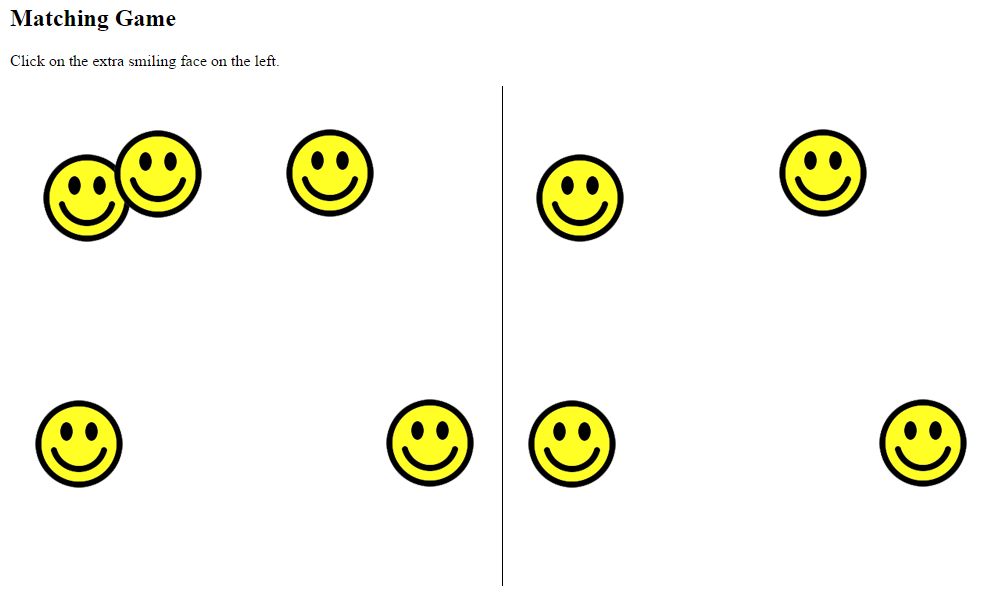
Instructions

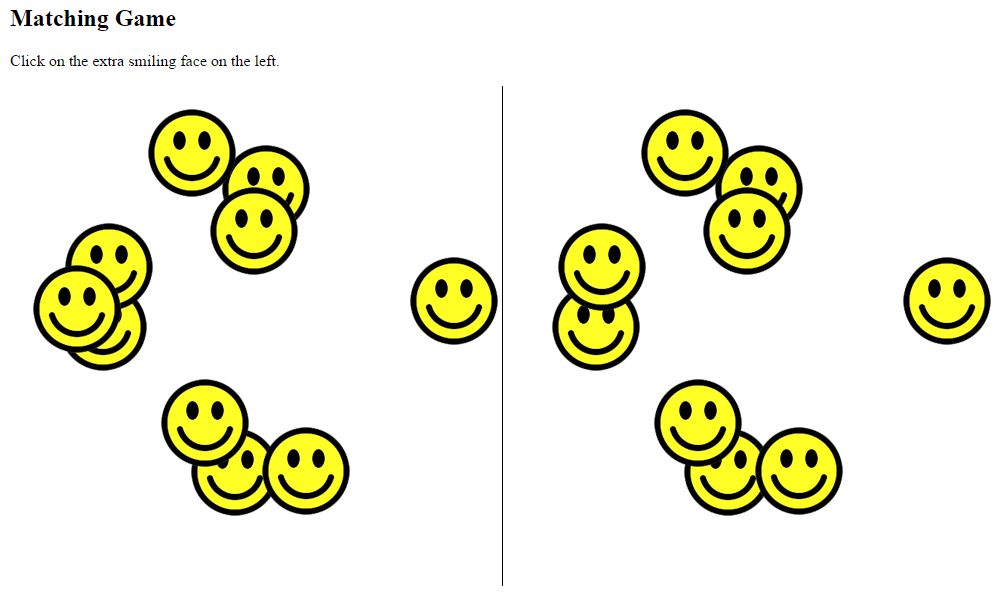
**Assignment Overview**

For this workshop assignment, you will write a matching game. Please see the accompanying video for a walk-through of gameplay.   
  
Some of the JavaScript skills you will need for this assignment are demonstrated in the Patterns exercise earlier this week in the DOM section. Make sure that you have completed that exercise and understand the code in it, especially the code that is related to DOM manipulation, before you begin this assignment.  
  
When the game starts, five smiley faces are shown on the left and four are shown on the right. This is illustrated below.

  
The left and right sides are identical, except for one thing: the left side has one extra face. The user needs to click on that extra face. If anything except the correct face is clicked, a message is displayed saying that the game is over. If the correct face is clicked, all the currently displayed faces are removed and a new set of faces is shown at random positions.

Each time a new set of faces is shown there will be 5 more faces than before, on both the left and the right sides. There will always be one more face shown on the left than on the right. The other faces on the right and left will be identical in position to each other.

For example, let’s imagine you are playing the game shown in the previous figure. After clicking on the extra face (top middle) all the faces disappear and the following new set of faces are shown, at new random positions. As you can see, on both sides 5 more faces than before are shown



After playing the game by correctly clicking on the extra face many times, a lot of faces will be shown. This is illustrated below.



**Which Browser to Use**

This project was developed using the Chrome browser and has been tested to work on Firefox. However, to avoid any potential trouble with inconsistencies between browsers, it may be wise for you to use Chrome.

**Assignment Tasks**

To help you organize your approach, detailed instructions are provided below in the form of 4 main tasks, each separated into smaller parts.

**Summary of Task 1:** Set up the HTML and CSS.

**Summary of Task 2:**Write the JavaScript for the left side of the game.

**Summary of Task 3:**Write the JavaScript for the right side of the game.

**Summary of Task 4:**Finish the game logic.

Add the smiley image to your **images** folder.

**https://learn.nucamp.co/pluginfile.php/619/mod\_assign/intro/smile.png**

**TASK 1: Set up the HTML and CSS**

For this task, you will create the HTML page without any JavaScript. The result will look like the below screenshot when viewed in a browser. As you can see, only the instructions and the middle line are visible.

|  |  |
| --- | --- |
| **Part 1: Set up the HTML**   * Create a basic valid HTML5 document as you learned to do in Week 1, and give it the name **matching-game.html**. Be sure to set the **DOCTYPE**, the **meta charset**, and an appropriate **title**(such as 'Matching Game'). * In the **body** element, add an **h1** element with the text content of **Matching Game**.  *Beneath this, add a****button****element with the****type****of****"button"****, an****onclick****attribute with the value of****"runGame()"****, and text content of****"Start Game"****(without the quotes).* * *Text with the instructions e.g. ‘Click on the extra smiling face on the left.’*   **Part 2: Add left and right divs (and add the <script> tags)**   * You will also need 2 divs (id="leftSide", and id="rightSide")  *and 1 container ‘main’ (id="x") element if you use the button*   *You might just want to add one image as a test…* <div id="leftSide"><img src="images/smile.png" style="top: 100px; left: 100px;"></div> | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport"   content="width=device-width, initial-scale=1.0">      <title>Matching Game</title>  </head>  <body>      <h1>Matching Game</h1>      <button onclick="generateFaces()">Start Game</button>      <p>Click on the extra smiling face on the left.</p>      <main id="x">          <div id="leftSide">*add image*</div>          <div id="rightSide"></div>      </main>  <script>  //PART 2 AND 3 GO HERE  </script>  </body>  </html> |
| **Part 3: Add CSS**   * Use an internal stylesheet to add the following styles.   1. For all **img** elements:   position: absolute;  This is so that we can fix the exact position of any image later. Although we haven’t actually added any images to the webpage at this stage, we are adding this style rule now so that we can easily control their exact position later.   * 1. For all **div** elements:   position: absolute;  width: 500px;  height: 500px;  This will set both the left and right divs to be 500px square each, with absolute positioning.   * 1. For **only** the div with the id of **rightSide**(recall how to use an ID selector):   **left: 500px;**  **border-left: 1px solid;**  This moves the rightSide div 500 pixels to the right, so that it is to the right of the leftSide div. Then it uses the **border-left** property to create a vertical line between the two divs. | <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport"   content="width=device-width, initial-scale=1.0">      <title>Matching Game</title>  <style>      img {position: absolute;}      div {          position: absolute;          width: 500px;          height: 500px;      }      #rightSide {          left: 500px;          border-left: 1px solid;      }  </style>  </head> |

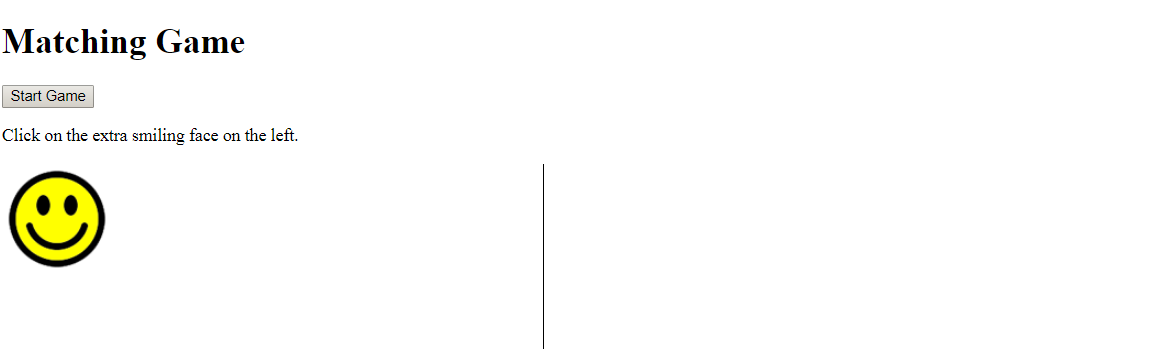
The result will look like this when viewed in a browser. As you can see, only the instructions and the middle line are visible. ***After viewing… remove the image.***

**CSS position**   
static: Default value. Elements render in order, as they appear in the document flow

absolute: The element is positioned relative to its first positioned (not static) ancestor element

fixed: The element is positioned relative to the browser window

relative: The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position



**TASK 2: Generate the left side images - Alternate Approach.**

For Part 2: [Generating the left side images]

* For this task you need to add **JavaScript** code which generates (a variable called) *numberOfFaces* images on the left side. (There are no right side faces in part 2).
* The result will look like this when viewed in a browser.

NOTE: YOU NEED AN ONLOAD/ONCLICCK EVENT TO MAKE THIS WORK: onload="generateFaces()"> or onclick="generateFaces()"

let numberOfFaces = 5;

const theLeftSide = **document.getElementById**("leftSide");

const theRightSide = document.getElementById('rightSide');

// *theLeftSide var now controls all properties/methods of the div element with the id “leftSide”*

function generateFaces() {

for (i = 1; i <= numberOfFaces; i++) { // *standard for loop*

let randomTop = i \* 100;

let randomLeft = i \* 100;

const face = document.createElement("img");

face.src = "smile.png";

face.style.top = randomTop + "px";

face.style.left = randomLeft + "px";

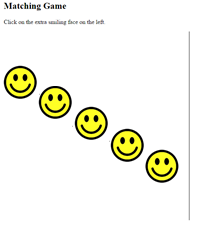
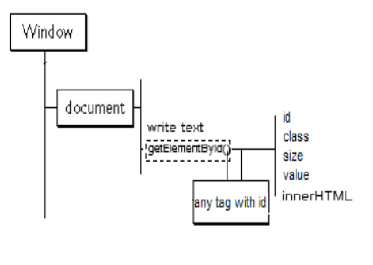
theLeftSide.appendChild(face);

}

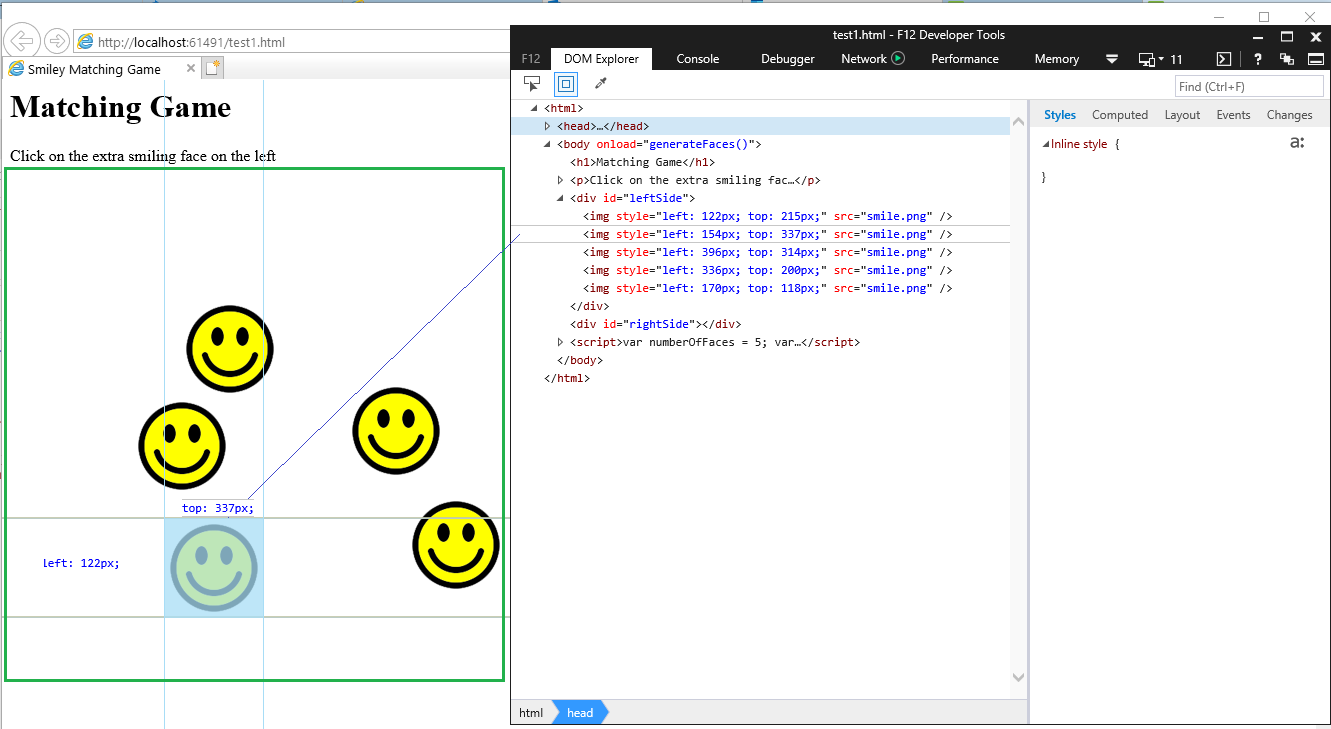
}

Think of the variable **thisSmiley** as a dynamic way to create the html tag:

<img src="smile.png" style="top: 100px; left: 100px;">

  
Now change “i” to a random number.

|  |
| --- |
| Remember from last week: Random Numbers  Math.random();     // returns a random number from 0 to .99999999999  Math.floor():      // rounds a number down to the lowest whole number  Math.floor(Math.random() \* 400); // returns a random integer from 0 to 399  **Change the code** for randomTop, and randomLeft to use a random number from 0 to 399  let randomTop = i \* 100;  let randomLeft = i \* 100;  This is the updated code:  let randomTop = Math.floor(Math.random() \* 400);  let randomLeft = Math.floor(Math.random() \* 400); |

Run the code (as is), inspect the source, and you will see something like the following. *Mouse over the images…*

A couple of DOM methods -

The **appendChild()** method is used to create a node as the last child of the node (used for creating a new element). theLeftSide.appendChild(thisSmiley);

The **cloneNode()** method creates a copy of a node, and returns that clone.

cloneNode(true/~~false~~) uses only 1 parameter: true

true - Clone the node, its attributes, and its descendants

*A couple more DOM methods -*The **createElement** () method is used to create any HTML element. Then you add attributes and style to it…   
var face = document.createElement('img');

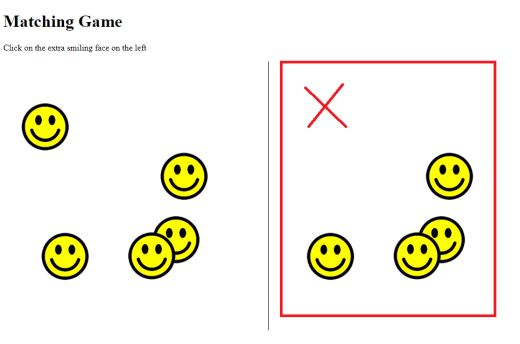
The **removeChild()** method removes a specified child node of the specified element..

removeChild(*node*) has 1 parameter: fistChild, lastChild, etc…

#### TASK 2: Generate the left side images - Instructors Approach

|  |  |
| --- | --- |
| **Part 1: Global variables**  Declare **3** global variables (might as well add the 3rd one now…)  1) one let for **numberOfFaces** and assign it the numeric value of 5 2) one const for **theLeftSide**, and assign it the contents of that div 3) one const for theRightSide, and assign it the contents of that div   * Declare a function named **generateFaces** with an empty parameter list. * In this function, the faces are generated using a **for** a loop. Set up a for loop to iterate **numberOfFaces**times. In each iteration:   + Create an **img** node (face) using **document.createElement()**with the appropriate argument. At the end of this operation, the variable **face**should point to a newly created **img** node (that has not yet been attached to the DOM. Declare a variable named **face**and for its value).   + Set the image's **src**property as follows, making sure that the **smile.png** file is truly in your images folder (download link is just below the screenshot above):  face.src = 'images/smile.png'; The position of the **face** node is controlled by the **top**and **left**style properties. You will next generate random numbers to use as the values for these properties.   + Declare a variable named **randomTop**. Use**Math.floor()**and **Math.random()**to generate a random number between **1-400**, and set this number as the value of randomTop.   + Declare another variable named **randomLeft**. Again, generate a random number in the 1-400 range and set it as the value of randomLeft.   + Set the value of **face.style.top** to randomTop, making sure to add **'px'** to the end, e.g.: face.style.top = randomTop + 'px';   + Similarly, set the value of **face.style.left** to **randomLeft**, making sure to add **'px'**.   + Remember in **Part 1**, you created a global variable named **theLeftSide**. Use **appendChild()** to add the newly created **face** image element node as a child of **theLeftSide.** * The first time that the **generateFaces** function is run, the **for** loop inside it should iterate 5 times, causing 5 smiley face images to appear at random locations on the left side.   **Add 3 more variables**  1) one for number of faces and assign it the numeric value of 5 2) one for the left side, and assign it the contents of that div 3) one for the right side, and assign it the contents of that div  These two event listeners are explained on p.8  don’t do them yet… | <script>  // window.addEventListener('load', generateFaces);  let numberOfFaces = 5;  const theLeftSide = document.getElementById('leftSide');  const theRightSide = document.getElementById('rightSide');  //EVERYTHING BELOW IS COVERED IN THE ALTERNATE APPROACH  function generateFaces() {      for (let i = 0; i < numberOfFaces; i++) {    const face = document.createElement('img');    face.src = 'images/smile.png';    const randomTop = Math.floor(Math.random() \* 400) + 1;          const randomLeft = Math.floor(Math.random() \* 400) + 1;      face.style.top = randomTop + 'px';          face.style.left = randomLeft + 'px';          theLeftSide.appendChild(face);      }  // Start here…      const leftSideImages = theLeftSide.cloneNode(true);      leftSideImages.removeChild(leftSideImages.lastChild);      theRightSide.appendChild(leftSideImages);      theLeftSide.lastChild.addEventListener('click', nextLevel);      document.getElementById("x").addEventListener('click', gameOver);  } |

|  |  |
| --- | --- |
| **Part 3: Load the Function**   * To call the **generateFaces**function when the webpage is loaded, use one or the other of the below options, not both. * ~~Add an inline~~**~~onload~~**~~event handler to the~~***~~body~~***~~element~~:   This gives you a second way to load your game…   * **REM:** You already have an **onclick** event (task 1.part 1)   <body onload="generateFaces()" or   <button onclick="generateFaces()"  **At this point, you should be able to view your webpage in your browser and see that it looks like the screenshot shown above for Task 2.** | <body>        <h1>Matching Game</h1>      <button onclick="generateFaces()">Start Game</button>      <p>Click on the extra smiling face on the left.</p>      <div id="x">          <div id="leftSide"></div>          <div id="rightSide"></div>      </div>      <script>          // window.addEventListener('load', generateFaces);          let numberOfFaces = 5;          const theLeftSide = document.getElementById('leftSide');          const theRightSide = document.getElementById('rightSide');          function generateFaces() {              for (let i = 0; i < numberOfFaces; i++) {                  const face = document.createElement('img');                  face.src = 'images/smile.png';                  const randomTop = Math.floor(Math.random() \* 400) + 1;                  const randomLeft = Math.floor(Math.random() \* 400) + 1;                  face.style.top = randomTop + 'px';                  face.style.left = randomLeft + 'px';                  theLeftSide.appendChild(face);              }              const leftSideImages = theLeftSide.cloneNode(true);              leftSideImages.removeChild(leftSideImages.lastChild);              theRightSide.appendChild(leftSideImages);              theLeftSide.lastChild.addEventListener('click', nextLevel);              document.getElementById("x").addEventListener('click', gameOver);          }      </script>  </body> |

**TASK 3: Handle the right side**

You will now extend the JavaScript you developed in Task 2 to generate images on the right side. At the end of this task, your assignment should look similar to this in the browser, with smiley faces on both sides, one less on the right side:

|  |  |
| --- | --- |
| **Part 1: Global variable**   * Declare a new global variable named **theRightSide**, below where you declared the variable **theLeftSide.**Point this variable to the div with the ID of **rightSide**. **ALREADY DONE**   **Part 2: Clone images to the right side**   * Add the following code in the **generateFaces** function, below the for loop you created in Task 2:   + Declare a variable called **leftSideImages**or similar. For its value, clone the entire **theLeftSide**div node, including all its children, i.e.:   const leftSideImages = theLeftSide.cloneNode(true);   * + After this, use **removeChild()** to remove the last child of **leftSideImages**. This will ensure that there is one less smiley face on the right side.   + Use **appendChild()** to append **leftSideImages** as a child of **theRightSide**. * At this point, the DOM will look like this: http://home.cse.ust.hk/~rossiter/mooc/matching_game/dom_structure.png | let numberOfFaces = 5;          const theLeftSide = document.getElementById('leftSide');          const theRightSide = document.getElementById('rightSide');          function generateFaces() {              for (let i = 0; i < numberOfFaces; i++) {                  const face = document.createElement('img');                  face.src = 'images/smile.png';                  const randomTop = Math.floor(Math.random() \* 400) + 1;                  const randomLeft = Math.floor(Math.random() \* 400) + 1;                  face.style.top = randomTop + 'px';                  face.style.left = randomLeft + 'px';                  theLeftSide.**appendChild**(face);              }              const leftSideImages = theLeftSide.**cloneNode**(true);              leftSideImages.**removeChild**(leftSideImages.lastChild);              theRightSide.**appendChild**(leftSideImages); |

* The **cloneNode()** method creates a copy of a node, and returns that clone.  
  cloneNode(true/~~false~~) uses only 1 parameter: true (false wouldn’t do anything)  
   true - Clone the node, its attributes, and its descendants

Test your code

#### TASK 4: Finish the game logic

In this task, you will complete the JavaScript code developed in Tasks 2 and 3 to include adding and removing event handlers and other game logic.

|  |  |
| --- | --- |
| **Part 1: Add an event handler function to the extra face using addEventListener**  The player is supposed to click on the extra smiley face on the left side that is ***not***on the right side. Remember, the faces on the right side are a clone of the left side, except for the last child which was been removed on the **theRightSide** node.   That means the extra smiley face on the left side will be the last child of the **theLeftSide**node. You will need to add an event handler to this face so that when it is clicked, the program will go to the next level.   * Inside the **generateFaces** function, below the existing code, set up an event listener for the **last child** of **theLeftSide**with **addEventListener**. * Have **addEventListener()** listen for the event '**click**' as the first argument. For the second argument, use the function name **nextLevel**(or something similar). * Outside of the function **generateFaces**, declare a new function named **nextLevel**(or whatever name you used for the second argument in the event listener you just created.) * The starting code for this function is provided below:   function nextLevel(event) {  event.stopPropagation();  numberOfFaces += 5;  generateFaces();  }   * Add a line event.stopPropagation();   This is necessary in order to ensure that the event does not also get applied to other elements in the web page, such as other faces. That would trigger the function multiple times, which is not what we want.  Add a while loop to clear out the original images on both sides… Otherwise you will just add additional images on top of what is already there…  You can use the removeChild method to do this, looping until there are no more children  **OR**  With two lines of code you can delete all images from both sides              theLeftSide.innerHTML = '';              theRightSide.innerHTML = '';   * Add a line numberOfFaces += 5;   increases the number stored in numberOfFaces by 5,  so that the next time the faces are generated there are 5 more than before on both sides   * Add a line generateFaces(); Basically, this restarts the game,   And means that a new set of faces is generated. Because of the increase in value of **numberOfFaces**,there will be 5 more faces than before on both sides. | The **addEventListener(*event, function*)** method attaches an event handler to a specified element. It has 2 required parameters:  **event** - Required. A String that specifies the name of the event. (i.e., *click*, mouseover, keydown, mouseout)  **function** - Required. Specifies a function to run when the event occurs.         theLeftSide.**lastChild**.**addEventListener**('click', **nextLevel**);         document.getElementById("x").addEventListener('click', gameOver);  // the gameOver() function is included at the end          } //End of generateFaces function  function nextLevel(event) {      event.**stopPropagation**();      while (theLeftSide.firstChild) {          theLeftSide.removeChild(theLeftSide.firstChild);      }      while (theRightSide.firstChild) {         theRightSide.removeChild(theRightSide.firstChild);      }      numberOfFaces += 5;      generateFaces();  } //End of nextLevel function |

**The stopPropagation() method prevents propagation of the same event from being called. It prevents further spread of the current event.**

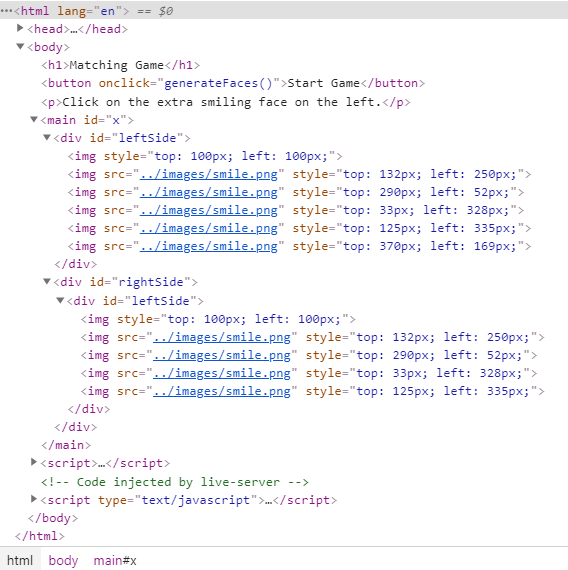
**See Below**

**Event** **Propagation**

**Event** **propagation** is a blanket term for both **event** *bubbling* and **event** *capturing*.

Consider the page we have built so far…

We have images inside of divs, inside of other divs, not to mention the body and html…



Our DOM Model (propagation path) looks something like this…

*window*,

document,

HTML,

BODY,

DIV, (main)

DIV, (rightSide)

DIV, (leftSide)

IMG

A click on an image does not only generate a click event for the corresponding IMG element, but also for the parent DIV, for the grandfather DIV and so on...

In our case we really only care about the IMGs in the in the leftSide, DIVs… But the propagation is still across all the nodes in the propagation path.

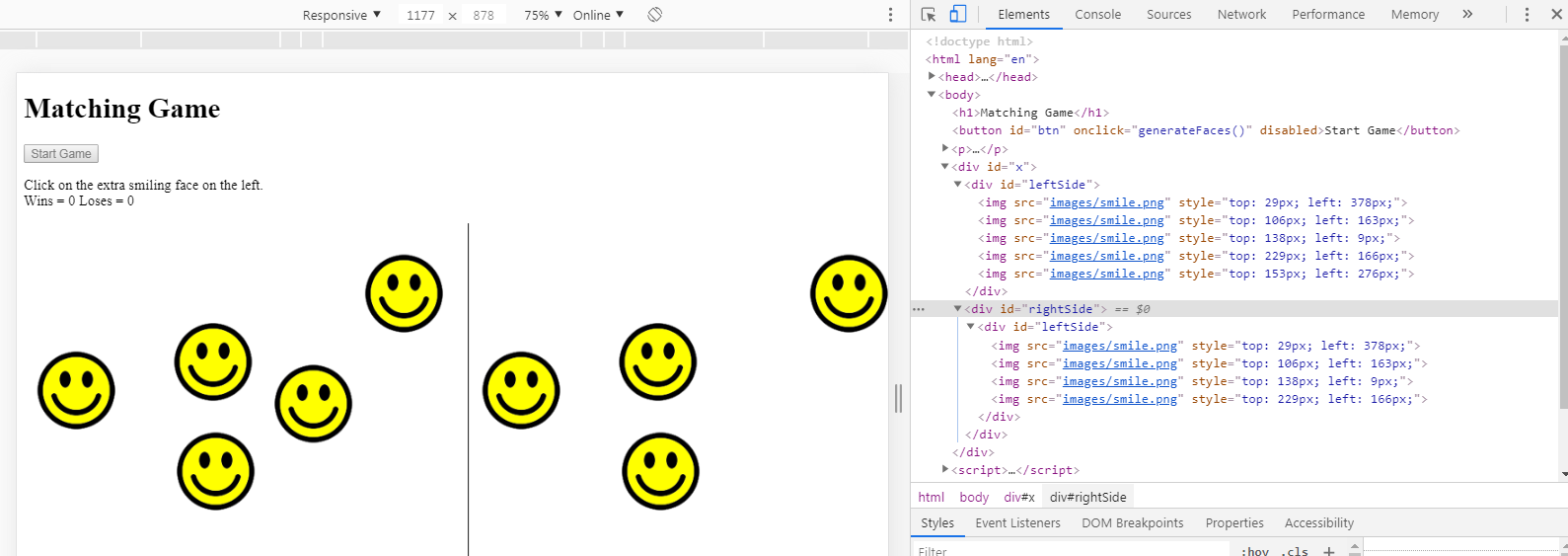
So, the event propagation can be stopped in any event listener by invoking the **stopPropagation** method of the event object. This means that all the listeners registered on the nodes on the propagation path that follow the current target will not be called.

**Notes**:

With **bubbling**, the event is first captured and handled by the innermost element and then propagated to **outer** elements.

With **capturing**, the event is first captured by the outermost element and propagated to the **inner** elements.

|  |  |
| --- | --- |
| **Part 2: Add an event handler function to the body using addEventListener()** ALREADY DONE   * You need to add another function for handling the situation when the player clicks on anything except the correct face. * In the **generateFaces** function, below where you added the last event listener, use **addEventListener()**one more time. You will use it on **document.body  (or document.getElementById("x")** if using a button**)** with the first argument **'click'**and the second argument of **gameOver**. * Outside of the **generateFaces** function, declare a third function named **gameOver**with an empty parameter list. * Inside this function, write an **alert()** with the message of 'Game Over!'.   **Part 3: Remove event handler functions using removeEventListener()**   * Inside the **gameOver** function, use **removeEventListener** twice to remove the event handlers you created with **addEventListener**. That means ~~you need to use it on~~**~~document.body~~**  (or **document.getElementById("x")** if using a button)  to remove the **gameOver** event handler function for the**'click'** event, and on **theLeftSide.lastChild**to remove the **nextLevel** event handler function for the **'click'** event.   **Part 4: Delete the child nodes**   * Each time the player clicks on the correct face, all faces must be removed before a new set of faces is generated and added to the page. So that means at the appropriate place, **all children of both theLeftSide and theRightSide div nodes need to be removed**. In the past week, you learned how to remove all child nodes of a parent node using a **while** loop. Use that knowledge to add **two while loops** to the **nextLevel** function to remove all child nodes of **theLeftSide**and **theRightSide.** Be sure to place these loops **before** the call to **generateFaces()**in the **nextLevel**function. * Don’t forget to reset the number of images back to 5. | theLeftSide.lastChild.addEventListener('click', nextLevel);    document.getElementById("x").addEventListener('click', gameOver);  function gameOver() {      alert('You Lose!');    document.getElementById("x").**removeEventListener**('click', gameOver);    theLeftSide.lastChild.**removeEventListener**('click', nextLevel)  theLeftSide.innerHTML = ''; //Alternative to removeChild loop      theRightSide.innerHTML = '';  while (theLeftSide.firstChild) {  theLeftSide.removeChild(theLeftSide.firstChild);  }  while (theRightSide.firstChild) {  theRightSide.removeChild(theRightSide.firstChild);  }      numberOfFaces = 5;  } |

****

|  |  |
| --- | --- |
| BONUS CHALLENGES When the game ends, show a button that will allow the player to restart the game. | <button id="btn" onclick="generateFaces()">Start Game</button>  function generateFaces() {           document.getElementById("btn").disabled = true;  function gameOver() {  document.getElementById("btn").disabled = false; |
| Add a counter that shows how many attempts the player has made, and show them the count at the end of the game. | <p>Click on the extra smiling face on the left.<br>      Wins = <span id="win">0</span> Loses = <span id="los">0</span>   </p          let wins = 0; //Global Variable          let loss = 0; //Global Variable  function nextLevel(event) {              wins = wins + 1;              document.getElementById("win").innerHTML = wins;  function gameOver() {              loss = loss + 1;              document.getElementById("los").innerHTML = loss; |
| Allow the user to choose how many additional faces are generated for each level, i.e. allow them to set the numberOfFaces variable (through a prompt or similar) at the beginning of the game.  Alternatively, allow the user to choose a mode at the beginning of the game (e.g. Easy, Normal, Difficult) then set the numberOfFaces accordingly (e.g. 2 for easy, 5 for normal, 8 for difficult...) | <p>Click on the extra smiling face on the left.<br>      Wins = <span id="win">0</span> Loses = <span id="los">0</span>      <br>How many faces do you want to generate:   <input id="numFace" type="number" value="1" style="width: 40px">      </p>  =-=-=-=-=-=-=-=-=-=-=-=-=-=-  <script>      let wins = 0; //Global Variable      let loss = 0; //Global Variable      let numberOfFaces; // = 5;      let numFace;  // static      let newStart = true; // check for new game      const theLeftSide = document.getElementById("leftSide");      const theRightSide = document.getElementById('rightSide');  =-=-=-=-=-=-=-=-=-=-=-=-=-=-      function generateFaces() {          if(newStart == true){            numberOfFaces = document.getElementById('numFace').value;            numFace = document.getElementById('numFace').value;          }  =-=-=-=-=-=-=-=-=-=-=-=-=-=-      function nextLevel(event) {          wins = wins + 1;          document.getElementById("win").innerHTML = wins;          newStart = false;          event.stopPropagation();          while (theLeftSide.firstChild) {              theLeftSide.removeChild(theLeftSide.firstChild);          }          theRightSide.innerHTML = ''; //alturnative to remove child loop          numberOfFaces = Number(numberOfFaces) + Number(numFace);  //5;          generateFaces();      } //End of nextLevel function  =-=-=-=-=-=-=-=-=-=-=-=-=-=-      function gameOver() {          loss = loss + 1;          document.getElementById("los").innerHTML = loss;          alert('You Lose!');          document.getElementById("btn").disabled = false;          document.getElementById("x").removeEventListener('click', gameOver);          theLeftSide.lastChild.removeEventListener('click', nextLevel)          while (theLeftSide.firstChild) {              theLeftSide.removeChild(theLeftSide.firstChild);          }          theRightSide.innerHTML = '';          numberOfFaces = document.getElementById('numFace').value;  //5;          newStart = true;      } |

**Final Code (next page)**

**Base Code**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport"

        content="width=device-width, initial-scale=1.0">

    <title>Matching Game</title>

    <style>

        img {position: absolute;}

        div {

            position: absolute;

            width: 500px;

            height: 500px;

        }

        #rightSide {

            left: 500px;

            border-left: 1px solid;

        }

    </style>

</head>

<body>

  <h1>Matching Game</h1>

  <button onclick="generateFaces()">Start Game</button>

  <p>Click on the extra smiling face on the left.</p>

  <main id="x">

      <div id="leftSide"><img style="top: 100px; left: 100px;"></div>

      <div id="rightSide"></div>

  </main>

<script>

    let numberOfFaces = 5;

    const theLeftSide = document.getElementById("leftSide");

    const theRightSide = document.getElementById('rightSide');

    // theLeftSide var now controls all properties/methods of the div element with the id “leftSide”

    function generateFaces() {

        for (i = 1; i <= numberOfFaces; i++) {  // standard for loop

            let randomTop = Math.floor(Math.random() \* 400);

            let randomLeft = Math.floor(Math.random() \* 400);

            const face = document.createElement("img");

                face.src = "../images/smile.png";

                face.style.top = randomTop + "px";

                face.style.left = randomLeft + "px";

            theLeftSide.appendChild(face);

        }

        const leftSideImages = theLeftSide.cloneNode(true);

        leftSideImages.removeChild(leftSideImages.lastChild);

        theRightSide.appendChild(leftSideImages);

        theLeftSide.lastChild.addEventListener('click', nextLevel);

        document.getElementById("x").addEventListener('click', gameOver);

    }

    function nextLevel(event) {

        event.stopPropagation();

        while (theLeftSide.firstChild) {

            theLeftSide.removeChild(theLeftSide.firstChild);

        }

        theRightSide.innerHTML = ''; //alturnative to remove child loop

        numberOfFaces += 5;

        generateFaces();

    } //End of nextLevel function

    function gameOver() {

        alert('You Lose!');

        document.getElementById("x").removeEventListener('click', gameOver);

        theLeftSide.lastChild.removeEventListener('click', nextLevel)

        while (theLeftSide.firstChild) {

            theLeftSide.removeChild(theLeftSide.firstChild);

        }

        theRightSide.innerHTML = '';

        numberOfFaces = 5;

    }

</script>

</body>

</html>

**Bonus Code**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport"

        content="width=device-width, initial-scale=1.0">

    <title>Matching Game</title>

    <style>

        img {position: absolute;}

        div {

            position: absolute;

            width: 500px;

            height: 500px;

        }

        #rightSide {

            left: 500px;

            border-left: 1px solid;

        }

    </style>

</head>

<body>

  <h1>Matching Game</h1>

  <button onclick="generateFaces()" id="btn">Start Game</button>

  <p>Click on the extra smiling face on the left.<br>

      Wins = <span id="win">0</span> Loses = <span id="los">0</span>

      <br>How many faces do you want to generate:

      <input id="numFace" type="number" value="1" style="width: 40px">

  </p>

  <main id="x">

      <div id="leftSide"><img style="top: 100px; left: 100px;"></div>

      <div id="rightSide"></div>

  </main>

<script>

    let wins = 0; //Global Variable

    let loss = 0; //Global Variable

    let numberOfFaces; // = 5;

    let numFace;  // static

    let newStart = true; // check for new game

    const theLeftSide = document.getElementById("leftSide");

    const theRightSide = document.getElementById('rightSide');

    // theLeftSide var now controls all properties/methods of the div element with the id “leftSide”

    function generateFaces() {

        if(newStart == true){

          numberOfFaces = document.getElementById('numFace').value;

          numFace = document.getElementById('numFace').value;

        }

        document.getElementById("btn").disabled = true;

        for (i = 1; i <= numberOfFaces; i++) {  // standard for loop

            let randomTop = Math.floor(Math.random() \* 400);

            let randomLeft = Math.floor(Math.random() \* 400);

            const face = document.createElement("img");

                face.src = "../images/smile.png";

                face.style.top = randomTop + "px";

                face.style.left = randomLeft + "px";

            theLeftSide.appendChild(face);

        }

        const leftSideImages = theLeftSide.cloneNode(true);

        leftSideImages.removeChild(leftSideImages.lastChild);

        theRightSide.appendChild(leftSideImages);

        theLeftSide.lastChild.addEventListener('click', nextLevel);

        document.getElementById("x").addEventListener('click', gameOver);

    }

    function nextLevel(event) {

        wins = wins + 1;

        document.getElementById("win").innerHTML = wins;

        newStart = false;

        event.stopPropagation();

        while (theLeftSide.firstChild) {

            theLeftSide.removeChild(theLeftSide.firstChild);

        }

        theRightSide.innerHTML = ''; //alturnative to remove child loop

        numberOfFaces = Number(numberOfFaces) + Number(numFace);  //5;

        generateFaces();

    } //End of nextLevel function

    function gameOver() {

        loss = loss + 1;

        document.getElementById("los").innerHTML = loss;

        alert('You Lose!');

        document.getElementById("btn").disabled = false;

        document.getElementById("x").removeEventListener('click', gameOver);

        theLeftSide.lastChild.removeEventListener('click', nextLevel)

        while (theLeftSide.firstChild) {

            theLeftSide.removeChild(theLeftSide.firstChild);

        }

        theRightSide.innerHTML = '';

        numberOfFaces = document.getElementById('numFace').value;  //5;

        newStart = true;

    }

</script>

</body>

</html>