HTML

LIST bulleted list (unordered) ordered list a,b,c I,j,k ii, iii, iv 1,2,3

Paragraph

Div

Span

Table

If a html element has an attribute called “style” then u are using CSS

CSS - Cascading Style Sheet

    <ol start="3">

        <li>India</li>

        <li>China</li>

        <li>Pakistan</li>

        <li>Bangladesh</li>

        <li>Sri lanka</li>

        <li>Bhutan</li>

        <li>Nepal</li>

    </ol>

Div

Is a block element

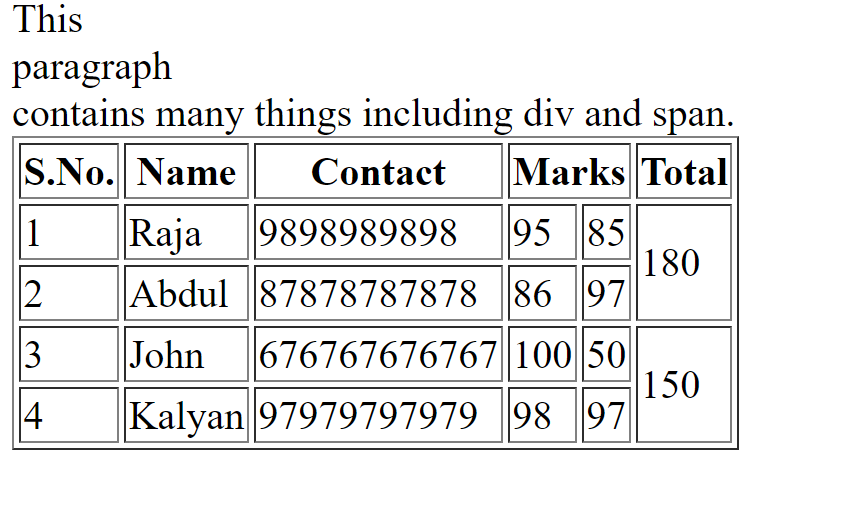
It appears in separate line

Span

Is an inline element

It appears in the same line where it is used

Inline vs block



    <table border="1">

        <thead>

            <tr>

                <th>S.No.</th>

                <th>Name</th>

                <th>Contact</th>

                <th colspan="2">Marks</th>

                <th>Total</th>

            </tr>

        </thead>

        <tbody>

            <tr>

                <td>1</td>

                <td>Raja</td>

                <td>9898989898</td>

                <td>95</td>

                <td>85</td>

                <td rowspan="2">180</td>

            </tr>

            <tr>

                <td>2</td>

                <td>Abdul</td>

                <td>87878787878</td>

                <td>86</td>

                <td>97</td>

            </tr>

            <tr>

                <td>3</td>

                <td>John</td>

                <td>676767676767</td>

                <td>100</td>

                <td>50</td>

                <td rowspan="2">150</td>

            </tr>

            <tr>

                <td>4</td>

                <td>Kalyan</td>

                <td>97979797979</td>

                <td>98</td>

                <td>97</td>

            </tr>

        </tbody>

    </table>

Regarding table:

Border

Border style

Border width (border only)

Padding

Cell spacing

Rowspan

Colspan

Hyperlink

<a href=””>display text here</a>

Hyperlink

Can be given to a page or an element in the same page etc

HTML Styles:

----------------------

Java challenge:

**Learning Platform - Requirement 6**

**Requirement 6:**  
  
In this requirement, let's find the top performer in each courses from the given user scores.  
  
Create a Class **User** with the following private attributes

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| Name | String |
| emailId | String |
| contactNumber | String |
| userScoreList | List<UserScore> |

Create / Generate appropriate Getters & Setters,  
Add a default constructor and a parameterized constructor to take in all attributes in the given order:   
**User(String name, String emailId, String contactNumber,List<UserScore> userScoreList)**  
  
The Following methods are present in the **User** class

|  |  |
| --- | --- |
| **Method Name** | **Method Description** |
| static List<User> prefill() | This method returns a List of prefilled User objects (Given in the template) |

Create a Class **Course** with the following private attributes

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| Name | String |
| Category | String |
| Price | Double |
| userScoreList | List<UserScore> |

Create / Generate appropriate Getters & Setters,  
Add a default constructor and a parameterized constructor to take in all attributes in the given order:   
**Course(String name, String category, Double price,List<UserScore> userScoreList)**  
  
The Following methods are present in the **Course** class

|  |  |
| --- | --- |
| **Method Name** | **Method Description** |
| static List<Course> prefill() | This method returns a List of prefilled Course objects (Given in the template) |

Create a Class **UserScore** with the following private attributes

|  |  |
| --- | --- |
| **Attributes** | **Datatype** |
| completionPercentage | Double |
| user | User |
| course | Course |

Create / Generate appropriate Getters & Setters,  
Add a default constructor and a parameterized constructor to take in all attributes in the given order:   
**UserScore(Double completionPercentage, User user, Course course)**  
  
The Following methods are present in the **UserScore** class

|  |  |
| --- | --- |
| **Method Name** | **Method Description** |
| static Map<String,UserScore> getTopPerformer(List<UserScore> scoreList) | This method returns a Map with Course name as Key and Top performer's UserScore object as Value in alphabetical order of the course name. |

**Note**: The Top performer of a course is an user with maximum **completionPercentage** in the course.  
          Use **TreeMap** in the getTopPerformer method.  
               Use **System.out.format("%-15s %s\n","Course","Top performer")**

**Sample Input/Output 1:**

Enter the number of user scores:

**6**

**95.2,Harry,JavaScript**

**94.3,Oliver,HTML CSS**

**86.8,James,Ethical Hacking**

**90.0,Danny,Ethical Hacking**

**79.6,Danny,PhotoShop CS**

**87.3,Oliver,JavaScript**

Course                Top performer

Ethical Hacking   Danny

HTML CSS          Oliver

JavaScript           Harry

PhotoShop CS    Danny

**Sample Input/Output 2:**

Enter the number of user scores:

**7**

**39.8,Danny,MEAN Stack**

**66.2,Danny,AWS Developer**

**96.5,Harry,Java**

**56.9,Brandon,Java**

**78.4,Brandon,Web Developer**

**93.5,Barry,Web Developer**

**56.3,Winn,MEAN Stack**

Course                 Top performer

AWS Developer   Danny

Java                     Harry

MEAN Stack       Winn

Web Developer   Barry

**//User prefill**

**public static List<User> prefill(){**

**List<User> list = new ArrayList<>();**

**list.add(new User("Harry","harry@gmail.com","9874585258",new ArrayList<UserScore>()));**

**list.add(new User("Oliver","oliver@gmail.com","9515951263",new ArrayList<UserScore>()));**

**list.add(new User("Danny","danny@gmail.com","8745874585",new ArrayList<UserScore>()));**

**list.add(new User("Matt","matt@gmail.com","9636925686",new ArrayList<UserScore>()));**

**list.add(new User("James","james@gmail.com","8454585263",new ArrayList<UserScore>()));**

**list.add(new User("Rob","rob@gmail.com","8475216953",new ArrayList<UserScore>()));**

**list.add(new User("Brandon","brandon@gmail.com","8457548965",new ArrayList<UserScore>()));**

**list.add(new User("Winn","winn@gmail.com","9231252136",new ArrayList<UserScore>()));**

**list.add(new User("Barry","barry@gmail.com","9568956235",new ArrayList<UserScore>()));**

**list.add(new User("Joe","joe@gmail.com","8525623568",new ArrayList<UserScore>()));**

**list.add(new User("John","john@gmail.com","8454575412",new ArrayList<UserScore>()));**

**list.add(new User("Slade","slade@gmail.com","9584758652",new ArrayList<UserScore>()));**

**return list;**

**}**

**//Course prefill**

**public static List<Course> prefill(){**

**List<Course> list = new ArrayList<>();**

**list.add(new Course("ANGULAR","Web Development",Double.parseDouble("9360"),new ArrayList<UserScore>()));**

**list.add(new Course("Web Developer","Web Development",Double.parseDouble("9600"),new ArrayList<UserScore>()));**

**list.add(new Course("JavaScript","Web Development",Double.parseDouble("9699"),new ArrayList<UserScore>()));**

**list.add(new Course("HTML CSS","Web Development",Double.parseDouble("9000"),new ArrayList<UserScore>()));**

**list.add(new Course("MEAN Stack","Web Development",Double.parseDouble("9599"),new ArrayList<UserScore>()));**

**list.add(new Course("C","Programming",Double.parseDouble("5000"),new ArrayList<UserScore>()));**

**list.add(new Course("Java","Programming",Double.parseDouble("4999"),new ArrayList<UserScore>()));**

**list.add(new Course("Python","Programming",Double.parseDouble("5299"),new ArrayList<UserScore>()));**

**list.add(new Course("C Sharp","Programming",Double.parseDouble("4900"),new ArrayList<UserScore>()));**

**list.add(new Course("AWS Developer","IT Software",Double.parseDouble("10000"),new ArrayList<UserScore>()));**

**list.add(new Course("Ethical Hacking","IT Software",Double.parseDouble("9300"),new ArrayList<UserScore>()));**

**list.add(new Course("PhotoShop CS","Design",Double.parseDouble("9299"),new ArrayList<UserScore>()));**

**list.add(new Course("TShirt Design","Design",Double.parseDouble("8999"),new ArrayList<UserScore>()));**

**list.add(new Course("3D Modelling","Design",Double.parseDouble("11999"),new ArrayList<UserScore>()));**

**list.add(new Course("Music Theory","Music",Double.parseDouble("9499"),new ArrayList<UserScore>()));**

**list.add(new Course("Learn Guitar","Music",Double.parseDouble("9999"),new ArrayList<UserScore>()));**

**list.add(new Course("Become a Singer","Music",Double.parseDouble("7500"),new ArrayList<UserScore>()));**

**return list;**

**}**

-------------------Solution-----------------------

import java.util.ArrayList;

import java.util.List;

public class User {

private String name;

private String emailId;

private String contactNumber;

private List<UserScore> userScoreList;

public User() {}

public User(String name, String emailId, String contactNumber, List<UserScore> userScoreList) {

super();

this.name = name;

this.emailId = emailId;

this.contactNumber = contactNumber;

this.userScoreList = userScoreList;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmailId() {

return emailId;

}

public void setEmailId(String emailId) {

this.emailId = emailId;

}

public String getContactNumber() {

return contactNumber;

}

public void setContactNumber(String contactNumber) {

this.contactNumber = contactNumber;

}

public List<UserScore> getUserScoreList() {

return userScoreList;

}

public void setUserScoreList(List<UserScore> userScoreList) {

this.userScoreList = userScoreList;

}

public static List<User> prefill(){

List<User> list = new ArrayList<>();

list.add(new User("Harry","harry@gmail.com","9874585258",new ArrayList<UserScore>()));

list.add(new User("Oliver","oliver@gmail.com","9515951263",new ArrayList<UserScore>()));

list.add(new User("Danny","danny@gmail.com","8745874585",new ArrayList<UserScore>()));

list.add(new User("Matt","matt@gmail.com","9636925686",new ArrayList<UserScore>()));

list.add(new User("James","james@gmail.com","8454585263",new ArrayList<UserScore>()));

list.add(new User("Rob","rob@gmail.com","8475216953",new ArrayList<UserScore>()));

list.add(new User("Brandon","brandon@gmail.com","8457548965",new ArrayList<UserScore>()));

list.add(new User("Winn","winn@gmail.com","9231252136",new ArrayList<UserScore>()));

list.add(new User("Barry","barry@gmail.com","9568956235",new ArrayList<UserScore>()));

list.add(new User("Joe","joe@gmail.com","8525623568",new ArrayList<UserScore>()));

list.add(new User("John","john@gmail.com","8454575412",new ArrayList<UserScore>()));

list.add(new User("Slade","slade@gmail.com","9584758652",new ArrayList<UserScore>()));

return list;

}

}

import java.util.List;

import java.util.Map;

import java.util.TreeMap;

public class UserScore {

private Double completionPercentage;

private User user;

private Course course;

public UserScore() {}

public UserScore(Double completionPercentage, User user, Course course) {

super();

this.completionPercentage = completionPercentage;

this.user = user;

this.course = course;

}

public Double getCompletionPercentage() {

return completionPercentage;

}

public void setCompletionPercentage(Double completionPercentage) {

this.completionPercentage = completionPercentage;

}

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

public Course getCourse() {

return course;

}

public void setCourse(Course course) {

this.course = course;

}

static Map<String,UserScore> getTopPerformer(List<UserScore> scoreList)

{

//under construction

Map<String, UserScore> map=new TreeMap<String, UserScore>();

for(UserScore us:scoreList)

{

Course c = us.getCourse();

String cname=c.getName();

UserScore value = map.get(cname);

if(value!=null)

{

if(us.getCompletionPercentage()>value.getCompletionPercentage())

map.put(cname, us);

}else

{

map.put(cname, us);

}

}

return map;

}

}

import java.util.ArrayList;

import java.util.List;

public class Course {

private String name;

private String category;

private Double price;

private List<UserScore> userScoreList;

public Course() {}

public Course(String name, String category, Double price, List<UserScore> userScoreList) {

super();

this.name = name;

this.category = category;

this.price = price;

this.userScoreList = userScoreList;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCategory() {

return category;

}

public void setCategory(String category) {

this.category = category;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

public List<UserScore> getUserScoreList() {

return userScoreList;

}

public void setUserScoreList(List<UserScore> userScoreList) {

this.userScoreList = userScoreList;

}

public static List<Course> prefill(){

List<Course> list = new ArrayList<>();

list.add(new Course("ANGULAR","Web Development",Double.parseDouble("9360"),new ArrayList<UserScore>()));

list.add(new Course("Web Developer","Web Development",Double.parseDouble("9600"),new ArrayList<UserScore>()));

list.add(new Course("JavaScript","Web Development",Double.parseDouble("9699"),new ArrayList<UserScore>()));

list.add(new Course("HTML CSS","Web Development",Double.parseDouble("9000"),new ArrayList<UserScore>()));

list.add(new Course("MEAN Stack","Web Development",Double.parseDouble("9599"),new ArrayList<UserScore>()));

list.add(new Course("C","Programming",Double.parseDouble("5000"),new ArrayList<UserScore>()));

list.add(new Course("Java","Programming",Double.parseDouble("4999"),new ArrayList<UserScore>()));

list.add(new Course("Python","Programming",Double.parseDouble("5299"),new ArrayList<UserScore>()));

list.add(new Course("C Sharp","Programming",Double.parseDouble("4900"),new ArrayList<UserScore>()));

list.add(new Course("AWS Developer","IT Software",Double.parseDouble("10000"),new ArrayList<UserScore>()));

list.add(new Course("Ethical Hacking","IT Software",Double.parseDouble("9300"),new ArrayList<UserScore>()));

list.add(new Course("PhotoShop CS","Design",Double.parseDouble("9299"),new ArrayList<UserScore>()));

list.add(new Course("TShirt Design","Design",Double.parseDouble("8999"),new ArrayList<UserScore>()));

list.add(new Course("3D Modelling","Design",Double.parseDouble("11999"),new ArrayList<UserScore>()));

list.add(new Course("Music Theory","Music",Double.parseDouble("9499"),new ArrayList<UserScore>()));

list.add(new Course("Learn Guitar","Music",Double.parseDouble("9999"),new ArrayList<UserScore>()));

list.add(new Course("Become a Singer","Music",Double.parseDouble("7500"),new ArrayList<UserScore>()));

return list;

}

}

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

import java.util.Map.Entry;

import java.util.Scanner;

public class App {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter the number of user scores:");

int n=sc.nextInt();

List<User> uList = User.prefill();

List<Course> cList = Course.prefill();

List<UserScore> usl=new ArrayList<>();

for(int i=0;i<n;i++)

{

// 95.2,Harry,JavaScript

String detail=sc.nextLine();

if(detail.equals(""))

detail=sc.nextLine();

String[] arr = detail.split(",");

Double completionPercentage=Double.valueOf(arr[0]);

String userName=arr[1];

User user=null;

for(User u : uList)

{

if(u.getName().equals(userName))

user=u;

}

String courseName=arr[2];

Course course=null;

for(Course c : cList)

{

if(c.getName().equals(courseName))

course=c;

}

UserScore us=new UserScore();

us.setCompletionPercentage(completionPercentage);

us.setUser(user);

us.setCourse(course);

usl.add(us);

}

Map<String, UserScore> result = UserScore.getTopPerformer(usl);

System.out.format("%-15s %s\n","Course","Top performer");

for(Entry<String, UserScore> e:result.entrySet())

{

String courseName=e.getKey();

UserScore us = e.getValue();

User user=us.getUser();

String userName=user.getName();

System.out.format("%-15s %s\n",courseName,userName);

}

}

}

HTML Colors

Colors are used in 3 different ways apart from popular color names

Popular color names are like

Red, green, blue, etc.

But apart from named colors, we can form colors by custom

Combination of Red, Green and Blue

<h1 style="background-color:rgb(255, 0, 0);">rgb(255, 0, 0)</h1>

<h1 style="background-color:rgb(0, 0, 255);">rgb(0, 0, 255)</h1>

<h1 style="background-color:rgb(60, 179, 113);">rgb(60, 179, 113)</h1>

<h1 style="background-color:rgb(238, 130, 238);">rgb(238, 130, 238)</h1>

<h1 style="background-color:rgb(255, 165, 0);">rgb(255, 165, 0)</h1>

<h1 style="background-color:rgb(106, 90, 205);">rgb(106, 90, 205)</h1>

Next, method is HEX

#FF FF FF White

#00 00 00 Black

<h1 style="background-color:#ff0000;">#ff0000</h1>

<h1 style="background-color:#0000ff;">#0000ff</h1>

<h1 style="background-color:#3cb371;">#3cb371</h1>

<h1 style="background-color:#ee82ee;">#ee82ee</h1>

<h1 style="background-color:#ffa500;">#ffa500</h1>

<h1 style="background-color:#6a5acd;">#6a5acd</h1>

HSL

<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>

<h1 style="background-color:hsl(240, 100%, 50%);">hsl(240, 100%, 50%)</h1>

<h1 style="background-color:hsl(147, 50%, 47%);">hsl(147, 50%, 47%)</h1>

<h1 style="background-color:hsl(300, 76%, 72%);">hsl(300, 76%, 72%)</h1>

<h1 style="background-color:hsl(39, 100%, 50%);">hsl(39, 100%, 50%)</h1>

<h1 style="background-color:hsl(248, 53%, 58%);">hsl(248, 53%, 58%)</h1>

Task:

Identify the various ways of using colors in HTML

Challenge:

**Twisted Array**

Subbu a very active and smart boy always thinks of doing some insane operations on numbers.One day he got an idea to form a twisted array.  
Twisted array means that first k elements of the resultant array should be exactly the same as they will be in the sorted array and the rest of the elements should go in the same order as they occur in the original array. Given an array of integers of length N and k value,write a program to help Subbu to form the twisted array.  
  
**Input format:**  
First input is an integer that denotes the N value,size of the array.  N is always greater than or equal to k.  
Second input is a series of integers separated by a space that denotes the array values.  
Third input is an integer that denotes the k value.  
  
**Output format:**  
Output is a series of integers separated by space that denotes the modified array values.  
  
**Sample Input 1:**  
5  
5 4 3 2 1  
2  
**Sample Output 1:**  
1 2 5 4 3  
  
**Explanation:**  
The sorted array is 1,2,3,4,5  
The first 2 elements in sorted array is 1,2  
The Resulting array has first two elements same as the sorted array and the remaining elements are maintained in the given order.  
So the resultant array is **[ 1,2,5,4,3 ]**  
  
**Sample Input 2:**  
7  
4 9 1 32 12 6 10  
3  
**Sample Output 2:**  
1 4 6 9 32 12 10

Solution:

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.List;

**import** java.util.Scanner;

**public** **class** App {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

**int** n=sc.nextInt();

**int** arr[]=**new** **int**[n];

**for**(**int** i=0;i<n;i++)

{

arr[i]=sc.nextInt();

}

**int** k=sc.nextInt();

List<Integer> arrCopy=**new** ArrayList<>();

**for**(**int** i=0;i<n;i++)

arrCopy.add(arr[i]);

Collections.*sort*(arrCopy);

List<Integer> result=**new** ArrayList<>();

//k elements to be added to the result

**for**(**int** i=0;i<k;i++)

{

result.add(arrCopy.get(i));

}

// System.out.println(result);

//walk through original array and add elements to result (condionally)

**for**(**int** i=0;i<n;i++)

{

//if this current element is not found in result

**if**(!result.contains(arr[i]))

result.add(arr[i]);

}

System.***out***.println(result);

}

}