11/03/21
Ex. 3

Vanathi
185001188
CSE C

JavaScript event handling mechanisms, DOM

Learning Objective -

To develop a program to generate a registration form for a hospital to register new patient details. To develop a JS program to develop a memory matching game.

Programs -

index.html :

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

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>repl.it</title>
  <link href="style.css" rel="stylesheet" type="text/css" />
</head>

<body>

  <!-- Name. Specify some font and colour. (text, onfocus, onblur, oninvalid) should only be alphabet-->

  <div class="form-container">
    <form class="reg-form" method="get" action="/table.html">
      <h1 class="form-heading">Patient Registration Form </h1>
      <table>
        <tr>
          <td style="text-align: right;"><label for="fullname">Name</label></td>
          <td><input type="text" id="fullname" name="fullname" onfocus="highlightBox()" pattern="[a-zA-Z|' ']*" onblur="undoHighlight()" oninvalid="alert('Invalid name');"></td>
        </tr>
    <tr>
     <td style="text-align: right;"><label for="address">Address</label></td>
     <td><textarea id="address" name="address" rows="5" cols="50"
onSelect="highlightText(event)"></textarea></td>
    </tr>
    <tr>
     <td style="text-align: right;"><label for="age">Age</label></td>
     <td><input type="number" id="age" name="age"></td>
    </tr>
    <tr>
     <td style="text-align: right;"><label for="dob">Date of Birth</label></td>
     <td><input type="date" id="dob" name="dob"></td>
    </tr>
    <tr>
     <td style="text-align: right;"><label for="gender">Gender</label></td>
     <td><input type="radio" id="male" name="gender" value="male">
       <label for="male" style="font-weight: 500">Male</label>
       <input type="radio" id="female" name="gender" value="female">
       <label for="female" style="font-weight: 500">Female</label>
       <input type="radio" id="other" name="gender" value="other">
       <label for="other" style="font-weight: 500">Other</label>
     </td>
    </tr>
    <tr>
     <td style="text-align: right;"><label for="married">Marital Status</label></td>
     <td><select name="married" id="married">
       <option value="single">Single</option>
       <option value="married">Married</option>
       <option value="divorced">Divorced</option>
       <option value="widowed">Widowed</option>
      </select>
     </td>
    </tr>
    <tr>
     <td style="text-align: right;"><label for="phone">Phone No.</label></td>
     <td><input type="tel" id="phone" name="phone" pattern="[0-9]{10}"
oninvalid="alert('Invalid number');"></td>
    </tr>
    <tr>
     <td style="text-align: right;"><label for="addiction">Addiction</label></td>
     <td>
      <textarea id="addiction" name="addiction" rows="5" cols="50"
ondrop="onDrop(event)"  ondragover="allowDrop(event)"></textarea>
      <div class="addiction-options">
       <div id='li1' draggable="True" ondragstart='onDragStart(event);' > Alcohol </div>
       <div id='li2' draggable="True" ondragstart='onDragStart(event);'> Smoking
</div>
       <div id='li3' draggable="True" ondragstart='onDragStart(event);'> Tobacco </div>
       <div id='li4' draggable="True" ondragstart='onDragStart(event);'> Drug </li>
      </div>

     </td>
    </tr>
   </table>
   <div style="display:flex; justify-content:center;">
    <input class="btn" type="submit" value="Submit" id="submit-btn">
    <input class="btn" type="reset" value="Reset" id="submit-btn">
   </div>

  </form>

 </div>
 <script src="dragDrop.js"></script>
 <script src="script.js">
 </script>
</body>

</html>
```

---

style.css:

```
body {
 background-color: #bfdcae;
 font-family: Helvetica, Arial, sans-serif;
 font-size: 20px;
}

.form-container {
 background-color: #f1f1e8;
 padding: 2% 6% 5%;
 margin: 5% 20%;
 border-radius: 3em;
 box-shadow: rgba(0, 0, 0, 0.15) 0px 5px 15px 0px;
}

.form-heading {
 text-align: center;
 color: #440047;
}

td {
 padding: 15px;
}

label {
 font-weight: bold;

}

input, textarea, option, select {
 font-family: Helvetica, Arial, sans-serif;
 font-size: 20px;
}

textarea {
 outline: 0;
 box-shadow: none;
 border: 3px solid #206a5d;
 border-radius: 0.5em;
 padding: 1rem;
 resize: none;
}

input, select {
 background-color: #f1f1e8;
}

input[type=text], select {
 border-top-style: hidden;
```

```
  border-right-style: hidden;
  border-left-style: hidden;
  border-bottom: 3px solid #206a5d;

 width: 250px;
}

input:focus, textarea:focus {
 outline: none;
}

table {
  margin: 0 auto;
}

.btn{
 border: 3px solid #206a5d;
 padding: 1% 2%;
 border-radius: 0.5em;
 margin: 2%;
}

.btn:hover {
 border-color: #81b214;
 background-color: white;
 cursor: pointer;
}

/* Default Table Style */
.details {
  color: #333;
  background: white;
  border: 1px solid grey;
  font-size: 12pt;
  border-collapse: collapse;
}
.details thead th,
.details tfoot th {
  color: #777;
  background: rgba(0,0,0,.1);
```

```
}
.details caption {
  padding:.5em;
}
.details th,
.details td {
  padding: .5em;
  border: 1px solid lightgrey;
}

.addiction-options {
 display: flex;
 justify-content: space-around;
}

.addiction-options div {
 background-color: white;
 padding: 3% 5%;
 border-radius: 0.5em;
 cursor: pointer;
 border: 3px solid green;
}
```

---

script.js:

```
function highlightBox() {
  document.getElementById("fullname").style.borderBottomColor = "#81b214";
}

function displayPattern() {
 alert("Invalid Name! Name can contain only alphabets and spaces...");
}

function undoHighlight() {
  document.getElementById("fullname").style.borderBottomColor = "#206a5d";
}

function highlightText(ev) {
 const selection = ev.target.value.substring(ev.target.selectionStart,
ev.target.selectionEnd);
}

document.getElementById("age").addEventListener('keydown', increment, false);

function increment() {
   var number = parseInt(age.value);
   number++;
   age.value = number;
}
```

<u>dragAndDrop.js :</u>

```
const toFill = document.getElementById('addiction');
let added = []

function allowDrop(ev) {
 ev.preventDefault();
}

function onDragStart(ev) {
 ev.dataTransfer.setData("text", ev.target.innerHTML);
}

function onDrop(ev) {
  ev.preventDefault();
  var data = ev.dataTransfer.getData("text");
 if(added.includes(data) == false)
 {
  added.push(data);
   toFill.value += data + '\n';
 }
}
```

<u>table.html :</u>

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

```
<head>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width">
 <title>repl.it</title>
 <link rel="stylesheet" href="style.css">
</head>

<body>
 <div class="table-container">
  <table id="details" class="details" width="75%">

  </table>
 </div>

 <script>
  const urlParams = new URLSearchParams(window.location.search);
  const arr = ["fullname", "address", "age", "dob", "gender", "married", "phone",
"addiction"];
  let tablecontent = "";
  for (ele in arr)
  {
   let new_row = "<tr>";
   new_row += "<td>"+arr[ele].charAt(0).toUpperCase() + arr[ele].slice(1)+"</td>";
   new_row += "<td>"+urlParams.get(arr[ele])+"</td></tr>";

   tablecontent += new_row;
  }
  document.getElementById('details').innerHTML = tablecontent;
 </script>
</body>

</html>
```

---

game.html :

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

```
<head>
 <script src="https://kit.fontawesome.com/cc7d8d5698.js"
crossorigin="anonymous"></script>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width">
 <title>Game</title>
 <link href="game.css" rel="stylesheet" type="text/css" />
</head>

<body>
 <header style="margin: 25px 0;">
  <h1 style="display:inline;"> Memory Game </h1> <i class='fa fa-2x fa-lightbulb'></i>
 </header>
 <div id="board" class="board"></div>
 <div class="game-info"><h1 id="level"></h1><h1 id="timer"></h1></div>
 <script src="game.js"></script>
</body>
```

---

game.css :

```
body {
 font-family: Tahoma, Verdana, sans-serif;
 font-weight: 900;
 text-align: center;
}

header {
 font-size: 30px;
}

.board {
 display: flex;
 flex-wrap: wrap;
 width: 636px;
 height: 636px;
 border: 5px solid #235952;
 padding: 0;
 margin: auto;
}
```

```
.card {
 background-color: #adce74;
 border: 3px solid white;
 width: 100px;
 height: 100px;
 margin: 0;
 padding: 0;

 display: flex;
 justify-content: center;
 align-items: center;
}

.flipped {
 background-color: #61b15a;
}

.found {
 background-color: white;
}
.picture {
 color: #fff76a;
}


.game-info{
 display: flex;
 margin: auto;
 justify-content: center;
}

.game-info h1 {
 margin: 1%;
}
```

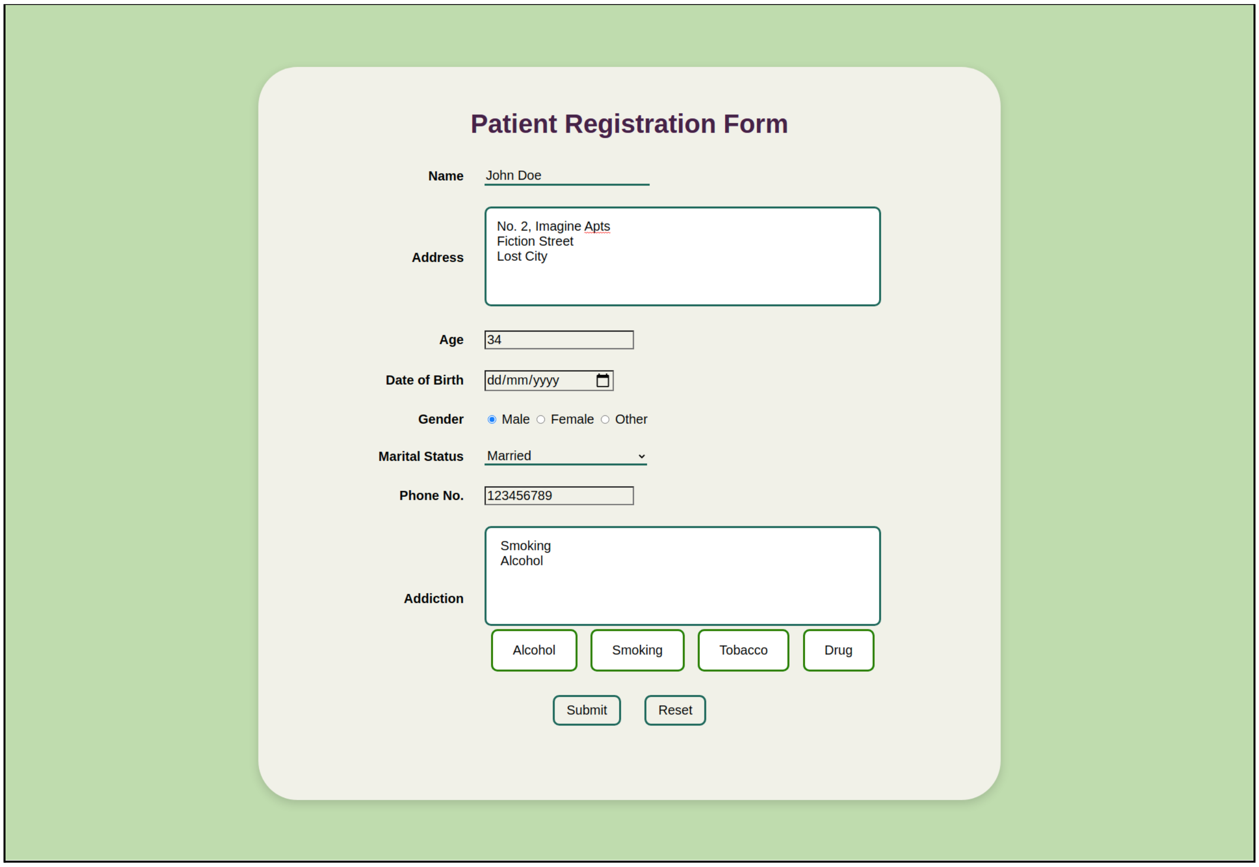
---

game.js :

```
/* initializing the board at the start*/
let no_cards = 36;
let board = document.getElementById('board');
let board_vals = [];
let score = 0;
let level = 1;
let timer = 420;

let icon_names = [
 'apple-alt', 'balance-scale', 'band-aid', 'baseball-ball', 'bicycle', 'birthday-cake',
 'bolt', 'bomb', 'bone', 'book', 'bowling-ball', 'bus',
 'dog', 'candy-cane', 'chess-king', 'chess-knight', 'clock', 'cloud'
]

let flipped = 0; // to keep track of number of cards flipped and which cards were flipped
let flipped_cards = []

for(let i=0; i<36; i++)
{
 let node = document.createElement("DIV");
 node.className = 'card';
 node.id = i;
 board.append(node);
 node.addEventListener("click", flipCard);
}

document.getElementById('level').innerHTML = "Level "+level;

resetBoard(true);

// function to reset board state
function resetBoard(restartGame){
 score = 0;
 board_vals = [];
 for(let i=0; i<no_cards/2; i++)
 {
  board_vals.push(i);
  board_vals.push(i);
 }
 for(let i=0; i<no_cards; i++)
 {
  document.getElementById(i).classList.remove('found');
  document.getElementById(i).addEventListener("click", flipCard);
 }

 shuffle(board_vals);
 if(restartGame){
  level = 1;
  timer = 480;
  document.getElementById('level').innerHTML ="Level "+level;
 }
 startTimer();
}

// when a card is flipped
function flipCard(event){
 const selected = event.target.id;
 let card = document.getElementById(selected);

 if(flipped < 2 && !card.classList.contains('flipped')){
  card.classList.add('flipped');
  let val = board_vals[selected];
  card.innerHTML='<i class="picture fas fa-4x fa-'+icon_names[val]+'"></i>';
  console.log(icon_names[val]);
  flipped += 1;
  flipped_cards.push(card);
 }
 if(flipped == 2){
  setTimeout(checkMatch, 1500);
 }
}

// to check match
function checkMatch(){
 let card1 = flipped_cards[0];
 let card2 = flipped_cards[1];
 if(board_vals[card1.id] == board_vals[card2.id]){
  card1.classList.add('found');
  card2.classList.add('found');
  card1.removeEventListener("click", flipCard);
  card2.removeEventListener("click", flipCard);
  score++;
  if(score == 18){
   alert("You win!");
   if(level < 3)
    level++;
   else
    level = 1;

   if(level == 1)
    timer = 480;
   else if(level == 2)
    timer = 360;
   else
    timer = 240;

   document.getElementById('level').innerHTML = "Level "+level;

   resetBoard(false);
  }
 }
 card1.classList.remove('flipped');
 card1.innerHTML="";
 card2.classList.remove('flipped');
 card2.innerHTML="";

 flipped = 0;
 flipped_cards = [];
}

// To shuffle the board
function shuffle(array) {
  var currentIndex = array.length, temporaryValue, randomIndex;
  while (0 !== currentIndex) {
    randomIndex = Math.floor(Math.random() * currentIndex);
    currentIndex -= 1;
    temporaryValue = array[currentIndex];
    array[currentIndex] = array[randomIndex];
    array[randomIndex] = temporaryValue;
  }
  return array;
```

```
}

// To run the timer
function startTimer(){
 const x = setInterval(function() {
  let minutes = Math.floor(timer / 60).toLocaleString('en-US', {minimumIntegerDigits: 2, useGrouping:false});
  let seconds = (timer - minutes * 60).toLocaleString('en-US', {minimumIntegerDigits: 2, useGrouping:false});

  // Display the result in the element with id="demo"
  document.getElementById("timer").innerHTML = minutes + ":" + seconds + "s ";

  // If the count down is finished, write some text
  if (timer <= 0) {
   clearInterval(x);
   console.log("hello");
   document.getElementById("timer").innerHTML = "Time's Up!";
   resetBoard(true);

  }
  timer--;
 }, 1000);

}
```

Outputs -



Patient Registration Form

Name: John Doe

Address: No. 2, Imagine Apts
Fiction Street
Lost City

Age: 34

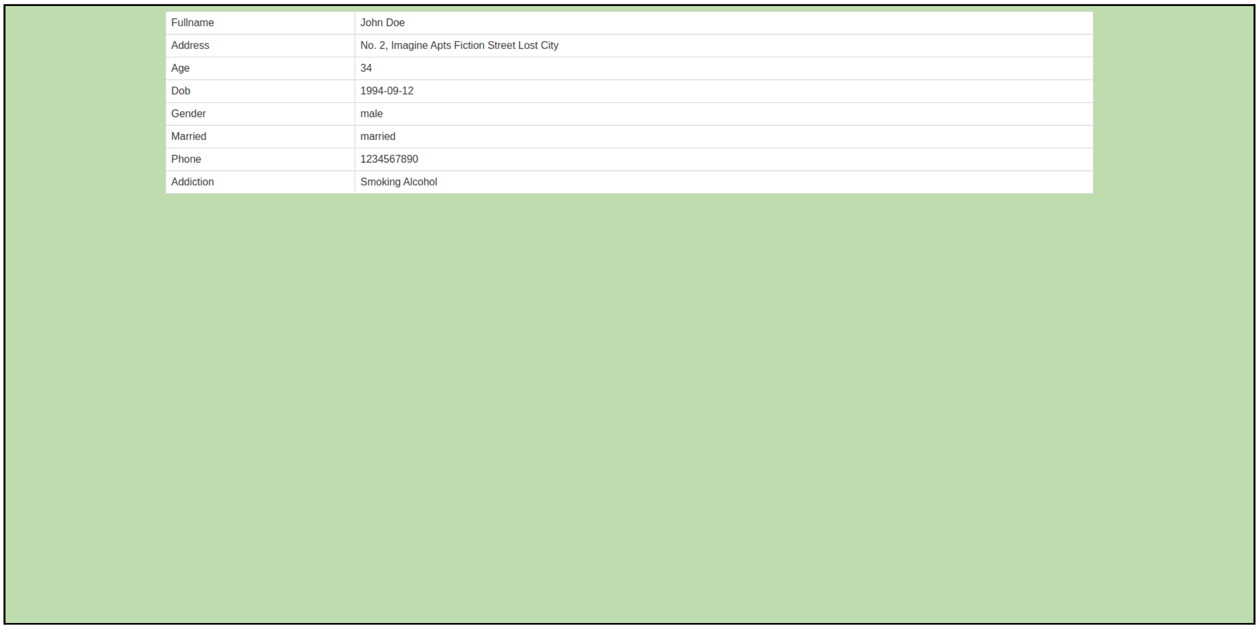
Date of Birth: dd/mm/yyyy

Gender: ● Male ○ Female ○ Other

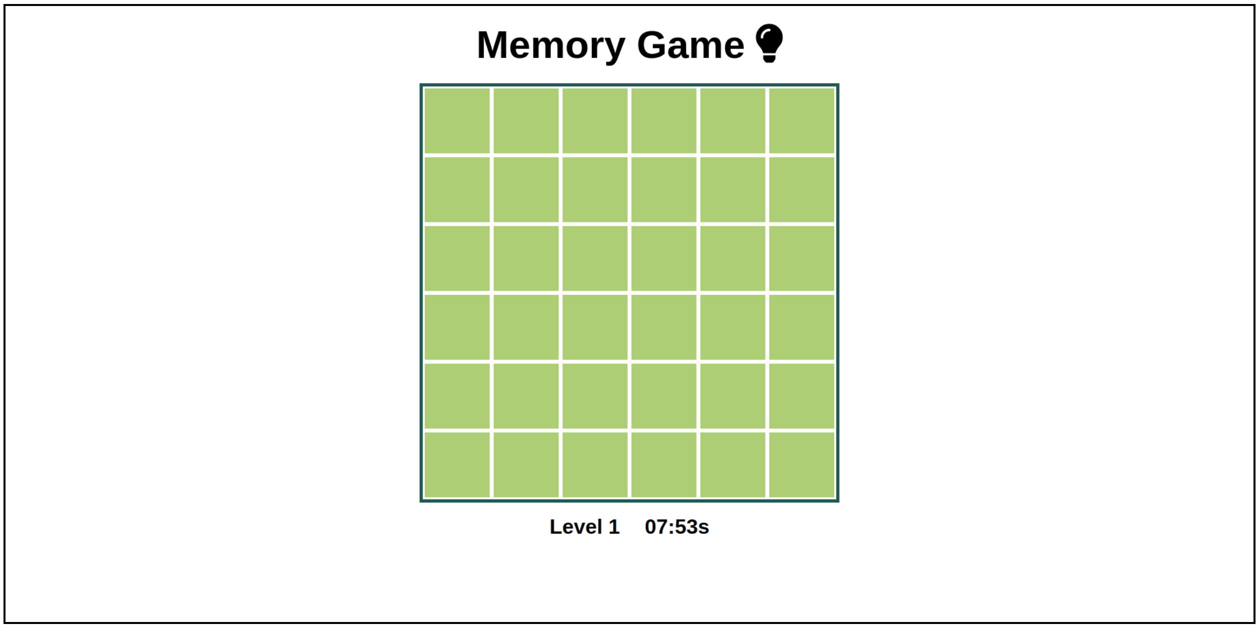
Marital Status: Married

Phone No.: 123456789

Addiction: Smoking
Alcohol

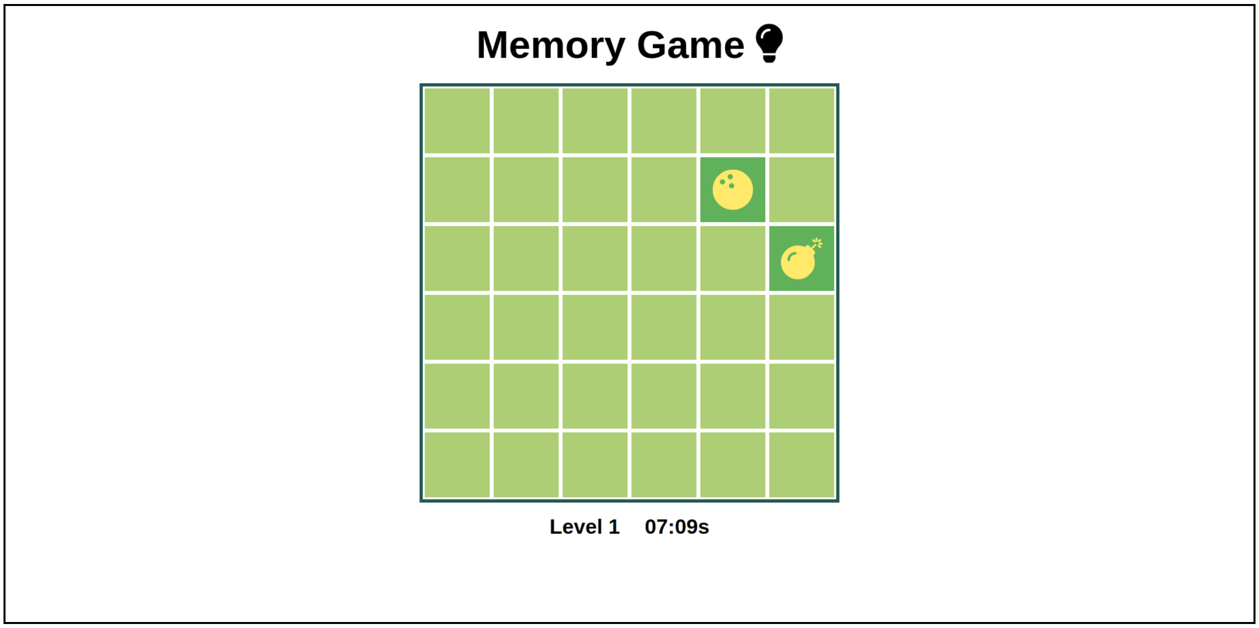


Fullname	John Doe
Address	No. 2, Imagine Apts Fiction Street Lost City
Age	34
Dob	1994-09-12
Gender	male
Married	married
Phone	1234567890
Addiction	Smoking Alcohol



Memory Game

Level 1 07:53s



Memory Game

Level 1 07:09s

Learning Outcomes -

1. I successfully designed a website for a registration form for a hospital to register new patient details.
2. I successfully designed a memory game using JS
3. I learnt how to implement various concepts like timers, drag and drop elements and form validation using javascript
4. I understood event handling and the DOM.
5. I learnt a method to pass form values from one page to another HTML page
6. Overall, it was a good learning experience about various JS and HTML concepts.