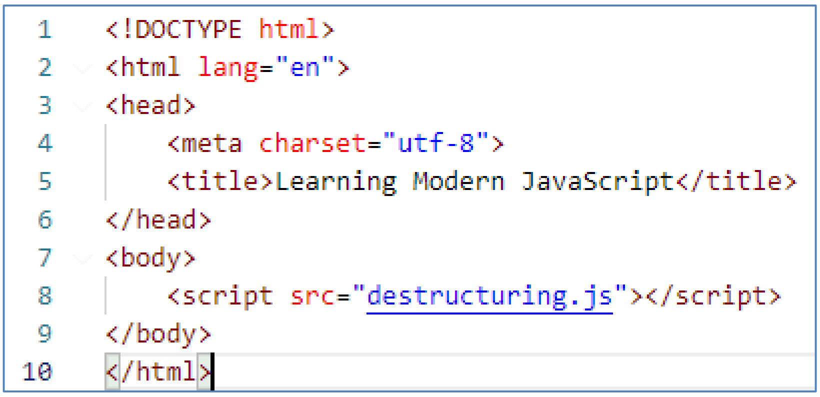
This page says

11

g. Creating a Destructuring Object

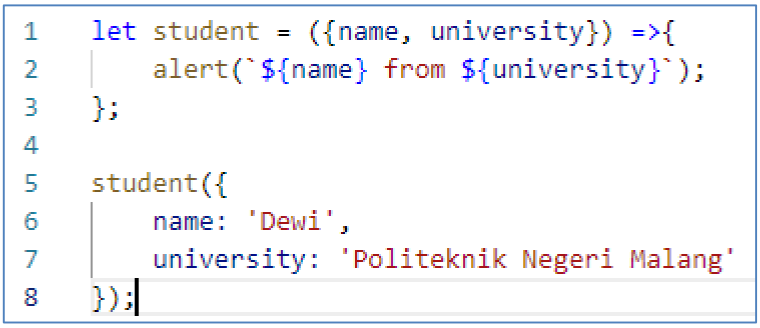
1. Create a .html file page to display the results, and a .js file to write the JavaScript code.

destructuring.html



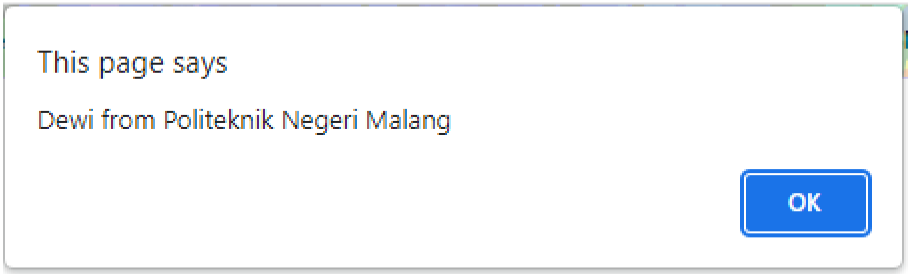
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="destructuring.js"></script>
</body>
</html>
```

destructuring.js



```
let student = ({name, university}) =>{
    alert(`${name} from ${university}`);
};

student({
    name: 'Dewi',
    university: 'Politeknik Negeri Malang'
});
```

2. Open the destructuring.html file in your browser



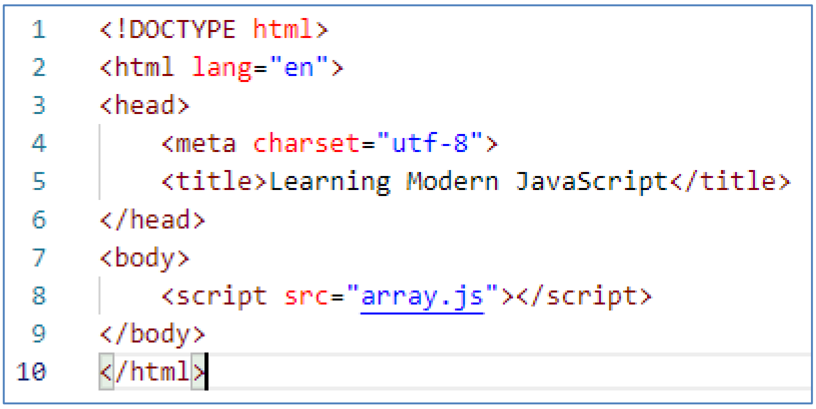
This page says

Dewi from Politeknik Negeri Malang

h. Creating a Destructuring Array

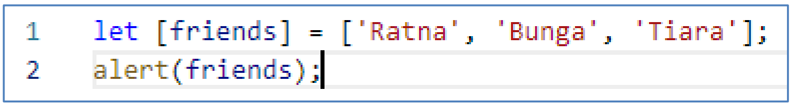
1. Create a .html file page to display the results, and a .js file to write the JavaScript code.

array.html



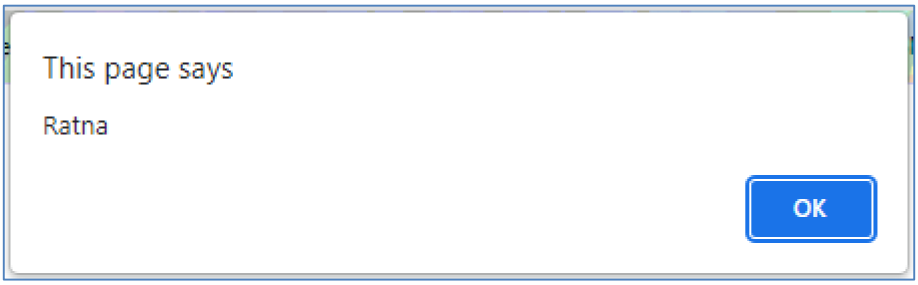
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="array.js"></script>
</body>
</html>
```

array.js



```javascript
let [friends] = ['Ratna', 'Bunga', 'Tiara'];
alert(friends);
```

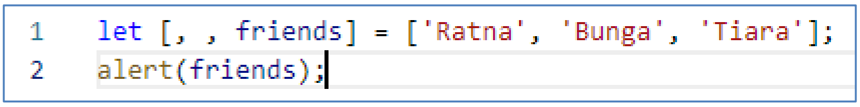
2. Open the array.html file in your browser



This page says

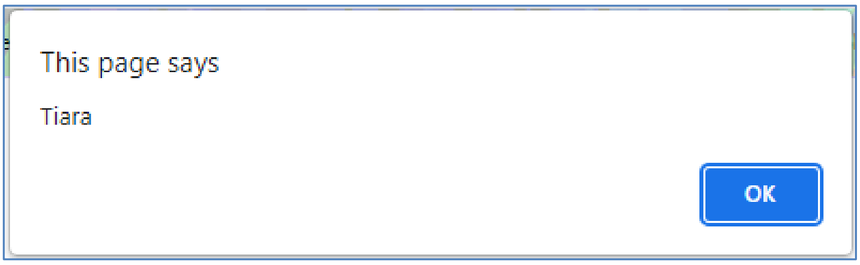
Ratna

3. Modify the code in array.js



```javascript
let [, , friends] = ['Ratna', 'Bunga', 'Tiara'];
alert(friends);
```

4. Reopen array.html in the browser, the results will be different



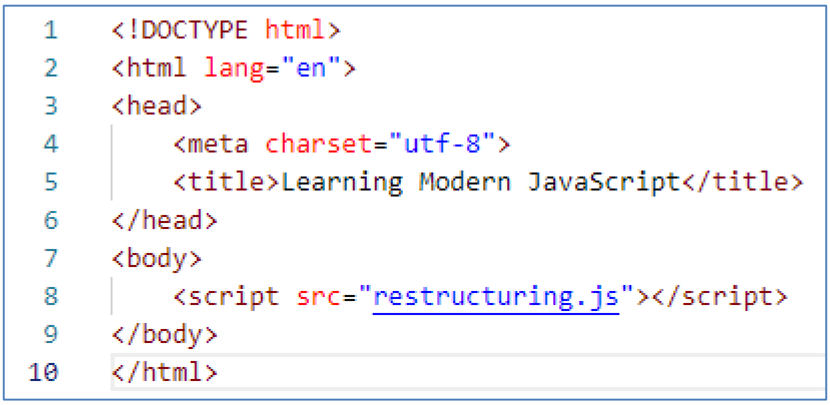
This page says

Tiara

i. Creating a Restructuring

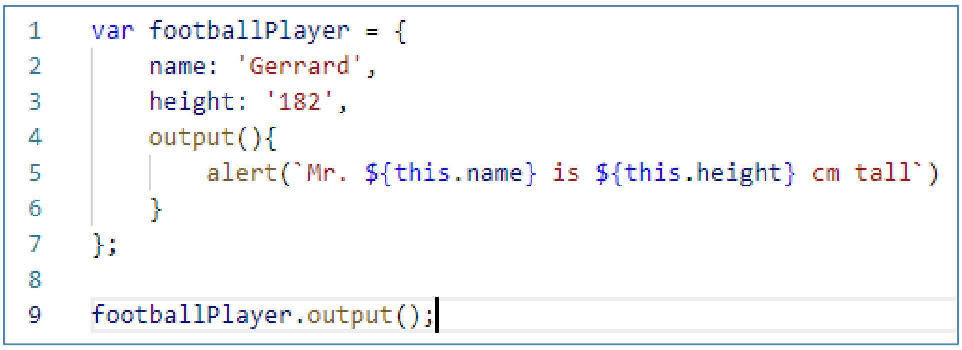
1. Create a .html file page to display the results, and a .js file to write the JavaScript code.

restructuring.html



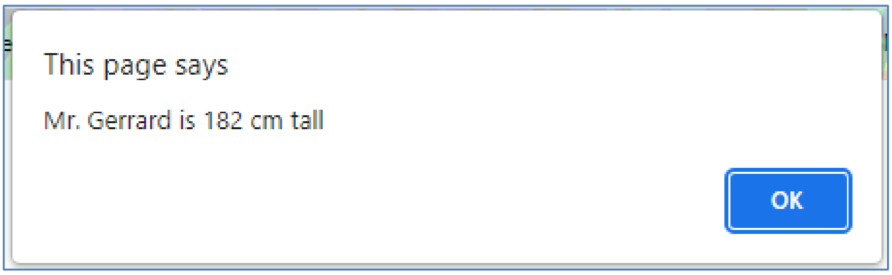
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="restructuring.js"></script>
</body>
</html>
```

restructuring.js



```javascript
var footballPlayer = {
    name: 'Gerrard',
    height: '182',
    output(){
        alert(`Mr. ${this.name} is ${this.height} cm tall`)
    }
};

footballPlayer.output();
```

2. Open the restructuring.html file in your browser



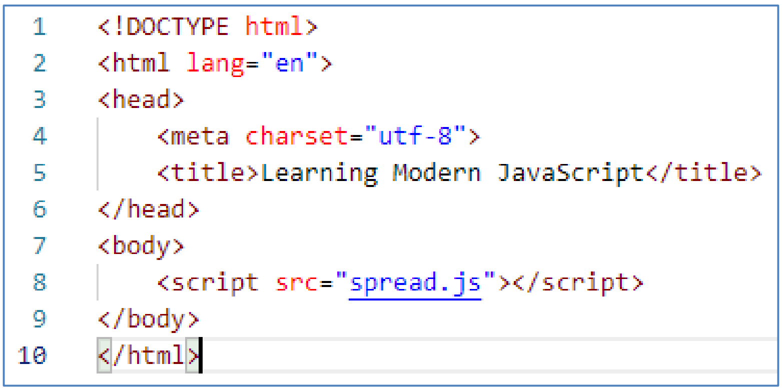
This page says

Mr. Gerrard is 182 cm tall

j. Creating Spread and Rest Operators

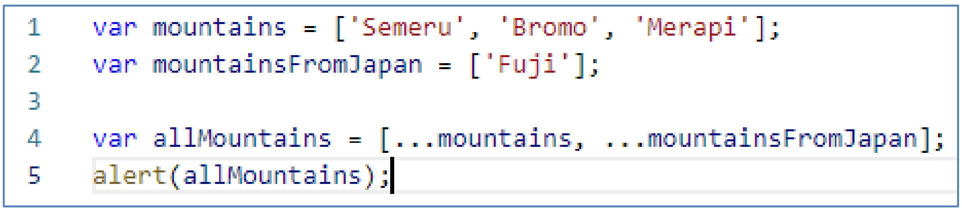
1. Create a .html file page to display the results, and a .js file to write the JavaScript code.

spread.html



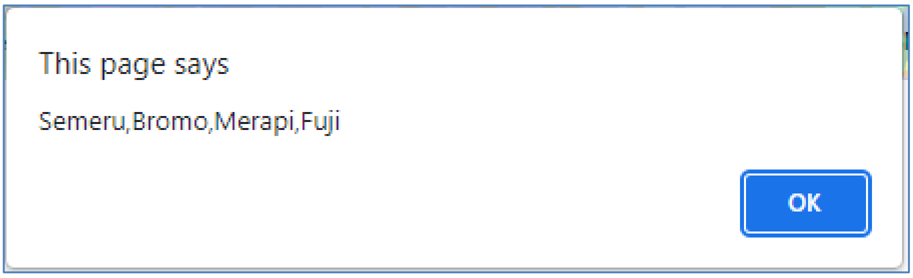
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="spread.js"></script>
</body>
</html>
```

spread.js



```
var mountains = ['Semeru', 'Bromo', 'Merapi'];
var mountainsFromJapan = ['Fuji'];

var allMountains = [...mountains, ...mountainsFromJapan];
alert(allMountains);
```

2. Open the spread.html file in your browser

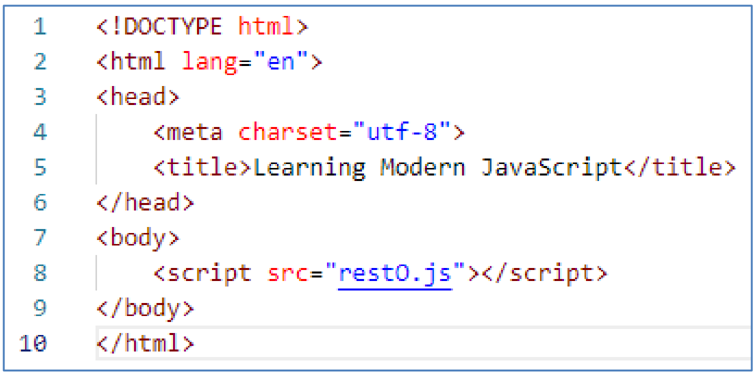


This page says

Semeru,Bromo,Merapi,Fuji

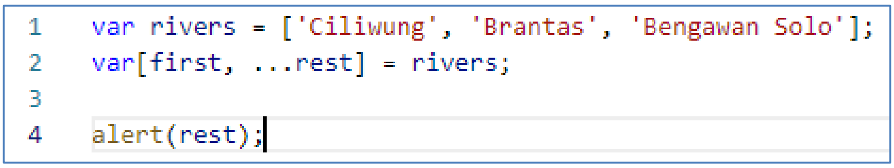
3. To do a practical about rest operation, create two new .html and .js files

restO.html



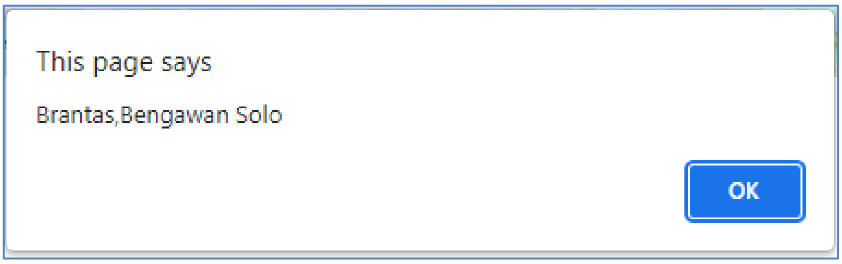
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="restO.js"></script>
</body>
</html>
```

restO.js



```
var rivers = ['Ciliwung', 'Brantas', 'Bengawan Solo'];
var[first, ...rest] = rivers;

alert(rest);
```

4. Open the restO.html file in your browser



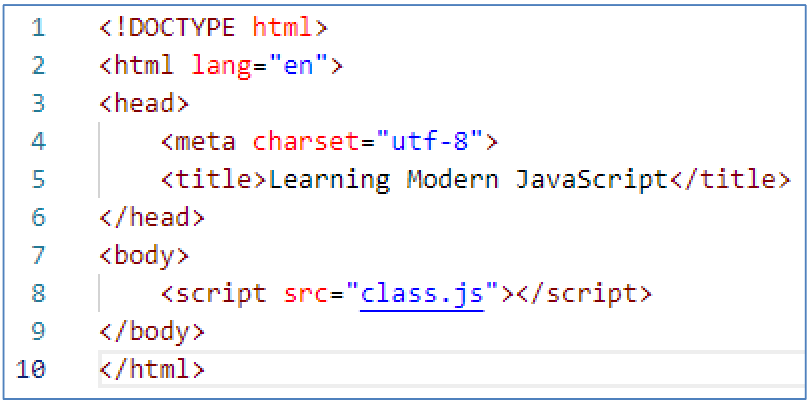
This page says

Brantas,Bengawan Solo

k. Creating Constructor and Super Classes

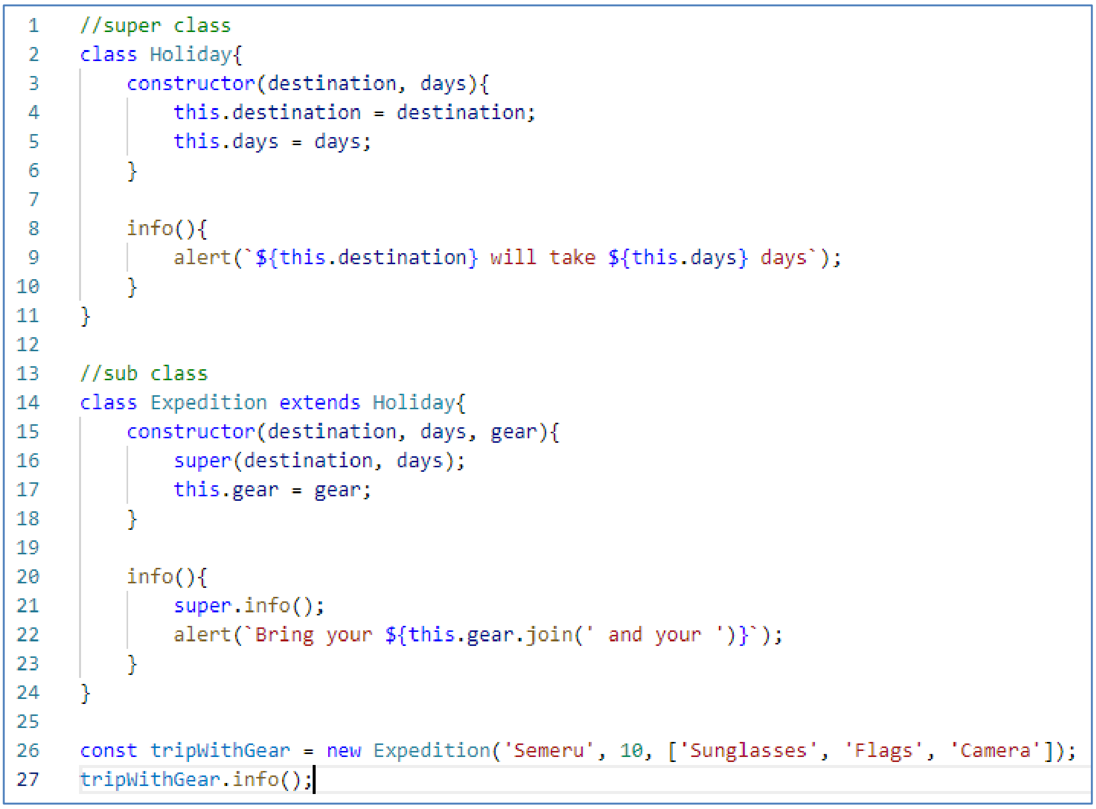
1. Create a .html file page to display the results, and a .js file to write the JavaScript code.

class.html



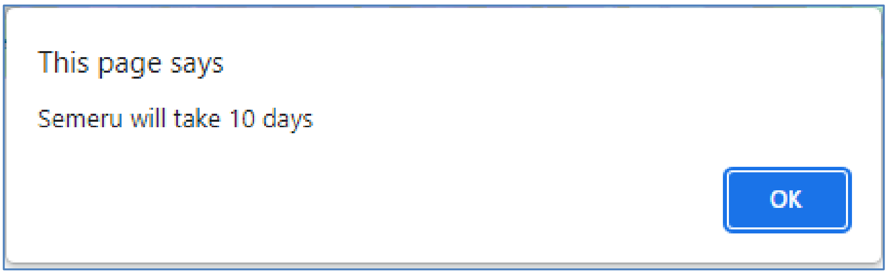
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Learning Modern JavaScript</title>
</head>
<body>
    <script src="class.js"></script>
</body>
</html>
```

class.js



```javascript
//super class
class Holiday{
    constructor(destination, days){
        this.destination = destination;
        this.days = days;
    }

    info(){
        alert(`${this.destination} will take ${this.days} days`);
    }
}

//sub class
class Expedition extends Holiday{
    constructor(destination, days, gear){
        super(destination, days);
        this.gear = gear;
    }

    info(){
        super.info();
        alert(`Bring your ${this.gear.join(' and your ')}`);
    }
}

const tripWithGear = new Expedition('Semeru', 10, ['Sunglasses', 'Flags', 'Camera']);
tripWithGear.info();
```

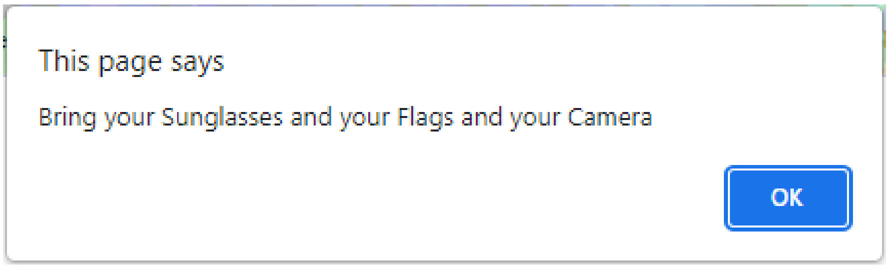
2. Open the class.html file in your browser



This page says

Semeru will take 10 days

3. Click the OK button, the next page will appear



This page says

Bring your Sunglasses and your Flags and your Camera

D. ASSIGNMENT

1. Make a practicum report containing screenshots of the results for each experiment. Provide an explanation of the results of the screenshots for each experiment based on your understanding supported from various reference sources.