

TUGAS PRAKTIKUM

Mata Kuliah : Pemrograman Berbasis Framework

MODUL 1: Modern JavaScript

Oleh:

Rizki Saputra Wibisono NIM. 1741720142

TI 3D / 26



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

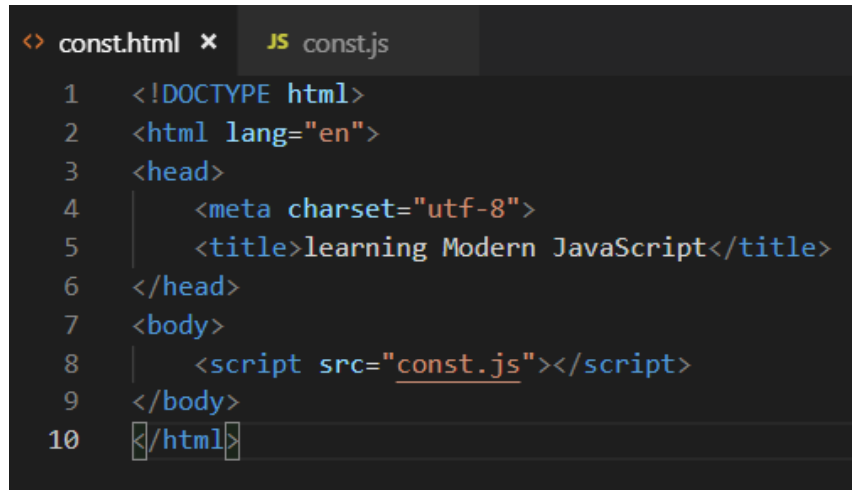
POLITEKNIK NEGERI MALANG

FEBRUARI 2020

a. Membuat variabel menggunakan const

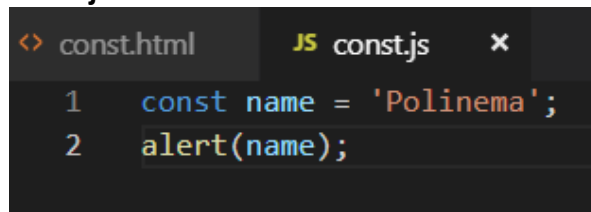
1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

const.html

A screenshot of a code editor showing the HTML file 'const.html'. The code is as follows:

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

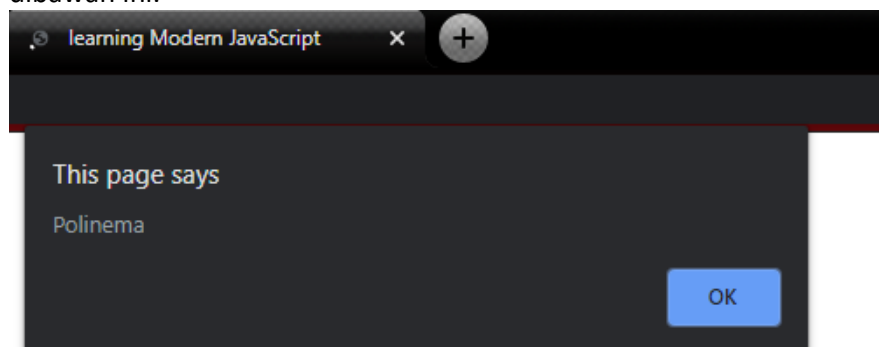
const.js

A screenshot of a code editor showing the JavaScript file 'const.js'. The code is as follows:

```
const name = 'Polinema';
alert(name);
```

Penjelasan : const digunakan untuk mendeklarasikan variable readonly, tapi bukan berarti variable tersebut tetap, hanya saja variable tersebut tidak dapat di deklarasi ulang.

2. Selanjutnya buka file const.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



b. Membuat variabel menggunakan let

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

let.html

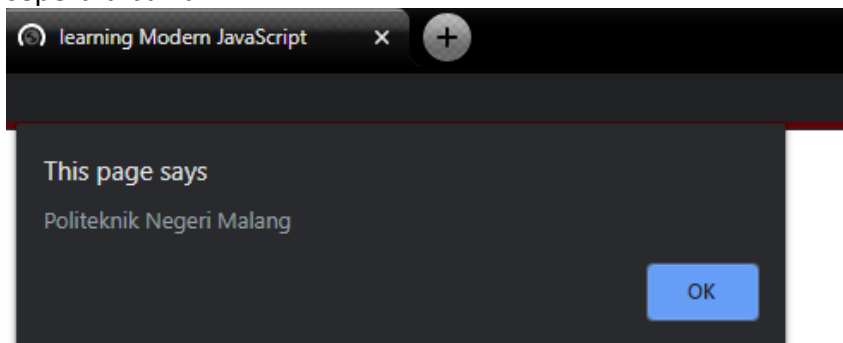
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>learning Modern JavaScript</title>
6 </head>
7 <body>
8   <script src="let.js"></script>
9 </body>
10 </html>
```

Let.js

```
1 if(true){
2   let name = 'Polinema';
3   name = 'Politeknik Negeri Malang';
4   alert(name);
5 }
```

Penjelasan : let adalah variable local yang cakupannya lebih kecil dari var

2. Selanjutnya buka file let.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



c. Membuat Template Strings

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

template.html

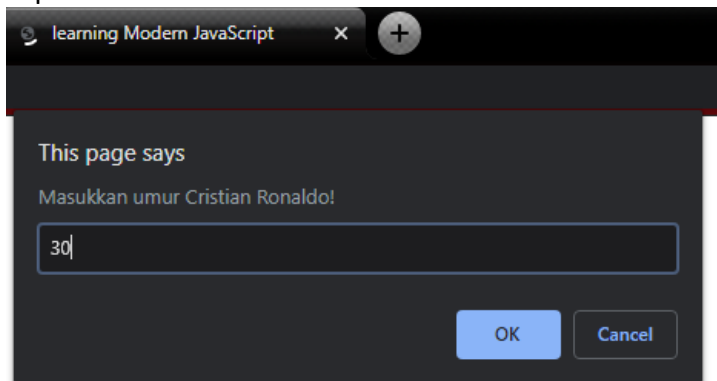
```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="utf-8">
5  |   <title>learning Modern JavaScript</title>
6  </head>
7  <body>
8  |   <script src="template.js"></script>
9  </body>
10 </html>
```

template.js

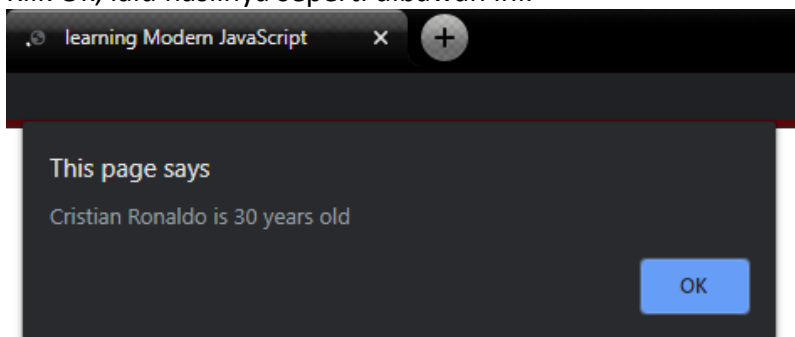
```
1  let fname = 'Cristian';
2  let lname = 'Ronaldo';
3  let age = prompt("Masukkan umur Cristian Ronaldo!");
4
5  // Cara lama
6  // let result = fname + ' ' + lname + 'is' + age + 'years old';
7  // alert(result);
8
9  // Memakai template strings
10 let result = `${fname} ${lname} is ${age} years old`
11 alert(result);
```

Penjelasan : membuat tampilan popup yang berisikan inputan yang nantinya akan di tampilkan kembali

2. Selanjutnya buka file template.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



Klik Ok, lalu hasilnya seperti dibawah ini.



d. Membuat default parameters

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

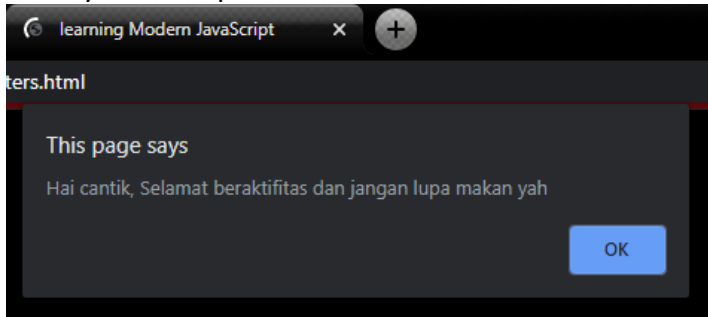
defaultParameters.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="utf-8">
5  |   <title>learning Modern JavaScript</title>
6  </head>
7  <body>
8  |   <script src="defaultParameters.js"></script>
9  </body>
10 </html>
```

defaultParameters.js

```
1 function welcome(user = 'cantik', message = 'Selamat beraktifitas dan jangan lupa makan yah'){
2     alert(`Hai ${user}, ${message}`);
3 }
4
5 welcome();
```

2. Selanjutnya buka file defaultParameters.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



e. Membuat Arrow Function 1

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

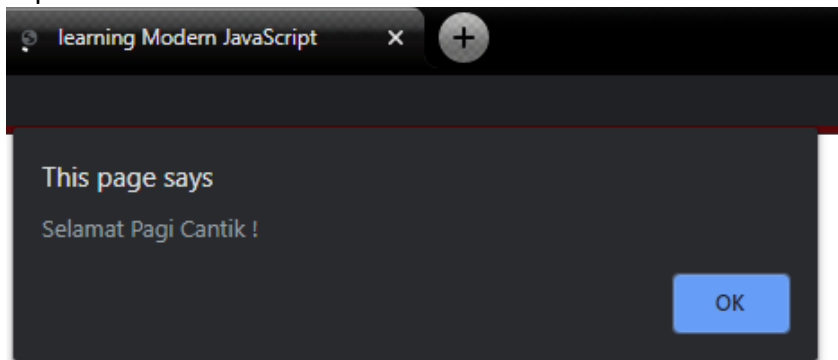
arrow.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="utf-8">
5     <title>learning Modern JavaScript</title>
6 </head>
7 <body>
8     <script src="arrow.js"></script>
9 </body>
10 </html>
```

arrow.js

```
1 // Tanpa arrow
2 // function gretting(message) {
3 //     return alert(`${message} Cantik !`);
4 // }
5
6 // Menggunakan arrow
7 let gretting = message => alert(`${message} Cantik !`);
8
9 gretting('Selamat Pagi');
```

2. Selanjutnya buka file arrow.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



f. Membuat Arrow Function 2

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

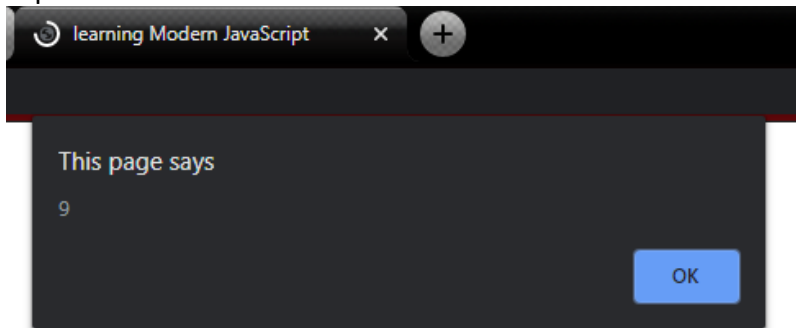
arrowf.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>learning Modern JavaScript</title>
6  </head>
7  <body>
8    <script src="arrowf.js"></script>
9  </body>
10 </html>
```

arrowf.js

```
1  // Tanpa arrow
2  // const function = function(a,b) {
3  //   return a + b;
4  // }
5  // alert(func(3,3));
6
7  // Menggunakan arrow
8  const func = (a,b) => {
9    return a + b;
10 };
11 alert(func(5,4));
```

2. Selanjutnya buka file arrowf.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



g. Membuat Destructuring Object

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

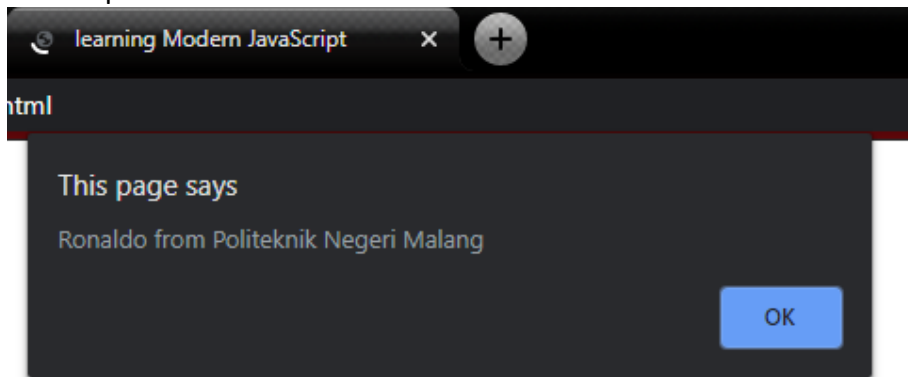
destructuring.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>learning Modern JavaScript</title>
6  </head>
7  <body>
8    <script src="destructuring.js"></script>
9  </body>
10 </html>
```

destructuring.js

```
1  let polStudent = ({name, polytechnic}) => {
2    alert(`${name} from ${polytechnic}`);
3  };
4
5  polStudent({
6    name: 'Ronaldo',
7    polytechnic: 'Politeknik Negeri Malang'
8  });
```


2. Selanjutnya buka file destructuring.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



h. Membuat Destructuring an Array

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

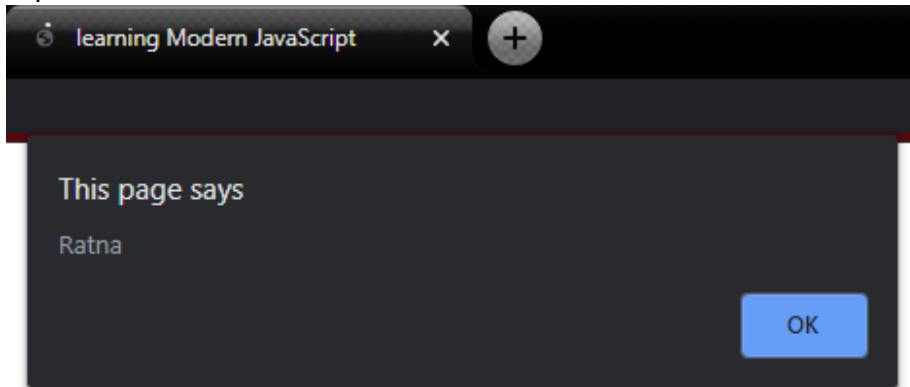
array.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="utf-8">
5  |   <title>learning Modern JavaScript</title>
6  </head>
7  <body>
8  |   <script src="array.js"></script>
9  </body>
10 </html>
```

array.js

```
1  let [wife] = ['Ratna', 'Bunga', 'Tiara'];
2  alert(wife);
```

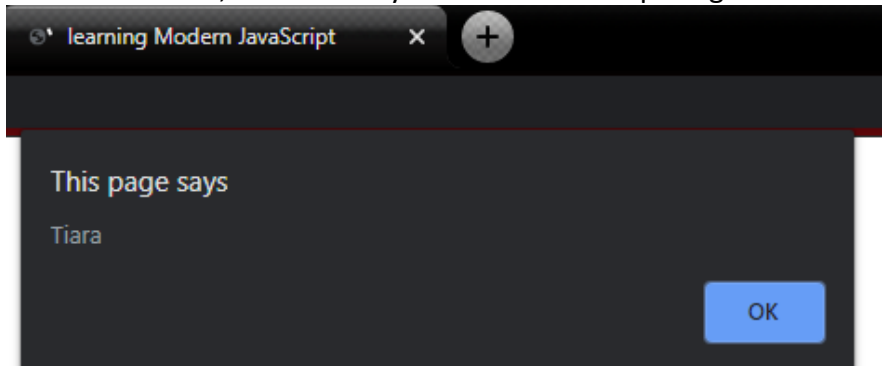
2. Selanjutnya buka file array.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



3. Kemudian ganti sedikit code pada array.js seperti dibawah ini

```
1 let [, , wife] = ['Ratna', 'Bunga', 'Tiara'];  
2 alert(wife);
```

4. Bukalah kembali, maka hasilnya akan berbeda seperti gambar



i. Membuat Restructuring

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

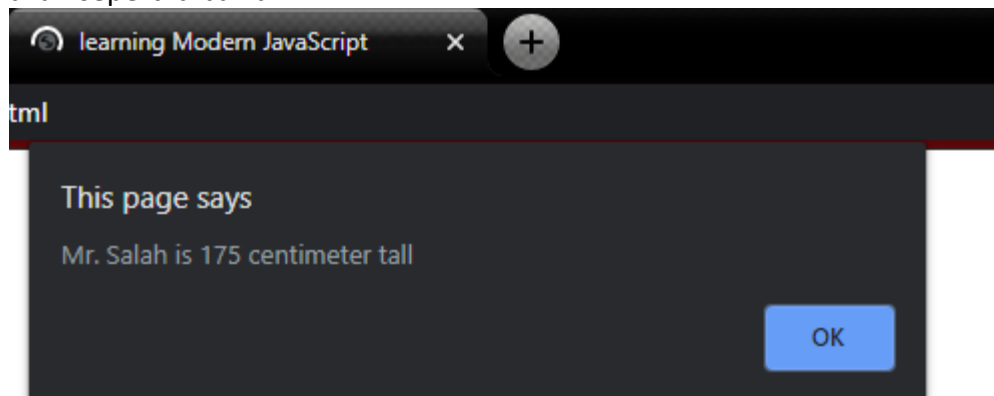
restructuring.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="utf-8">
5  |   <title>learning Modern JavaScript</title>
6  </head>
7  <body>
8  |   <script src="restructuring.js"></script>
9  </body>
10 </html>
```

restructuring.js

```
1  var pemainSepakbola = {
2  |   name: 'Salah',
3  |   height: '175',
4  |   output(){
5  |       alert(`Mr. ${this.name} is ${this.height} centimeter tall`);
6  |   }
7  };
8  pemainSepakbola.output();
```

2. Selanjutnya buka file restructuring.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



j. Membuat Spread and Rest operator

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

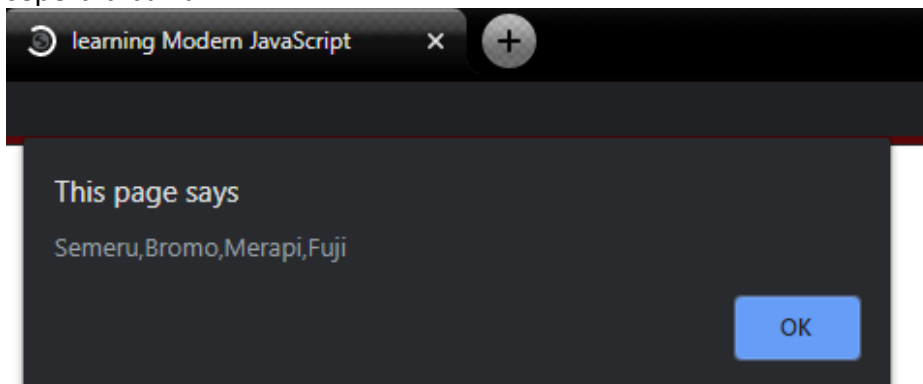
spread.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>learning Modern JavaScript</title>
6 </head>
7 <body>
8   <script src="spread.js"></script>
9 </body>
10 </html>
```

spread.js

```
1 var mountains = ['Semeru', 'Bromo', 'Merapi'];
2 var mountainsFromJapan = ['Fuji'];
3
4 var allMountains = [...mountains, ...mountainsFromJapan];
5 alert(allMountains);
```

2. Selanjutnya buka file spread.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



3. Untuk melakukan praktek rest operation, buatlah dua file .html dan .js seperti dibawah ini

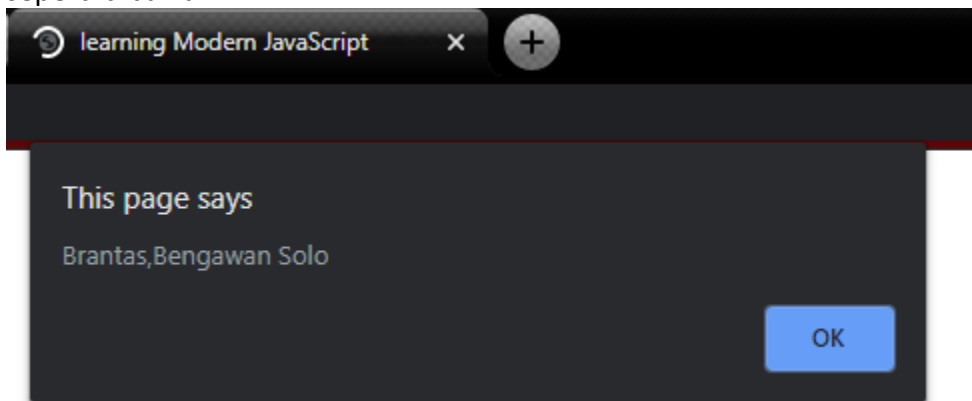
restO.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>learning Modern JavaScript</title>
6 </head>
7 <body>
8   <script src="rest0.js"></script>
9 </body>
10 </html>
```

restO.html

```
1 var rivers = ['Ciliwung', 'Brantas', 'Bengawan Solo'];
2 var [first, ...rest] = rivers;
3
4 alert(rest);
```

4. Selanjutnya buka file restO.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



k. Membuat Classes Constructor and Super

1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

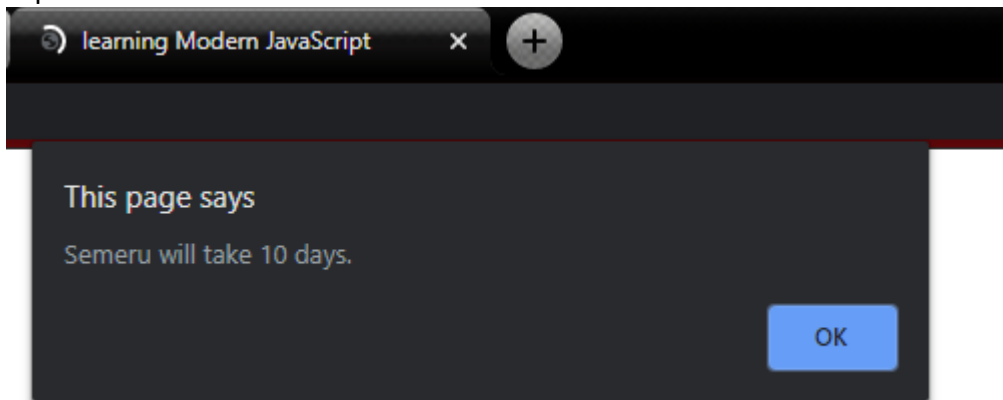
class.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="utf-8">
5  |   <title>learning Modern JavaScript</title>
6  </head>
7  <body>
8  |   <script src="class.js"></script>
9  </body>
10 </html>
```

class.js

```
1  // super class
2  class Holiday {
3  |   constructor(destination, days){
4  |       this.destination = destination;
5  |       this.days = days;
6  |   }
7  |   info(){
8  |       alert(`${this.destination} will take ${this.days} days.`);
9  |   }
10 }
11
12 // sub class
13 class Expedition extends Holiday {
14 |   constructor(destination, days, gear){
15 |       super(destination, days);
16 |       this.gear = gear;
17 |   }
18
19 |   info(){
20 |       super.info();
21 |       alert(`Bring your ${this.gear.join(' and your ')}`);
22 |   }
23 }
24
25 const tripWithGear = new Expedition('Semeru', 10, ['Sunglasses', 'Flags', 'Camera']);
26 tripWithGear.info();
```

2. Selanjutnya buka file class.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



3. Klik OK, maka akan muncul page selanjutnya dengan pesan

