**Shuffle & Sort task documentation**

* Create an html file and add html structure to it
* Within HTML structured body as per our plan and design, create the 9 boxes in the left side
* On the right side we need to add two buttons namely Sort and Shuffle.
* **Left side: -** Create a parent element and within it create a child element

(1 to 9 Boxes)

Child elements 1 to 9 are arranged in any order and we apply CSS for the child elements



* **Right Side: -** As per design we already placed two buttons, when we use any of this buttons code executes and result shows in left side boxes



* Whenever you click on Right side buttons- click event will trigger and trigger will call the script functions to implement the code.
* **Logic in shuffle function: -** Get the children elements by using parent element ID

var list = document.getElementById("shuffle-sort-box");

var nodes = list.children;

* When we call for the shuffle function

function shuffleNodes(){

          var nodes = list.children, i = 0;

          nodes = Array.prototype.slice.call(nodes);

          nodes = shuffle(nodes);

        // Append all child elements to parent element by using appendChild() method.

          while(i < nodes.length)

          {

              list.appendChild(nodes[i]);

              ++i;

          }

      }

* Generate random numbers and exchange the element by using it within the array

      function shuffle(items){

        var cached = items.slice(0), temp, i = cached.length, rand;

        // Generate random numbers and exchange the element by using random numbers.

        while(--i)

        {

            rand = Math.floor(i \* Math.random());

            temp = cached[rand];

            cached[rand] = cached[i];

            cached[i] = temp;

        }

        return cached;

      }

* Append all child element to parent element by using appendChild() method.

      function shuffleNodes(){

          var nodes = list.children, i = 0;

          nodes = Array.prototype.slice.call(nodes);

          nodes = shuffle(nodes);

        // Append all child elements to parent element by using appendChild() method.

          while(i < nodes.length)

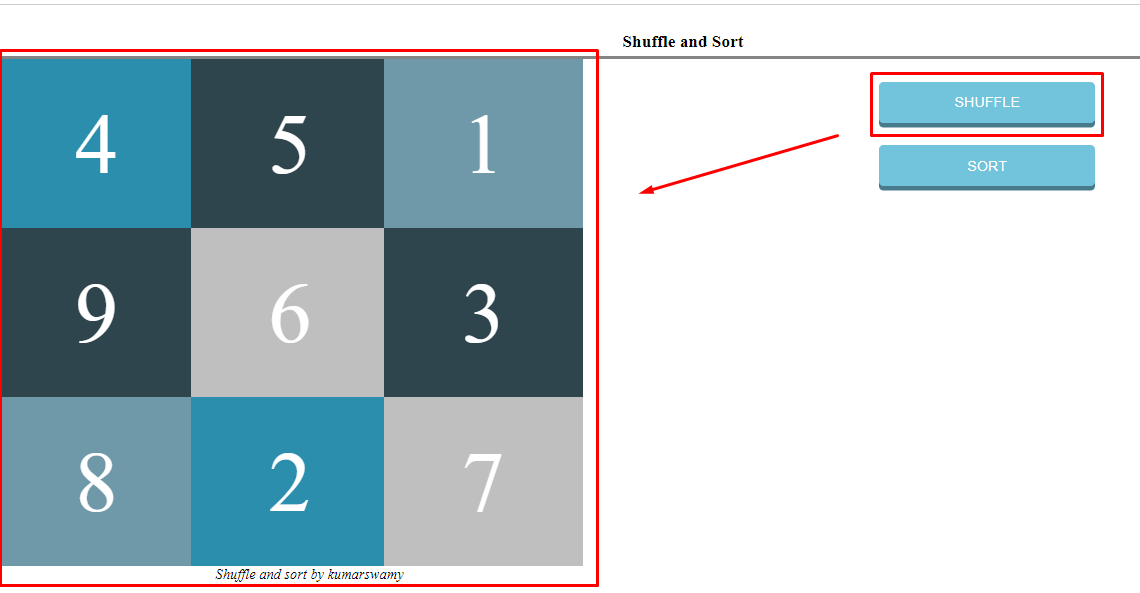
          {

              list.appendChild(nodes[i]);

              ++i;

          }

      }

**Output :**-  


* **Logic in sort function: -** Get the children elements by using parent element ID

var list = document.getElementById("shuffle-sort-box");

var nodes = list.children;

* When we call for the sort function

function sortNodes() {

        // Get the all child nodes by using childNodes.

        var items = list.childNodes;

        var itemsArr = [];

        // In child nodes check for elements(nodeType), if element is found push into array variable(itemsArr variable) else skip.

        for (var i in items) {

            console.log(items[i],'nodetype', items[i].nodeType)

            if (items[i].nodeType == 1) {

                itemsArr.push(items[i]);

            }

        }

        // Sort array element by using compare function with in sort method.

        itemsArr.sort(function(a, b) {

          return a.innerHTML == b.innerHTML

                  ? 0

                  : (a.innerHTML > b.innerHTML ? 1 : -1);

        });

        // Append all child element to parent element by using appendChild() method.

        for (i = 0; i < itemsArr.length; ++i) {

          list.appendChild(itemsArr[i]);

        }

      }

* In child nodes check for elements(nodeType), if element is found push into array variable (itemsArr variable) else skip.

        for (var i in items) {

            console.log(items[i],'nodetype', items[i].nodeType)

            if (items[i].nodeType == 1) {

                itemsArr.push(items[i]);

            }

        }

* Sort array element by using compare function with in sort method.

        itemsArr.sort(function(a, b) {

          return a.innerHTML == b.innerHTML

                  ? 0

                  : (a.innerHTML > b.innerHTML ? 1 : -1);

        });

* Append all child elements to parent element by using appendChild() method.

        for (i = 0; i < itemsArr.length; ++i) {

          list.appendChild(itemsArr[i]);

        }

**Output:-**

Graphical user interface, application

Description automatically generated

**Conclusion: -**

In the HTML page click on Shuffle button then in the left side we can see 1 to 9 boxes are in different format.



 After appearing of this screen, Click on Sort button and event triggers and in left side screen we can see output of 1 to 9 boxes are arranged in order format.

