Assignment01_Refresher

by Jiongfeng Chen, 02/22/2016

Javascript

Variable

Introduction of topic

Defined as containers for storing data values.

Question

Make a speech or introduction of myself by using variable (output to screen)

Code

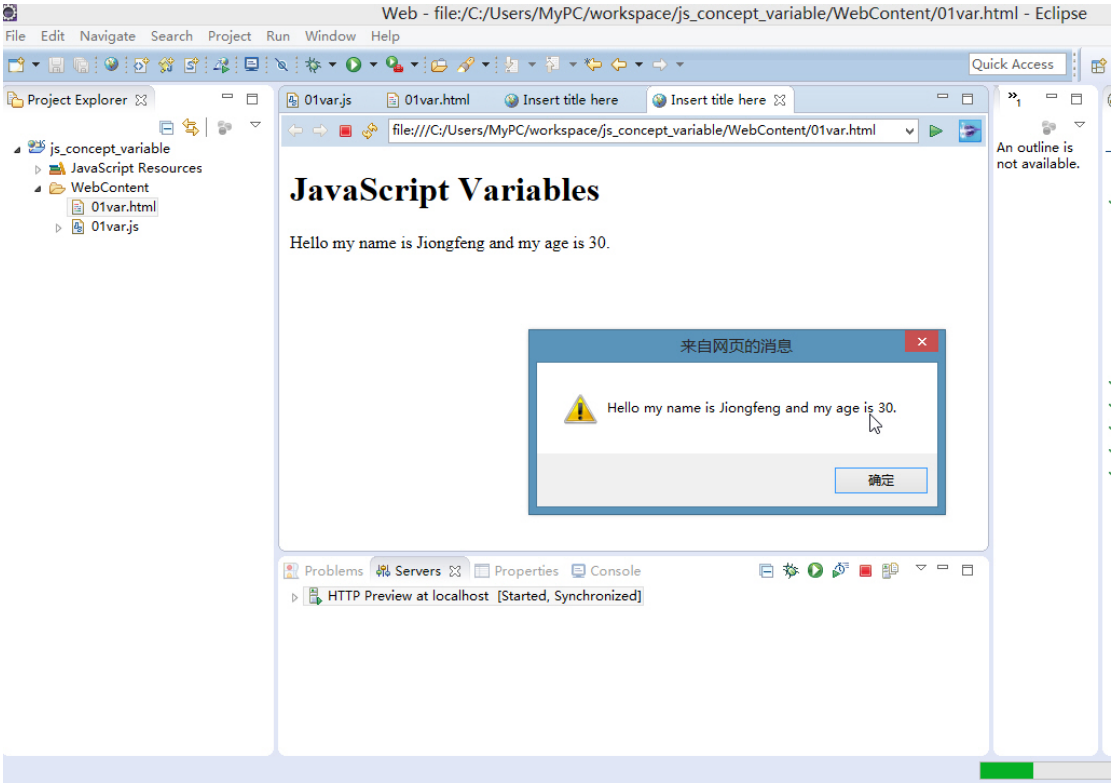
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<script type="text/javascript" src="01var.js"></script>
<title>Insert title here</title>
</head>
<body>

<h1>JavaScript Variables</h1>

<p id="speechLine"></p>

<script type="text/javascript">
var speech1 = speech();
document.getElementById("speechLine").innerHTML = speech1;
alert(speech());
</script>
</body>
```

```
</html>
```

"01var.js"

```
/**
 * question: give a small speech by using variables
 */
function speech()
{
    var age_a = 15;
    var age_b = 2;
    var age_c = age_a * age_b;
    var intro = "Hello Vedang, my name is";
    var name = "Jiongfeng";
    var intro_ful = intro +" "+ name;
    var speech = intro_ful + " and my age is " + age_c +".";
    return speech;
}
```

Output



Objects

Introduction of topic

Used to describe physical instance in life by giving some properties. For example, a television is an object. It has properties like brand, model, weight and color.

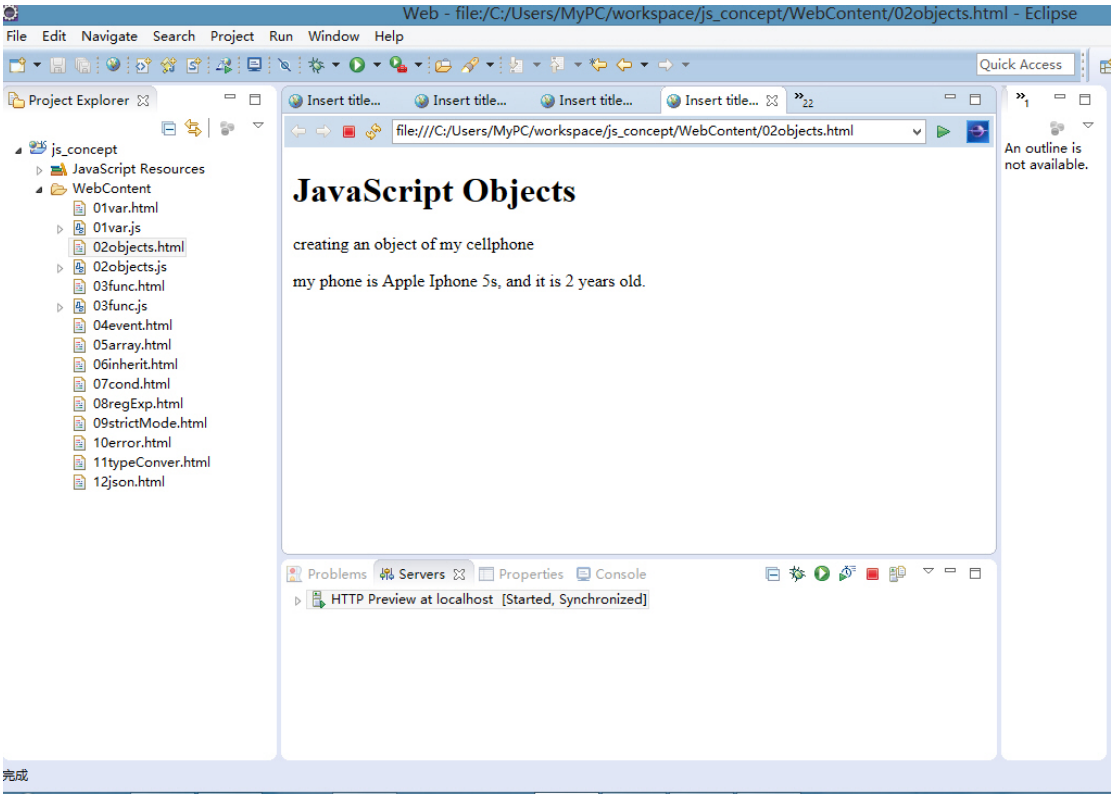
Question

Create an object of my cellphone

Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<script type="text/javascript" src="02objects.js"></script>
<title>Insert title here</title>
</head>
<body>

<h1>JavaScript Objects</h1>

<p>creating an object of my cellphone</p>
<p id="phone"></p>

<script type="text/javascript">
var cellp = {
    brandName : "Apple Iphone",
    model  : "5s",
    color  :  "black",
    years  : 2,
};
document.getElementById("phone").innerHTML = " my phone is " +
    cellp.brandName + " " + cellp.model + ", and it is " +
cellp.years + " years old.";
</script>

</body>
  </html>
```

Output



Functions

Introduction of topic

Defined as a block of code designed to perform a particular task. It is executed when something called.

Question

Create a function which can multiply two numbers

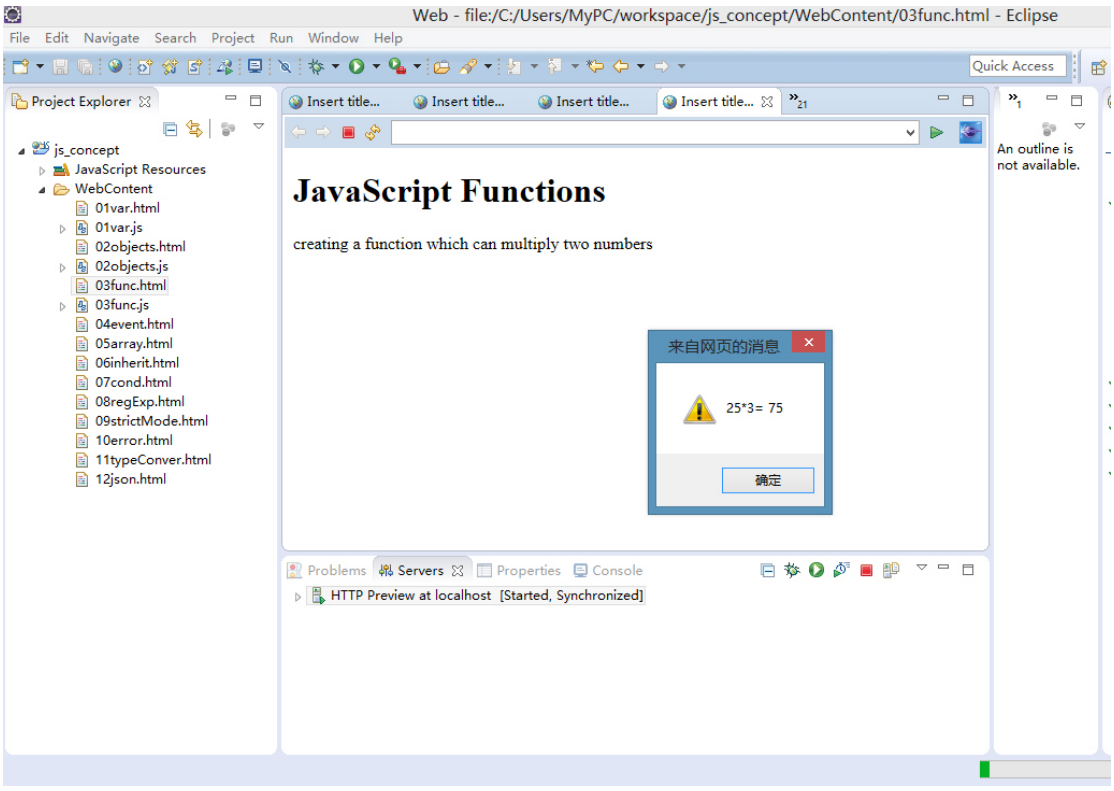
Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<script type="text/javascript" src="03func.js"></script>
<title>Insert title here</title>
</head>
<body>
```

```
<h1>JavaScript Functions</h1>

<p>creating a function which can multiply two numbers</p>

<script type="text/javascript">

alert("25*3= " + calc(25, 3));

</script>
</body>
  </html>
```

```
//03func.js
function calc(a, b)
{
    return a*b;
  }
```

Output



Events

Introduction of topic

Defined to things that happen to HTML elements, while JS can "react" on these events.

Question

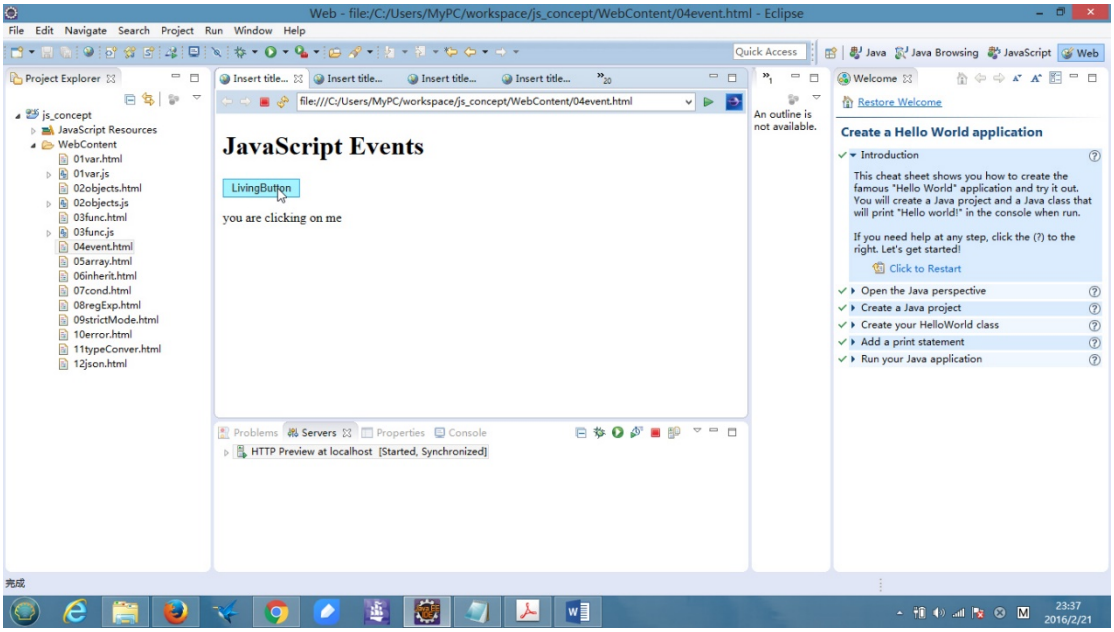
Create a button which can tell what you are doing by using event

Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<script type="text/javascript"></script>
<title>Insert title here</title>
</head>
<body>

<h1>JavaScript Events</h1>

<button onclick="click1()"
onmouseover="mouseover1()">LivingButton</button>

<script type="text/javascript">
function click1(){
    document.getElementById("text").innerHTML = "you are clicking on
me";
}
function mouseover1(){
    document.getElementById("text").innerHTML = "you are moving over
my head";
}
</script>
<p id="text"> </p>


</body>
```

```
</html>
```

Output



Arrays

Introduction of topic

Defined to store multiple values in a single variable.

Question

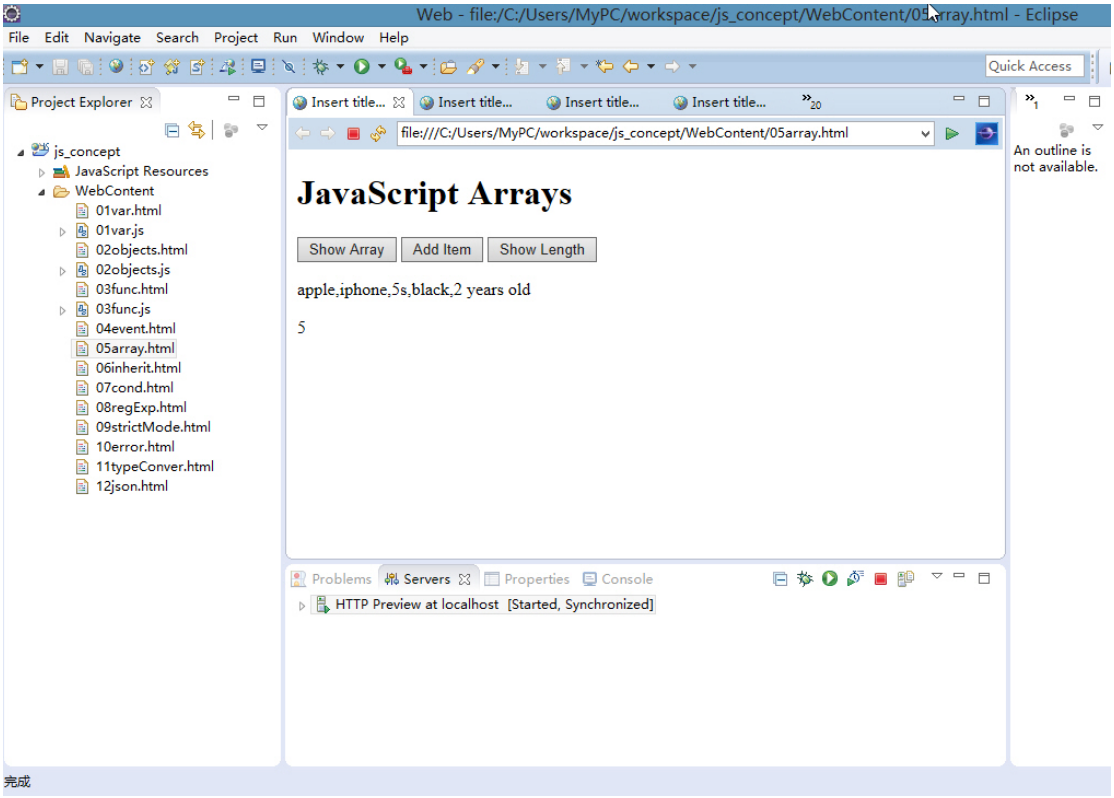
Add and detail some info of my phone by using array and tell how many things you know about my phone

Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<script type="text/javascript"></script>
<title>Insert title here</title>
</head>
<body>

<h1>JavaScript Arrays</h1>
```

```
<button onclick="click1()">Show Array</button>
<button onclick="click2()">Add Item</button>
<button onclick="click3()">Show Length</button>

<p id="text"> </p>
<p id="textNum"> </p>

<script type="text/javascript">
var phone = ["apple", "iphone", "5s", "black"];
function click1(){
    document.getElementById("text").innerHTML = phone;
}
function click2(){
    phone.push("2 years old");
    document.getElementById("text").innerHTML = phone;
}
function click3(){
    document.getElementById("textNum").innerHTML = phone.length;
}
</script>

</body>
  </html>
```

Output



Inheritance

Introduction of topic

In object-oriented programming, defined as inheritance is when an object or class is based on another object or class, using the same implementation specifying implementation to maintain the same behavior.

Question

Create an object of Pad product and create an Ipad product which is the inheritance from Pad

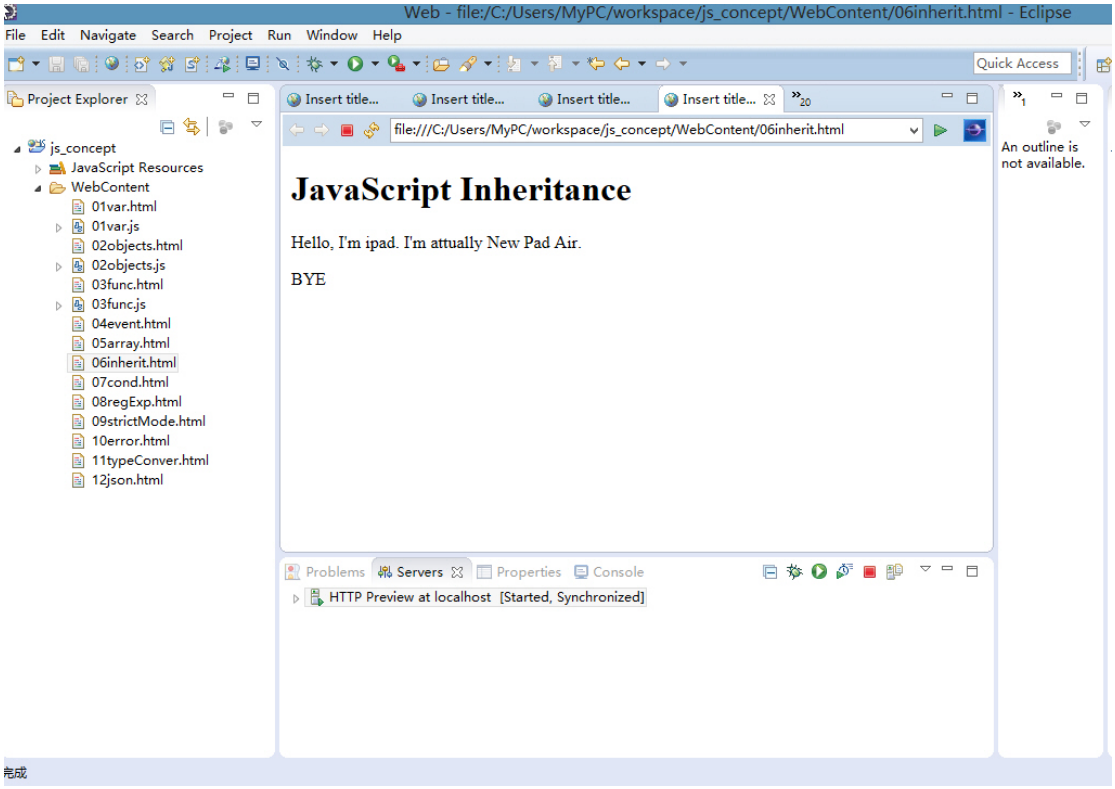
Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
```

```
<script type="text/javascript"></script>
<title>Insert title here</title>
</head>
<body>

<h1>JavaScript Inheritance</h1>

<p id="text1"> </p>
<p id="text2"> </p>

<script type="text/javascript">
//Define the Pad constructor
var Pad = function(Name) {
  this.Name = Name;
};
Pad.prototype.sayHello = function(){
    document.getElementById("text1").innerHTML = "Hello, I'm " +
this.Name;
};
Pad.prototype.sayBye = function(){
    document.getElementById("text2").innerHTML ="BYE";
};

function Ipad(Name, subject) {
  Pad.call(this, Name);
  this.subject = subject;
}
// Create a Ipad object that inherits from Pad prototype.
Ipad.prototype = Object.create(Pad.prototype);
Ipad.prototype.constructor = Ipad;
Ipad.prototype.sayHello = function(){
    document.getElementById("text1").innerHTML ="Hello, I'm " +
this.Name + ". I'm attually "
             + this.subject + ".";
};

var Ipad1 = new Ipad("ipad", "New Pad Air");
Ipad1.sayHello();
Ipad1.sayBye();
```

```
</script>

</body>
  </html>
```

Output



Conditions

Introduction of topic

Conditional statements will take different actions according to different situations by using if ... else et al.

Question

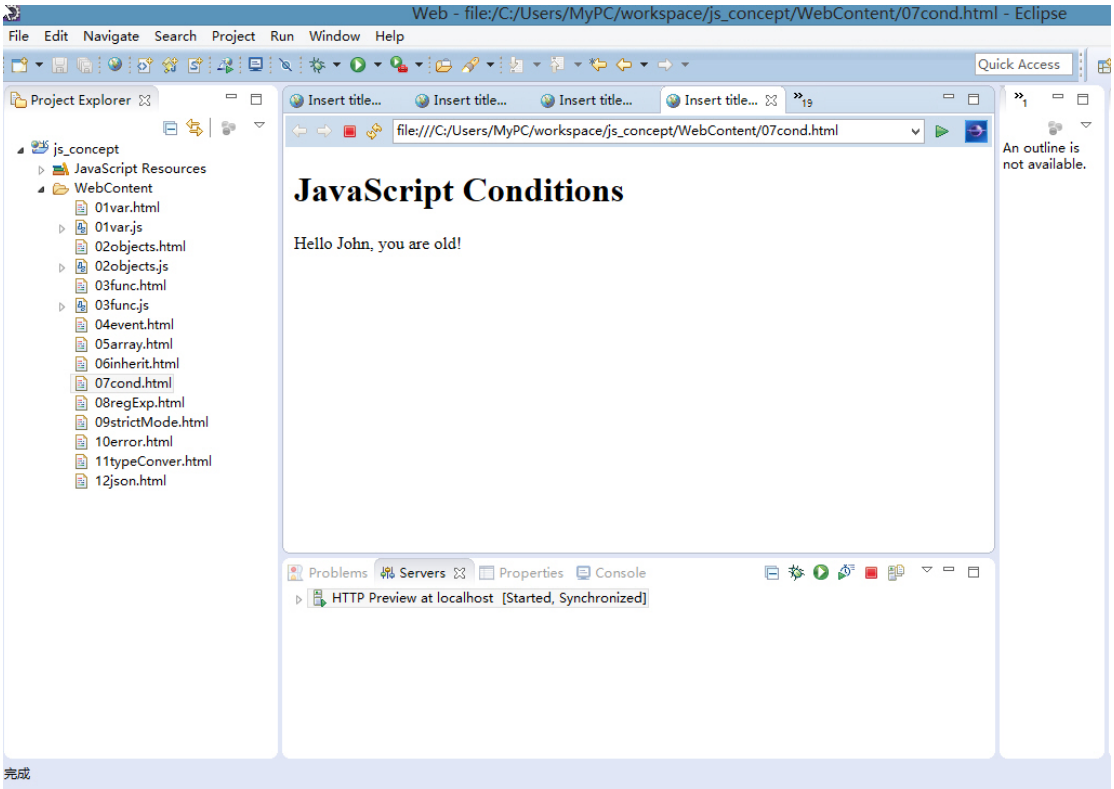
Tell me I am old or not, say not younger than 30 is old

Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
```

```
<script type="text/javascript"></script>
<title>Insert title here</title>
</head>
<body>

<h1>JavaScript Conditions</h1>

<p id="text1"> </p>
<script type="text/javascript">
var age = 30;

if (age >= 30) {
    document.getElementById("text1").innerHTML = "Hello John, you are
old!";
}
</script>

</body>
  </html>
```

Output



Regular Expressions

Introduction of topic

It is a sequence of characters that forms a search pattern, which can be used for text search and text replace operations.

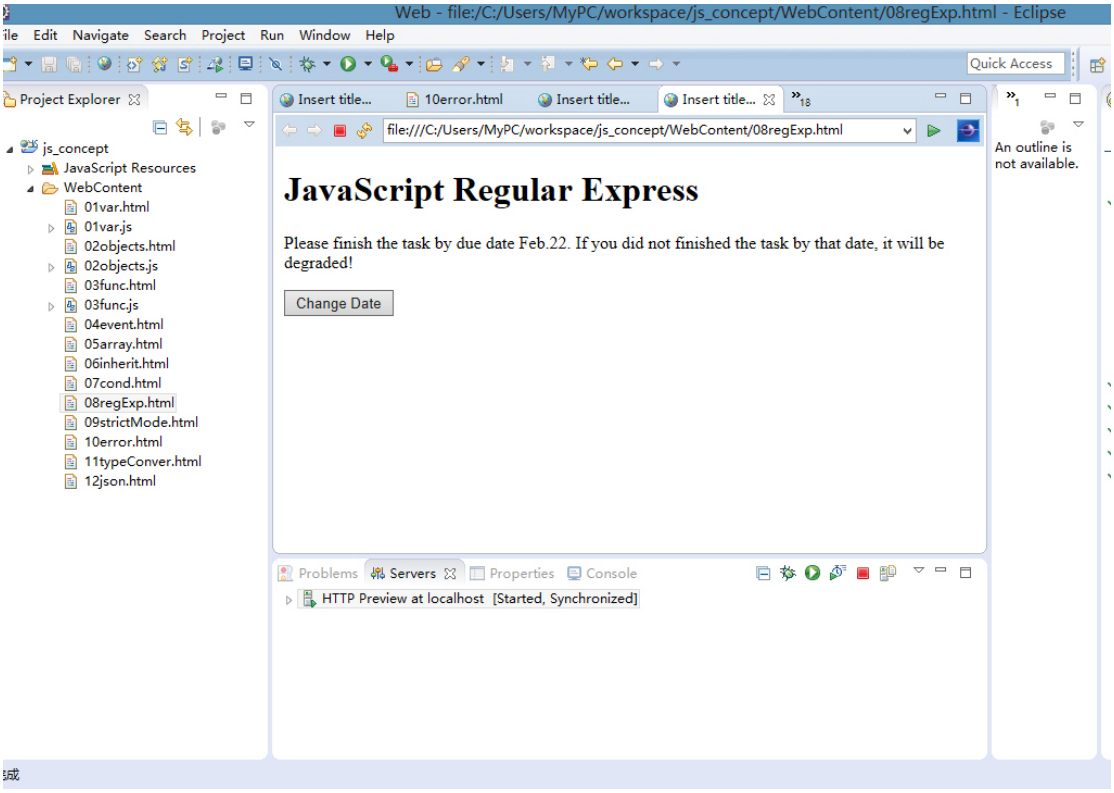
Question

Change the assignment due date in the text from Feb.23 to Feb.22 using RegExp

Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<script type="text/javascript"></script>
<title>Insert title here</title>
</head>
<body>

<h1>JavaScript Regular Express</h1>

<p id="text1">Please finish the task by due date Feb.23. If you did
not finished the task by that date, it will be degraded! </p>

<button onclick="changeDate()">Change Date</button>

<script type="text/javascript">
function changeDate() {
    var str1 = document.getElementById("text1").innerHTML;
    var str2 = str1.replace(/Feb.23/,"Feb.22");
    document.getElementById("text1").innerHTML = str2;
}
</script>

</body>
  </html>
```

Output



Strict mode

Introduction of topic

It is new in JavaScript 1.8.5 (ECMAScript version 5). JavaScript code should be executed strictly under this programing mode. With strict mode, we cannot, i.e., use undeclared variables.

Question

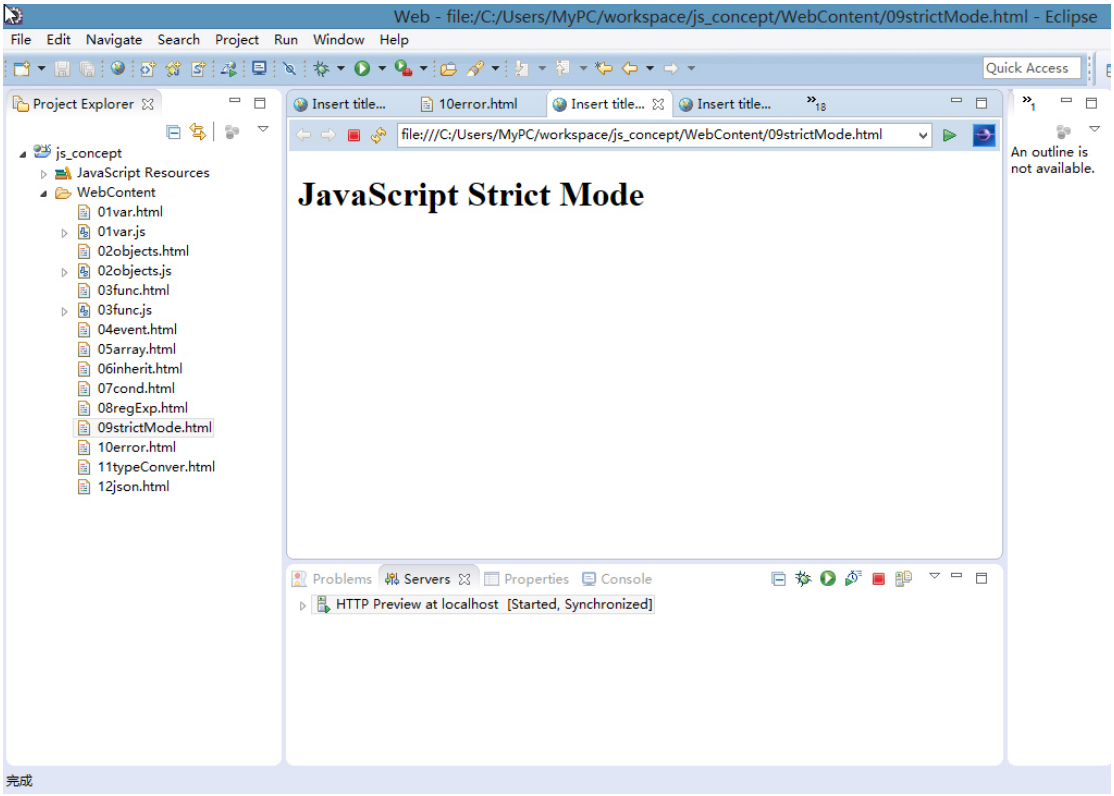
Check the difference of variable outputs when using the javascript under strict mode (as under strict mode, “not defined” will shows error and blank/no output)

Code

```
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="UTF-8">
<script type="text/javascript"></script>
<title>Insert title here</title>
</head>
<body>

<h1>JavaScript Strict Mode</h1>

<script type="text/javascript">
"use strict";
var a = 5.3;               // a is not defined, will cause an error
alert(a);
var arguments = 8.82;    // arguments not allowed, will cause an
error
alert(arguments);
var a = 010;            // Octal numeric literals are not allowed
alert(a);
</script>

</body>
  </html>
```

Output



Errors

Introduction of topic

Errors are equal to the try to catch and throw grammar. The try statement test a block of code for errors. The catch statement handle the error. The throw statement create custom errors.

Question

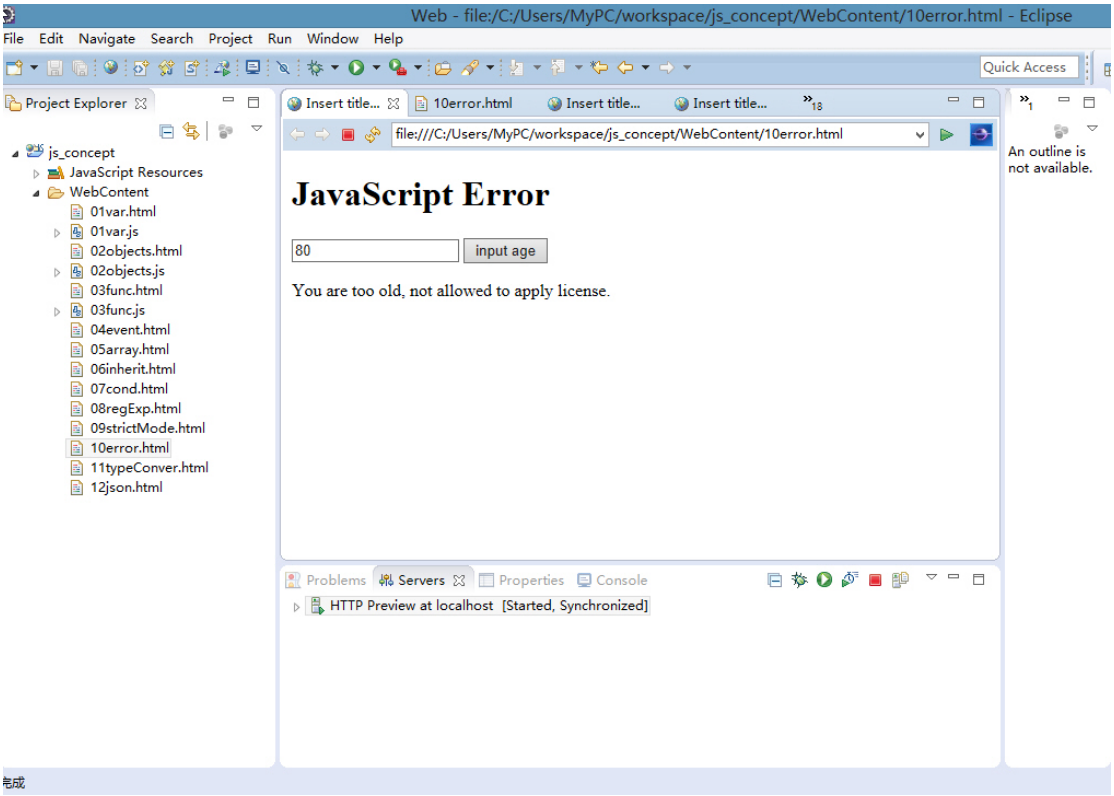
Input the age to apply for the driver license, too old or too young will throw error

Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<script type="text/javascript"></script>
<title>Insert title here</title>
```

```
</head>
<body>

<h1>JavaScript Error</h1>

<input id="age" type="text">
<button type="button" onclick="click1()">input age</button>
<p id="text1"></p>

<script type="text/javascript">
function click1() {
    var showText, x;
    showText = document.getElementById("text1");
    showText.innerHTML = "";
    x = document.getElementById("age").value;
    try {
        x = Number(x);
        if(x < 18)    throw "too young, not allowed to apply
license.";
        if(x > 70)   throw "too old, not allowed to apply license.";
    }
    catch(err) {
        showText.innerHTML = "You are " + err;
    }
}

</script>

</body>
  </html>
```

Output



Type Conversions

Introduction of topic

Javascript convert types among Number, String, Boolean, et al.

Question

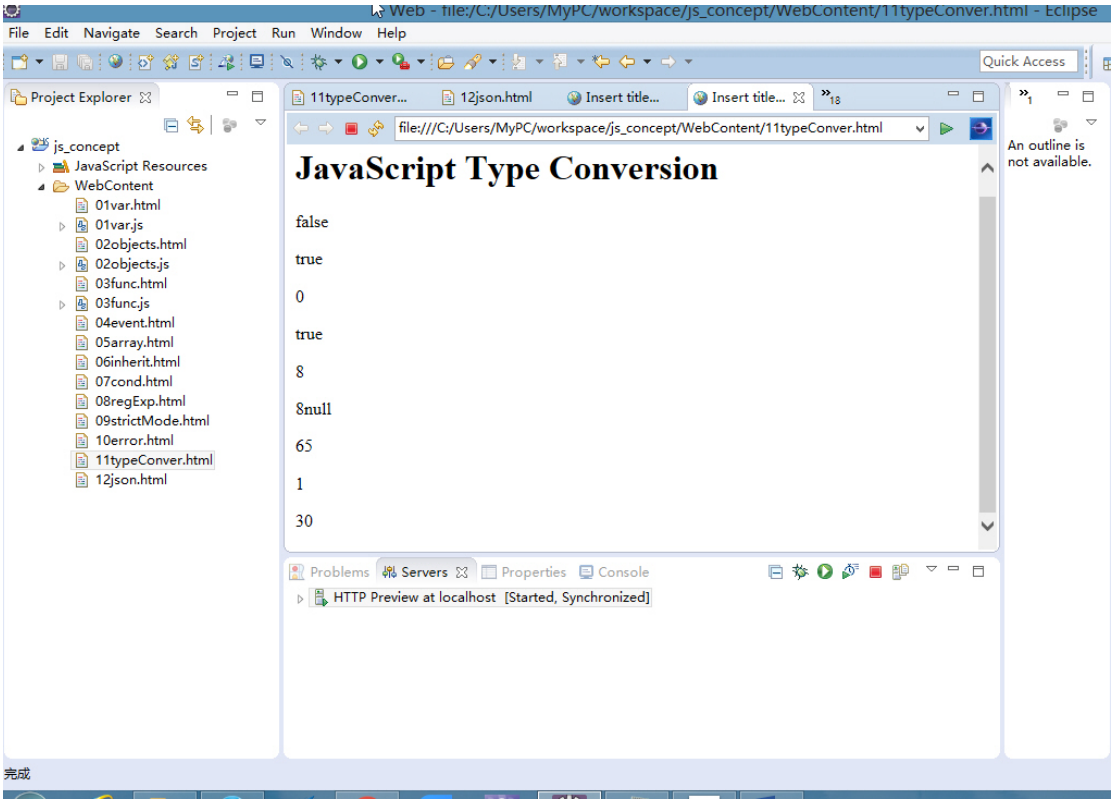
Create some test cases of conversion from string to number, number to string and print it, to see how type conversion work

Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<script type="text/javascript"></script>
<title>Insert title here</title>
</head>
<body>
```

```
<h1>JavaScript Type Conversion</h1>
<p id="text1"> </p>
<p id="text2"> </p>
<p id="text3"> </p>
<p id="text4"> </p>
<p id="text5"> </p>
<p id="text6"> </p>
<p id="text7"> </p>
<p id="text8"> </p>
<p id="text9"> </p>

<script type="text/javascript">
document.getElementById("text1").innerHTML=false.toString()
document.getElementById("text2").innerHTML=true.toString()
document.getElementById("text3").innerHTML=Number(false)
document.getElementById("text4").innerHTML=true
document.getElementById("text5").innerHTML=8 + null
document.getElementById("text6").innerHTML="8" + null
document.getElementById("text7").innerHTML="6" + 5
document.getElementById("text8").innerHTML="6" - 5
document.getElementById("text9").innerHTML="6" * "5"
</script>

</body>
  </html>
```

Output



JSON

Introduction of topic

JSON stands for JavaScript Object Notation, it is a format for storing and transporting data, and often used when data is sent from a server to a web page.

Question

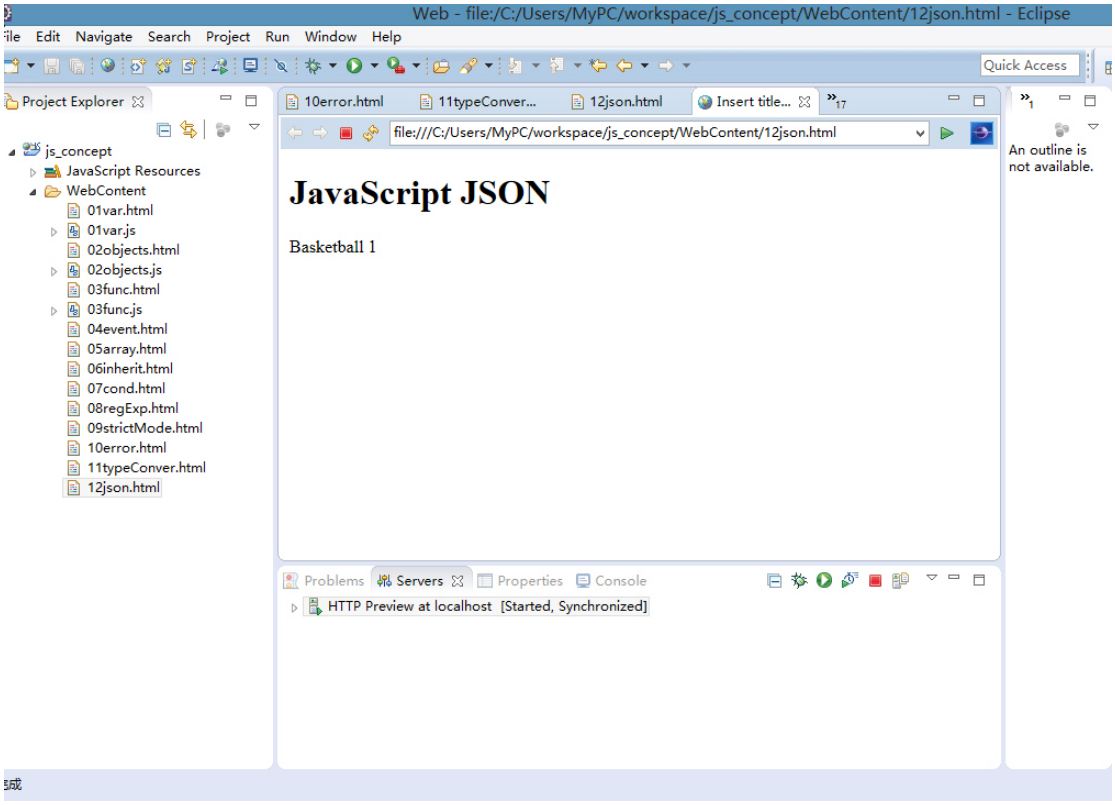
Create a JASON array of my favorite sports list, and print my most favorite one on the screen

Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<script type="text/javascript"></script>
```

```
<title>Insert title here</title>
</head>
<body>

<h1>JavaScript JSON</h1>

<p id="text1"> </p>

<script type="text/javascript">
var sports = '{"sports":[' +
'{"Name":"Basketball","Rank":"1" },' +
'{"Name":"Football","Rank":"2" },' +
'{"Name":"Swimming","Rank":"3" }]}';
obj = JSON.parse(sports);
document.getElementById("text1").innerHTML = obj.sports[0].Name + "
" + obj.sports[0].Rank;
</script>

</body>
  </html>
```

Output



HTML5

Local Storage

Introduction of topic

It stores data with no expiration date, unlike session storage, which stores data for one session. The data will be available even when the browser or browsing tab is closed or reopened. It is a simple client side database that allows the users to persist data in the form of key pairs or value pairs. It has a fairly simple API to write data into the local storage, it can store up to 10MB of data per domain. Unlike cookies, the data stored are not included with every HTTP request.

Question

Add number and save it to the local storage, and output to the screen. And set the color and save it to local storage, when the webpage is opened again, try to load the color as background color.

Code

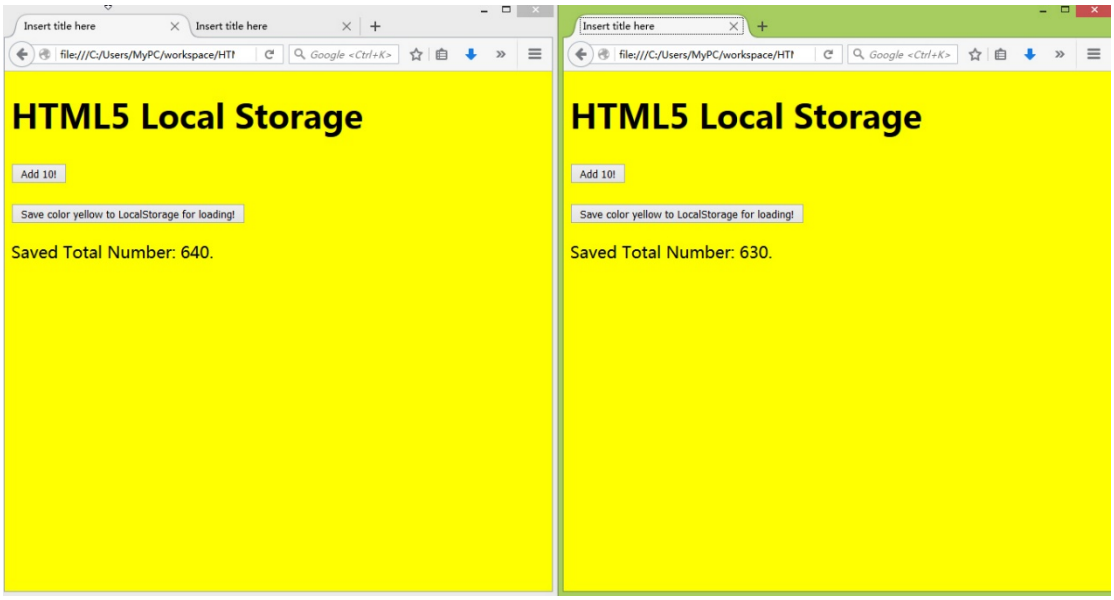
//Open two tabs and click the button alternatively to see how the local saved value changes.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script type="text/javascript" src="01LocStor.js"></script>
</head>
<body onload="changeColor()">

<h1>HTML5 Local Storage</h1>
```

```
<p><button onclick="counter1()" type="button">Add 10!</button></p>
<p><button onclick="setColor()" type="button">Save color yellow to
LocalStorage for loading!</button></p>

<div id="showNum"></div>
<script type="text/javascript">

</script>

</body>
</html>


//01LocStor.js
/**
 * question: save total number users give by clicking the button, and
reload the number when the page reload
 */
function counter1() {
    if(typeof(Storage) !== "undefined") {
    try
    {
            if (localStorage.numberTotal) {
                localStorage.numberTotal =
Number(localStorage.numberTotal)+10;
            } else {
                localStorage.numberTotal = 10;
            }
            document.getElementById("showNum").innerHTML = "Saved Total
Number: " + localStorage.numberTotal + ".";
    }
    catch (e)
    {
            if (e == QUOTA_EXCEEDED_ERR) {
                alert('Quota exceeded!');
        }
    }
    } else {
        document.getElementById("showNum").innerHTML = "does not support web
storage";
    }
}
function setColor() {
    if(typeof(Storage) !== "undefined") {
```

```
    try
    {
        localStorage.fontsize = 20;
        localStorage.bgcolor = '#ffff00';

    }
    catch (e)
    {
            if (e == QUOTA_EXCEEDED_ERR) {
                alert('Quota exceeded!');
        }
    }
    } else {
        document.getElementById("showNum").innerHTML = "does not support web
storage";
    }
}

function changeColor()
{
    if (localStorage.length != 0) {
    document.body.style.backgroundColor = localStorage.bgcolor;
    document.body.style.fontSize = localStorage.fontsize + 'px';
    document.getElementById("showNum").innerHTML = "!=0" +
localStorage.bgcolor;
    } else {
    document.body.style.backgroundColor = '#000000';
    document.body.style.fontSize = '50px'
        document.getElementById("showNum").innerHTML = "0" +
localStorage.bgcolor;
    }
}
```

Output



Media (Video and Audio)

Introduction of topic

It is used to play video and audio on the website

Question

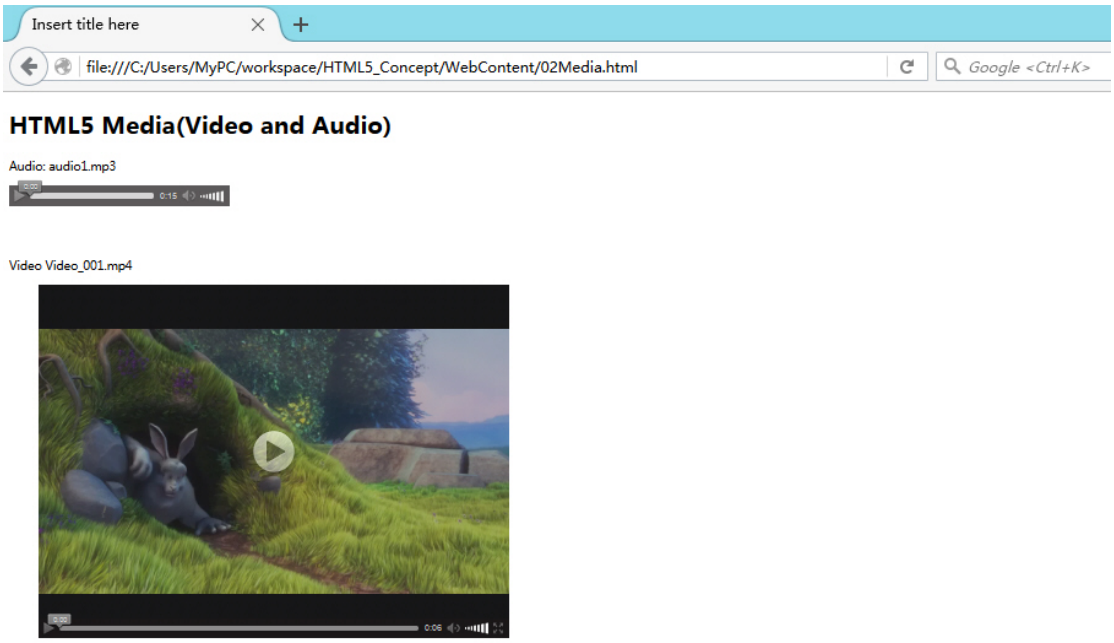
Download a sample video file and audio file and put them on the web page.

Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script type="text/javascript"></script>
</head>


<h1>HTML5 Media(Video and Audio)</h1>

<p>Audio: audio1.mp3 </p>
<audio controls>
  <source src="audio1.mp3" type="audio/mpeg">
</audio>
```

```
<br> <br><br>
<p>Video Video_001.mp4</p>
<video width="720" height="480" controls>
  <source src="Video_001.mp4" type="video/mp4">
</video>

<script type="text/javascript">
</script>

</body>
</html>
```

Output (the video file is downloaded from a free website of internet)



Input Type (make use of different input property options in HTML5 like patterns, autofocus, required, email etc. Place types you want, mention the properties used in your Introduction to Topic section)

Introduction of topic

Describe the input types of the <input> element, and it has different input property options. In this part, the properties, disabled, readonly, autofocus, required , maxlength, pattern, autocomplete, email are used in the code to help to design a web page for input the information. For example, the university name is set to SJSU, as the students are all coming from the same university. As the output screenshot shows, if we did not input last name for the input type with attribute “required”, it shows read color on it. Similarly if we did not match the pattern, it also shows red color.

Question

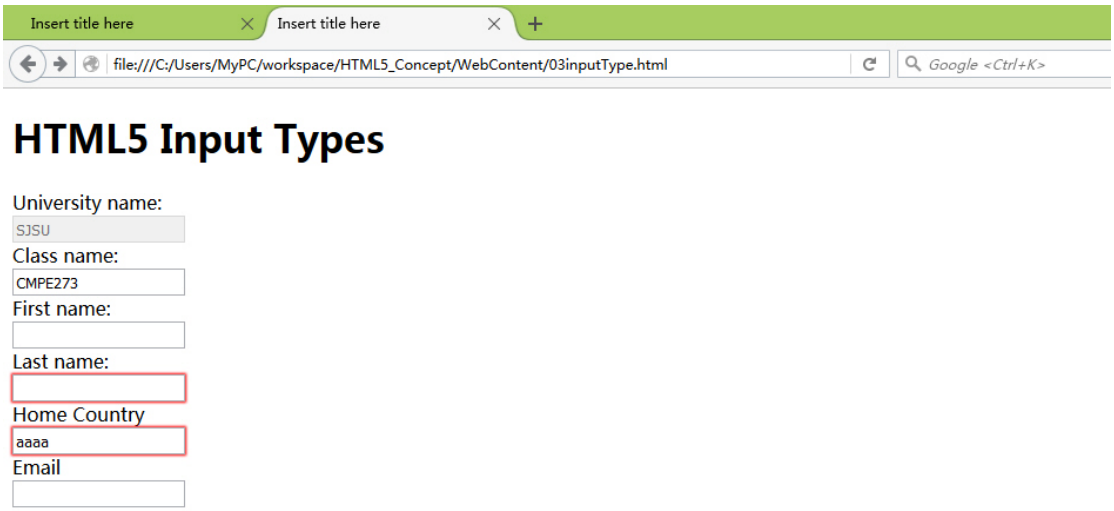
Make a web page to register your information on internet for class CMPE273

Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script type="text/javascript"></script>
</head>

<h1>HTML5 Input Types</h1>

<form>
University name:<br>
<input type="text" name="university" value="SJSU" disabled>
<br>Class name:<br>
<input type="text" name="classname" value="CMPE273" readonly>
<br>First name:<br>
<input type="text" name="fname" autofocus>
<br>Last name:<br>
```

```
<input type="text" name="lname" required maxlength="20">
<br>Home Country<br>
<input type="text" name="countrycode" pattern="[A-Za-z]{3}" title="3
letter">
<br>Email<br>
<input type="email" name="email" autocomplete="off">
</form>

<script type="text/javascript">
</script>

</body>
</html>
```

Output



Geolocation

Introduction of topic

The HTML Geolocation API is used to locate a user's position.

Question

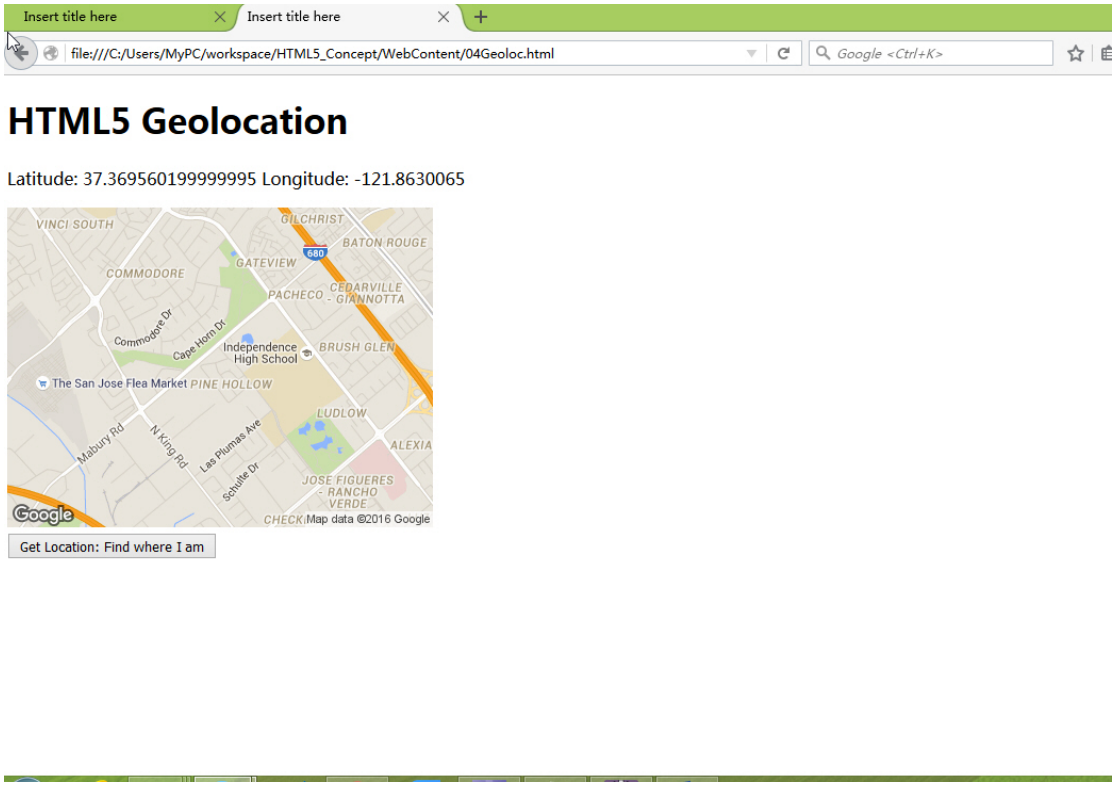
Design a web page to find where I am and show the map

Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script type="text/javascript"></script>
</head>


<h1>HTML5 Geolocation</h1>

<p id="showStatus"></p>
<div id="mapDiv"></div>
<button onclick="getLoc()">Get Location: Find where I am</button>

<script>
var a = document.getElementById("showStatus");

function getLoc() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(getPos, showError);
    } else {
        a.innerHTML = "Geolocation is not supported.";
    }
}

function getPos(position) {
    a.innerHTML = "Latitude: " + position.coords.latitude + " Longitude: "
+ position.coords.longitude;
    var pos_xy = position.coords.latitude + "," +
position.coords.longitude;
    var img_url =
"http://maps.googleapis.com/maps/api/staticmap?center="+pos_xy+"&zoom=14&si
ze=400x300&sensor=false";
    document.getElementById("mapDiv").innerHTML = "<img
src='"+img_url+"'>";
}

function showError(error) {
    switch(error.code) {
```

```
        case error.UNKNOWN_ERROR:
            a.innerHTML = "Unknown error occurred."
            break;
        case error.TIMEOUT:
            a.innerHTML = "Get user location timed out."
            break;
            case error.POSITION_UNAVAILABLE:
                a.innerHTML = "Location information is unavailable."
            break;
            case error.PERMISSION_DENIED:
                a.innerHTML = "Geolocation request denied by user."
            break;
    }
}
</script>


</body>
</html>
```

Output



JAVA

Queues

Introduction of topic

It is a kind of data structure. In Java, it is an interface, which is a subtype of the Collection interface.

Question

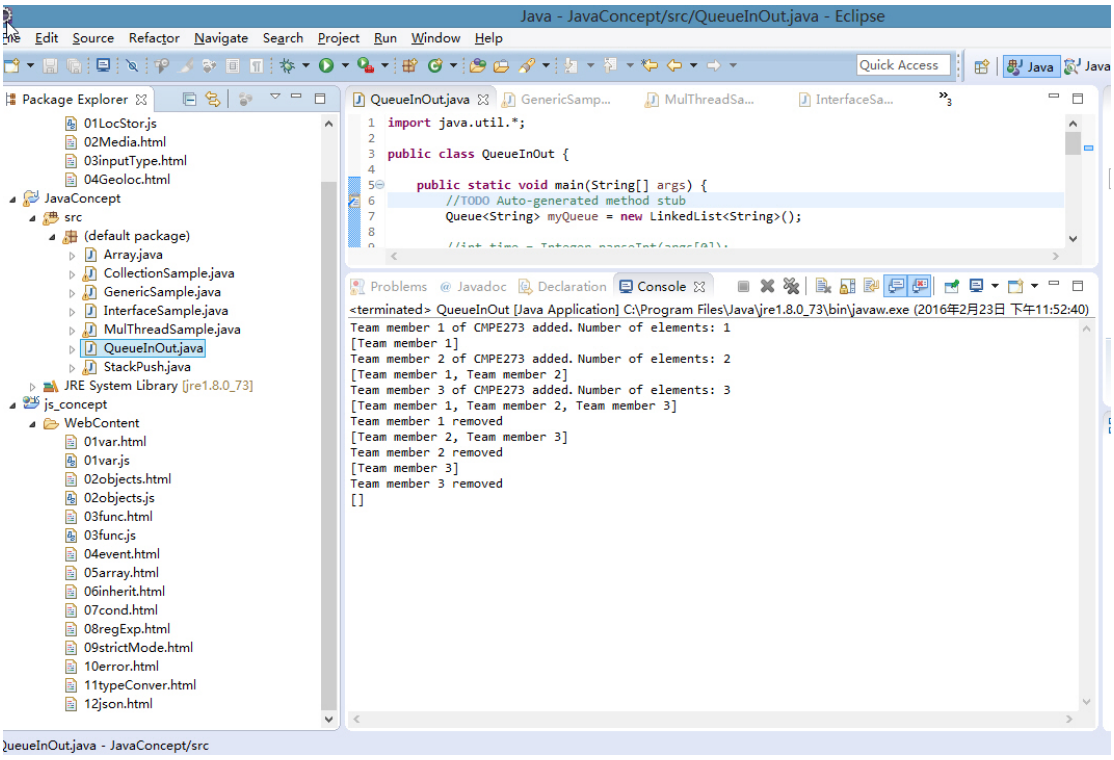
Create and process a sample queue containing string of "CMPE class 273"

Code

```
import java.util.*;

public class QueueInOut {

    public static void main(String[] args) {
        //TODO Auto-generated method stub
        Queue<String> myQueue = new LinkedList<String>();

        //int time = Integer.parseInt(args[0]);
        //Queue <Integer> queue = new LinkedList<Integer>();
        for (int i = 1; i <= 3; i++)
        {
        myQueue.add("Team member " + i );
        System.out.println(myQueue.toArray()[myQueue.size()-1] +
                " of CMPE273 added。" + "Number of elements: " +
myQueue.size());
        System.out.println(Arrays.toString(myQueue.toArray()));
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
```

```
        while (!myQueue.isEmpty()) {
            System.out.println(myQueue.remove()+" removed");
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println(Arrays.toString(myQueue.toArray()));
        }
    }

}
```

Output



Test using Junit Framework

```
import static org.junit.Assert.assertEquals;
import org.junit.Test;
public class QueueInOutTest {
@Test
public void evaluatesExpression() {
    QueueInOut queueInOut = new QueueInOut();
assertEquals("[This is, Class]", queueInOut.outScreen ());
```

```
}
  }

public class QueueInOut {
    public static String outScreen() {
        //TODO Auto-generated method stub
        Queue<String> myQueue = new LinkedList<String>();

        //int time = Integer.parseInt(args[0]);
        //Queue <Integer> queue = new LinkedList<Integer>();
        for (int i = 1; i <= 3; i++)
        {
        myQueue.add("Team member " + i );
        System.out.println(myQueue.toArray()[myQueue.size()-1] +
                " of CMPE273 added。" + "Number of elements: " +
myQueue.size());
        System.out.println(Arrays.toString(myQueue.toArray()));
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        while (!myQueue.isEmpty()) {
            System.out.println(myQueue.remove()+" removed");
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println(Arrays.toString(myQueue.toArray()));
        }
    }
    return Arrays.toString(myQueue.toArray();

}
}
```

Stacks

Introduction of topic

A Stack is a data structure where you add elements to the "top" of the

stack, and also remove elements from the top again. In Java, it is an class found as java.util.Stack class.

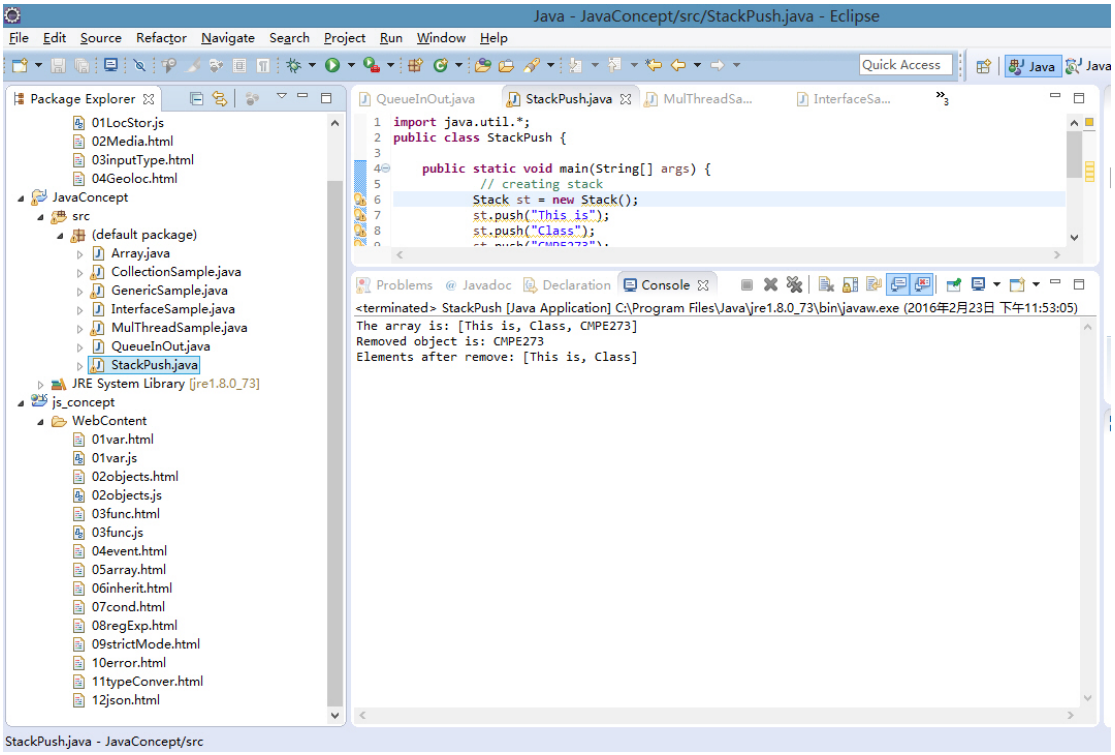
Question

Create and process a sample stack containing string of "CMPE class 273"

Code

```
import java.util.*;
public class StackPush {

    public static void main(String[] args) {
            // creating stack
           Stack st = new Stack();
           st.push("This is");
           st.push("Class");
           st.push("CMPE273");

           System.out.println("The array is: "+st);
           // removing one object
           System.out.println("Removed object is: "+st.pop());
           System.out.println("Elements after remove: "+st);
    }

}
```

Output



Arrays

Introduction of topic

It is a kind of data structure, which stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data.

Question

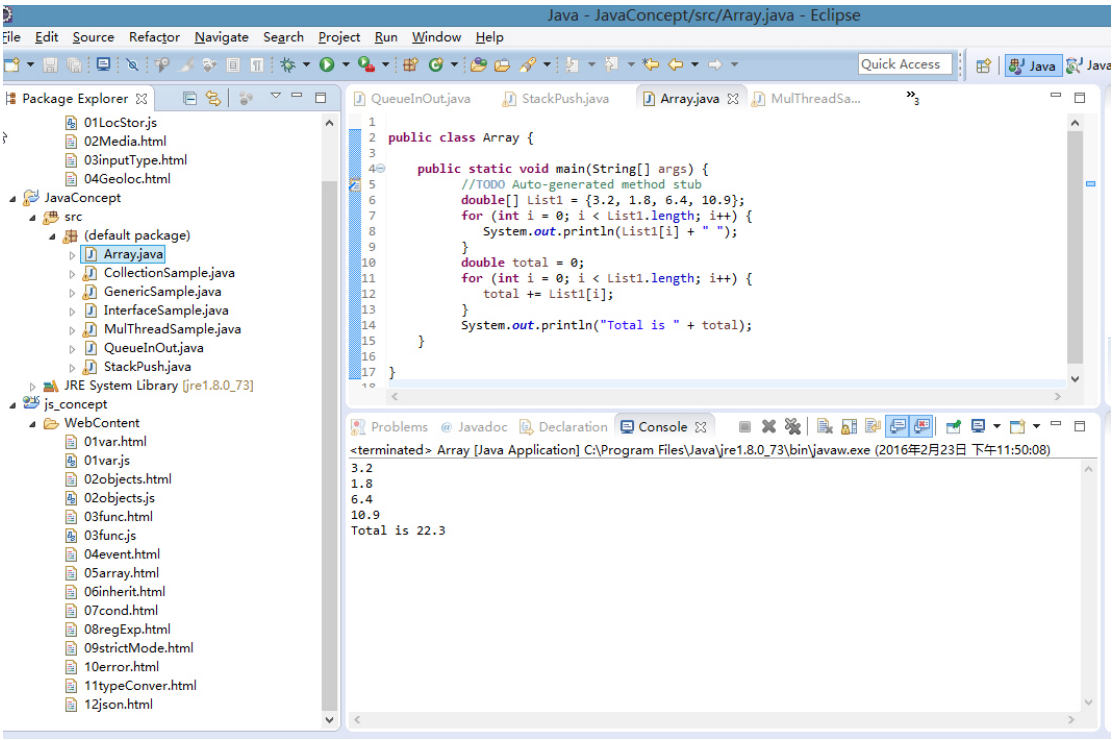
Create, initialize and process a sample arrays containing some digit numbers.

Code

```
public class Array {

   public static void main(String[] args) {
         //TODO Auto-generated method stub
         double[] List1 = {3.2, 1.8, 6.4, 10.9};
         for (int i = 0; i < List1.length; i++) {
            System.out.println(List1[i] + " ");
```

```
        }
        double total = 0;
        for (int i = 0; i < List1.length; i++) {
           total += List1[i];
        }
        System.out.println("Total is " + total);
    }

}
```

Output



Interfaces

Introduction of topic

An interface is a reference type in Java, and it is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface.

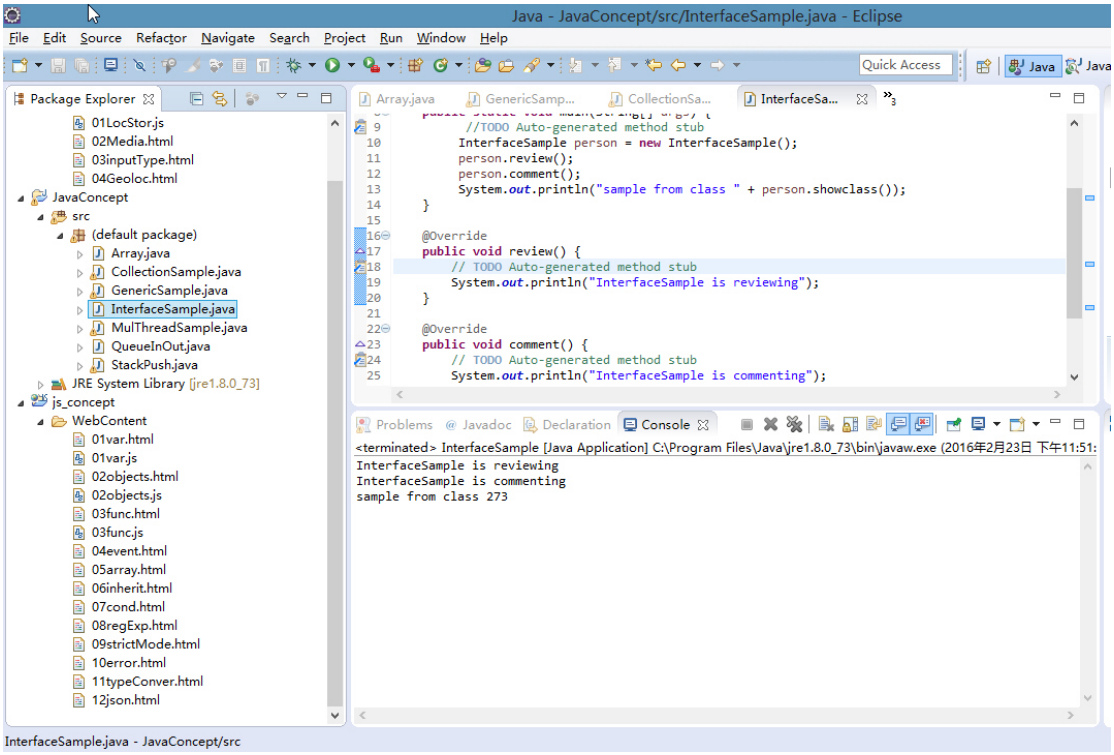
Question

Give a sample class which implements defined interface reviewer.

Code

```
interface Reviewer {
   public void review();
   public void comment();
}

public class InterfaceSample implements Reviewer {

    public static void main(String[] args) {
          //TODO Auto-generated method stub
         InterfaceSample person = new InterfaceSample();
         person.review();
         person.comment();
         System.out.println("sample from class " + person.showclass());
    }

    @Override
    public void review() {
        // TODO Auto-generated method stub
        System.out.println("InterfaceSample is reviewing");
    }

    @Override
    public void comment() {
        // TODO Auto-generated method stub
        System.out.println("InterfaceSample is commenting");
    }

    public int showclass(){
          return 273;
     }
}
```

Output



Collections

Introduction of topic

A collection, sometimes called a container, is simply an object that groups multiple elements into a single unit. Collections are used to store, retrieve, manipulate, and communicate aggregate data. Typically, they represent data items that form a natural group, such as a poker hand, a mail folder, or a telephone directory.

Question

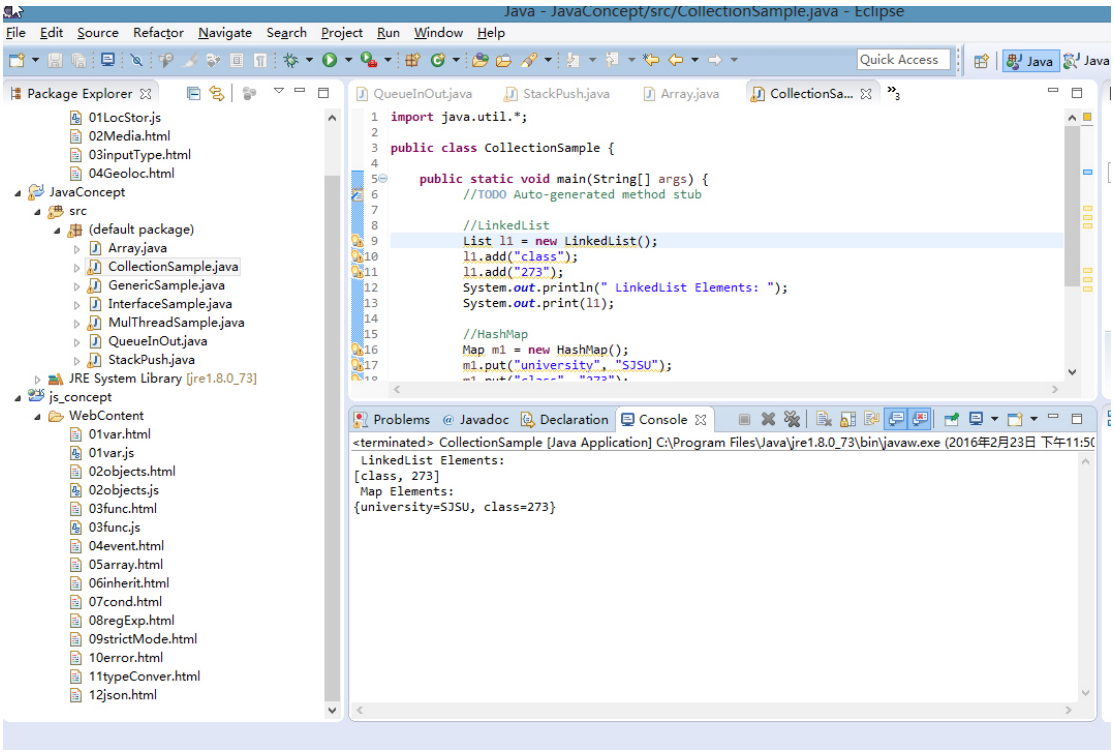
Give various class examples to show implementations of a collection interface method, for example, add().

Code

```
import java.util.*;

public class CollectionSample {
```

```
    public static void main(String[] args) {
        //TODO Auto-generated method stub

        //LinkedList
        List l1 = new LinkedList();
        l1.add("class");
        l1.add("273");
        System.out.println(" LinkedList Elements: ");
        System.out.print(l1);

        //HashMap
        Map m1 = new HashMap();
        m1.put("university", "SJSU");
        m1.put("class", "273");
        System.out.println();
        System.out.println(" Map Elements: ");
        System.out.print(m1);

    }

}
```

Output



Generics

Introduction of topic

Java Generic methods and generic classes enable programmers to specify, with a single method declaration, a set of related methods or, with a single class declaration, a set of related types, respectively. Using Java Generic concept, we might write a generic method for sorting an array of objects.

Question

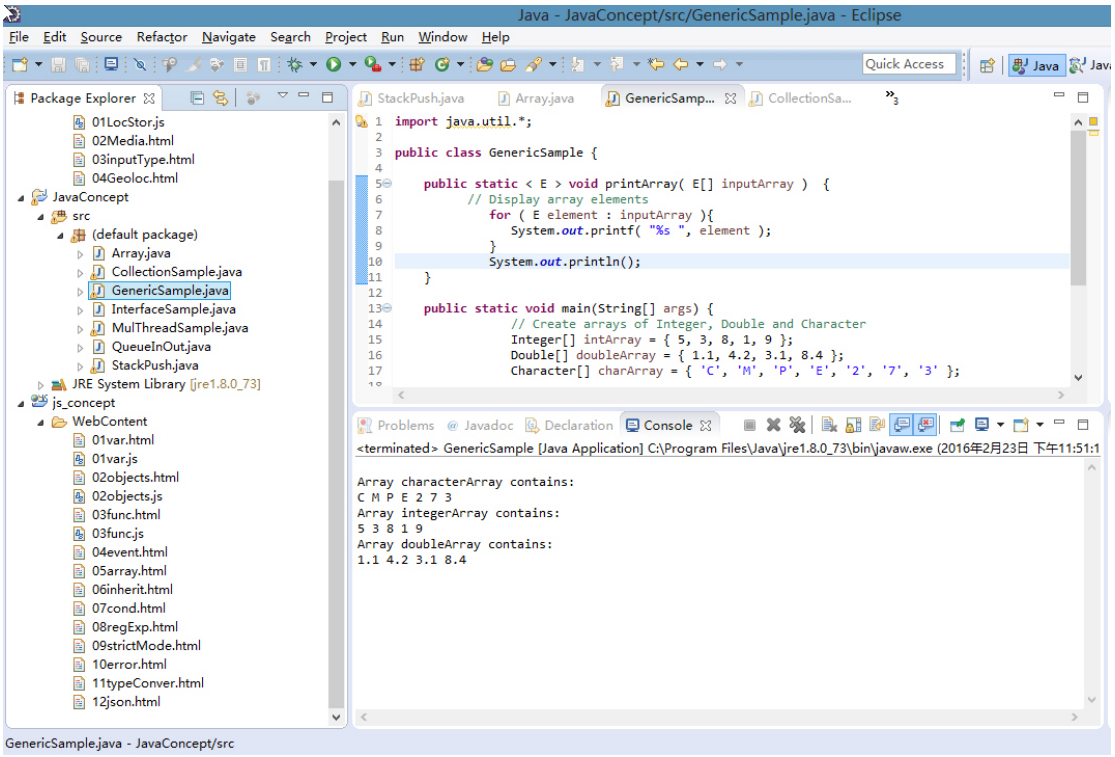
Give a Java Generics method print which can handle different input types.

Code

```
import java.util.*;

public class GenericSample {

    public static < E > void printArray( E[] inputArray )  {
         // Display array elements
            for ( E element : inputArray ){
               System.out.printf( "%s ", element );
            }
            System.out.println();
    }

    public static void main(String[] args) {
               // Create arrays of Integer, Double and Character
               Integer[] intArray = { 5, 3, 8, 1, 9 };
               Double[] doubleArray = { 1.1, 4.2, 3.1, 8.4 };
               Character[] charArray = { 'C', 'M', 'P', 'E', '2', '7',
'3' };

               System.out.println( "\nArray characterArray contains:" );
               printArray( charArray ); // pass a Character array
               System.out.println( "Array integerArray contains:" );
               printArray( intArray  ); // pass an Integer array
               System.out.println( "Array doubleArray contains:" );
```

```
                printArray( doubleArray ); // pass a Double array
            }
}
```

Output



Multithreading

Introduction of topic

A multi threaded program contains two or more small programs that can run concurrently and each one can handle different task at the same time making optimal use of the available resources specially multiple CPUs.

Question

When we play some games in the phone or pad,we need to use both hands to control handle at the same time. So design a thread representing task one to keep pressing the button on the left and

another thread representing pressing the button on the right. Create the two running threads, and show the running time since start. After 5 seconds stop task one, and after 10 seconds stop task two, to finish the concurrent running situation.

Code

```
import java.util.*;

//create a thread: new thread? -> thread combined with interface
"Runnable", so create class
//output thread status: where?? -> three methods, public create; public
void run(); public void start;
//start a thread: thread.run()? -> Thread(Runnable this, String name);
Thread.start(); Thread.sleep(1000);
class TaskConcur implements Runnable {

    private Thread T;
    private String TaskName;
    private int ProcDur;

    //create
    public TaskConcur(String strIn, int numDur) {
        TaskName = strIn;
        ProcDur = numDur;
    }
    public void run() {
    //business logic
        for(int i=ProcDur; i>=0; i--) {
            try {
                System.out.println(TaskName + "ing " + (ProcDur-i) + "
seconds.");
                Thread.sleep(1000);
            } catch(InterruptedException e) {
                System.out.println(TaskName + " interrupted");
            }
        }
        System.out.println(TaskName + " exit.");
    }
    public void start() {
        T = new Thread(this, TaskName);
```

```
            T.start();
        }
}

public class MulThreadSample {
    public static void main(String[] args) {
        TaskConcur taskOne = new TaskConcur("Left Hand Press", 5);
        taskOne.start();
        TaskConcur taskTwo = new TaskConcur("Right Hand Press", 10);
        taskTwo.start();
    }
}
```

Output