

```
// JavaScript Document
var _fcPlayer = {}; //namespace
/**
_fcPlayer.mediaLocation : use this to define the URL of the media folders.
..audio and ..images specify a folder. ..url defines the xml file location
for a particular product. Depending on the entire structure, this might
need to be moved to the settings
**/
_fcPlayer.mediaLocation = {};
_fcPlayer.mediaLocation.audio = settings.mediaLocation.audio;
_fcPlayer.mediaLocation.images = settings.mediaLocation.images;
_fcPlayer.mediaLocation.url = settings.mediaLocation.xmlURL;

/**
_fcPlayer.curentCardOnDisplay : this object is used as the main means of selection and navigation.
Each thubmnail is assigned an internal id, constructed as unit_lesson_item. This internal pointer
does NOT correspond to the physical unit/lesson numbering, but only as an internal index.
**/

_fcPlayer.curentCardOnDisplay = {};
_fcPlayer.curentCardOnDisplay.Unit = "1";
_fcPlayer.curentCardOnDisplay.Lesson = "1";
_fcPlayer.curentCardOnDisplay.Card = "0";

/**
The SubsetIndex is the index of the currently selected object with reference to the
current active subset. The active subset is defined by the cards that are currently
visible, as this is set from the selected filters and the card order. The total number
of cards in the current subset is kept in the the SubsetTotals property.
**/

_fcPlayer.curentCardOnDisplay.SubsetIndex = 0;
_fcPlayer.curentCardOnDisplay.SubsetTotals = 0;

/**
The current side of the active card and the global setting
**/

_fcPlayer.curentCardOnDisplay.setFace = 0;
_fcPlayer.curentCardOnDisplay.currentFace = 0;

/**
The structure of the Units and Lessons internally is sequential. In order to reflect
the real world scenario, these properties hold the actual number of Units, Lessons
as defined in the "no" attribute of the XML file
**/

_fcPlayer.translateLessons = [];
_fcPlayer.translateUnits = [""];

/**
```

```

objects to hold the settings and the object that represents the XML structure
  **/
_fcPlayer.globalSettings = {};
_fcPlayer.allCards = {};

/**
theLocation : id of element (X_X_X)
breaks down the id and returns an object with the unit, lesson, card of the id
  **/
function extractLocation(theLocation) {
    var GPSposition = {};
    GPSposition.Unit = "";
    GPSposition.Lesson = "";
    GPSposition.Card = "";
    GPSposition.Unit = theLocation.substring(0, theLocation.indexOf("_"));
    GPSposition.Lesson = theLocation.substring(theLocation.indexOf("_") + 1, theLocation.lastIndexOf("_"));
    GPSposition.Card = theLocation.substring(theLocation.lastIndexOf("_") + 1, theLocation.length);
    return GPSposition;
}

/**
elems : element tag name
shuffles ALL elements of the same type and re-inserts into DOM
  **/
function shuffle(elems) {
    allElems = (function() {
        var ret = [],
            l = elems.length;
        while (l--) {
            ret[ret.length] = elems[l];
        }
        return ret;
    })();

    var shuffled = (function() {
        var l = allElems.length,
            ret = [];
        while (l--) {
            var random = Math.floor(Math.random() * allElems.length),
                randEl = allElems[random].cloneNode(true);
            allElems.splice(random, 1);
            ret[ret.length] = randEl;
        }
        return ret;
    })(),
        l = elems.length;
    while (l--) {
        elems[l].parentNode.insertBefore(shuffled[l], elems[l].nextSibling);
        elems[l].parentNode.removeChild(elems[l]);
    }
    document.getElementById("shuffle").value = _fcPlayer.globalSettings.buttonLabels.unshuffle;

```

```

        document.getElementById("shuffle").name = "unshuffle";
        document.getElementById("shuffle").firstChild.firstChild.nodeValue = _fcPlayer.globalSettings.buttonLabels.unshuffle;
    }
    // Usage:
    //shuffle( document.getElementsByTagName('li') );

    /**
    takes theInput and returns in the XX (0theInput) form
    */
    function numberOfZeroes(theInput) {
        if (Number(theInput) < 10) {
            theInput = "0" + theInput;
        }
        return theInput;
    }

    /**
    This is a helper function for the Unshuffling. Unshuffling works by sorting the Id's, using the sort() function.
    However, for sort to work properly, you need to have the same number of digits in all the elements. If not so,
    30 will appear before 3. Therefore, 3 needs to appear as 03.
    */
    function addLeadingZeroes(theInput) {
        var leadedZero = extractLocation(theInput);
        var leadedWithZeroes = numberOfZeroes(leadedZero.Unit) + "_" + numberOfZeroes(leadedZero.Lesson) + "_" + numberOfZeroes(leadedZero.Card);
        return leadedWithZeroes
    }

    /**
    Restore the original order of elements and change the button label
    */
    function unshuffle() {
        var mylist = $('#left > ul');
        var listitems = mylist.children('li').get();
        listitems.sort(function(a, b) {
            var compA = addLeadingZeroes($(a).attr("id"));
            var compB = addLeadingZeroes($(b).attr("id"));
            return (compA < compB) ? -1 : (compA > compB) ? 1 : 0;
        })
        $.each(listitems, function(idx, itm) {
            mylist.append(itm);
        });
        document.getElementById("shuffle").value = _fcPlayer.globalSettings.buttonLabels.shuffle;
        document.getElementById("shuffle").name = "shuffle";
        document.getElementById("shuffle").firstChild.firstChild.nodeValue = _fcPlayer.globalSettings.buttonLabels.shuffle;
    }

    /**
    Set the counter of cards
    */
    function updateVisuals() {

```

```

var current = _fcPlayer.curentCardOnDisplay.SubsetIndex + 1;
var total = $("#left > ul > li.isOn").length;
var theText = current + " / " + total;
document.getElementById("counter").value = theText;
}

function setLastCardMargin() {
    try {
        document.getElementsByClassName("isOn likelastchild")[0].className = "isOn";
    } catch (err) {}
    if (document.getElementsByClassName("isOn")[document.getElementsByClassName("isOn").length - 1].nextSibling) {
        document.getElementsByClassName("isOn")[document.getElementsByClassName("isOn").length - 1].className = "isOn likelastchild";
    }
}

/**
theCurrentCardIndex : The subset's index of the card that will be displayd
Display the "theCurrentCardIndex" card in the centre screen
**/
function setCurrentCard(theCurrentCardIndex) {
    _fcPlayer.curentCardOnDisplay.SubsetIndex = theCurrentCardIndex;
    removeClass(document.getElementsByClassName("isActive")[0], "isActive");
    var ls = $("#left > ul > li.isOn")[theCurrentCardIndex];
    ls.className += (" isActive");
    if (ls.children[0].className != "empty") {
        document.getElementById("center").children[0].src = $("#left > ul > li.isOn")[theCurrentCardIndex].children[0].src;
        document.getElementById("center").children[1].innerHTML = "";
    } else {
        document.getElementById("center").children[0].src = "";
        document.getElementById("center").children[1].innerHTML = $("#left > ul > li.isOn")[theCurrentCardIndex].children[1].innerHTML;
    }
}

function selectTheUnits(that, evt) {
    //issue#5355070000004726011
    //spare some cpu and dont do unless necessary
    if (document.getElementById("shuffle").value == settings.buttonLabels.unshuffle) {
        toggleShuffle("unshuffle");
    }
    if (document.getElementById("hide").value == settings.buttonLabels.show) {
        toggleHide("show");
    }
    _fcPlayer.curentCardOnDisplay.currentFace = _fcPlayer.curentCardOnDisplay.setFace;
    $("#lessons").html(createLessonDropDowns(that.selectedIndex));
    _fcPlayer.curentCardOnDisplay.Unit = that.selectedIndex;
    _fcPlayer.curentCardOnDisplay.Card = "0";
    hideItems(that.selectedIndex);
    setLastCardMargin();
    setCurrentCard(0);
    determineCurrentCardLocation(0);
    determineTheMediaType(_fcPlayer.curentCardOnDisplay.Unit + "_" + _fcPlayer.curentCardOnDisplay.Lesson + "_" + _fcPlayer.curentCardOnDisplay.Card);
}

```

```

        checkButtonsState(0);
        updateVisuals();
        addAudioElement();
    }

function selectTheLessons(that, evt) {
    _fcPlayer.curentCardOnDisplay.currentFace = _fcPlayer.curentCardOnDisplay.setFace;
    hideLessonItems(that.selectedIndex);
    fcPlayer.curentCardOnDisplay.Lesson = that.selectedIndex;
    _fcPlayer.curentCardOnDisplay.Card = "0";
    setLastCardMargin();
    setCurrentCard(0);
    determineCurrentCardLocation(0);
    determineTheMediaType(_fcPlayer.curentCardOnDisplay.Unit + "_" + _fcPlayer.curentCardOnDisplay.Lesson + "_" + _fcPlayer.curentCardOnDisplay.Card);
    checkButtonsState(0);
    updateVisuals();
    addAudioElement();
}

function calculateTextSize(test_obj, my_str, maxWidth, maxHeight, step) {
    var myResults = new Array(2);
    document.getElementById(test_obj).innerHTML = ""
    document.getElementById(test_obj).scrollHeight = 0
    document.getElementById(test_obj).innerHTML = my_str;
    if ((document.getElementById(test_obj).scrollHeight > maxHeight) || (document.getElementById(test_obj).scrollWidth > maxWidth)) {
        document.getElementById(test_obj).style.fontSize = (parseFloat(document.getElementById(test_obj).style.fontSize) - step) + "px";
        //0.78 comes from the ratio of height and width, 136/175 = 0.78 or 381/490 = 0.78
        myResults = calculateTextSize(test_obj, my_str, maxWidth, maxHeight, step)
    } else {
        myResults[0] = document.getElementById(test_obj).scrollHeight
        myResults[1] = document.getElementById(test_obj).style.fontSize
    }
    return myResults;
}

function callback(myObj, myH) {
    return function() {
        myObj.style = 'height:' + myH + "px";
    }
}

function flipSides() {
    document.getElementById("progress").style.display = "block";
    var tempScrollLoc = document.getElementById("left").scrollTop;
    var ls = document.getElementsByTagName("li");
    _fcPlayer.curentCardOnDisplay.setFace = Math.abs(_fcPlayer.curentCardOnDisplay.setFace - 1);
    var localIterationCards = {};
    for (var i = 0; i < ls.length; i++) {
        localIterationCards = extractLocation(ls[i].id);
        var theOtherType = _fcPlayer.allCards["Unit" + localIterationCards.Unit]["Lesson" + localIterationCards.Lesson][localIterationCards.Card][2][_fcPlayer.curentCardOnDisplay.setFace];
        var locImgRef = ls[i].children[0];
    }
}

```

```

        var locSpanRef = ls[i].children[1];
        if (theOtherType == "i") { //TARGET!!!!!!!
            ls[i].children[0].src = _fcPlayer.mediaLocation.images + _fcPlayer.allCards["Unit" + localIterationCards.Unit]["Lesson" +
localIterationCards.Lesson][localIterationCards.Card][0][_fcPlayer.curentCardOnDisplay.setFace];
            ls[i].children[0].className = "imgThumb";
            ls[i].children[1].innerHTML = "";
            ls[i].children[1].className = "empty";
        } else {
            ls[i].children[0].src = "#";
            ls[i].children[0].className = "empty";
            ls[i].children[1].innerHTML = _fcPlayer.allCards["Unit" + localIterationCards.Unit]["Lesson" +
localIterationCards.Lesson][localIterationCards.Card][0][_fcPlayer.curentCardOnDisplay.setFace];
            ls[i].children[1].className = "textThumb";

            /*RESIZE TEXT - FULL*/
            document.getElementById("txt_test").style.fontSize = "24px";
            var myResults = calculateTextSize("txt_test", _fcPlayer.allCards["Unit" + localIterationCards.Unit]["Lesson" +
localIterationCards.Lesson][localIterationCards.Card][0][_fcPlayer.curentCardOnDisplay.setFace], 175, 136, 0.1)
            $(ls[i].children[1]).css("text-align", "center");
            if (myResults[0] > 70) {
                $(ls[i].children[1]).css("text-align", "left");
            }
            //setTimeout(callback(ls[i].children[1], myResults[0]), 2000);
            /*$("#l_1_3 > span").css("height", myResults[0] + "px");*/
            $(ls[i].children[1]).css("height", myResults[0] + "px");
            $(ls[i].children[1]).css("font-size", myResults[1]);
            /* I finally used jQuery because the following JS code doesn't work in IE...*/
            //ls[i].children[1].style='font-size:'+myResults[1]+'; height:'+myResults[0]+'px';

        }
    }
    //flip
    if (_fcPlayer.curentCardOnDisplay.setFace != _fcPlayer.curentCardOnDisplay.currentFace) {
        flipCenterCard();
    }
    //awitch side of ALL cards,
    //TODO: display waiting information
    var ua = navigator.userAgent.toLowerCase();
    if (ua.indexOf('safari') != -1) {
        if (ua.indexOf('chrome') > -1) {
            //alert("1") // chrome
        } else {
            //safari
            var objDiv = document.getElementById("left");
            objDivTargScroll = tempScrollLoc;
            document.getElementById("left").style.display = "none";
            window.setTimeout(function() {
                document.getElementById("left").style.display = "block";
                objDiv.scrollTop = objDivTargScroll;
            }, 10);
        }
    }

```

```

    }
    document.getElementById("progress").style.display = "none";
}

/**
theDirection : shuffle or unshuffle
Handles the shuffle button click for both occasions.
**/

function toggleShuffle(theDirection) {
    _fcPlayer.curentCardOnDisplay.currentFace = _fcPlayer.curentCardOnDisplay.setFace;
    if (theDirection == "shuffle") {
        shuffle(document.getElementsByTagName('li'));
        dualAppearance("unshuffle");
    } else {
        unshuffle();
        dualAppearance("shuffle");
    }
    setLastCardMargin();
    setCurrentCard(0);
    determineCurrentCardLocation(0);
    determineTheMediaType(_fcPlayer.curentCardOnDisplay.Unit + "_" + _fcPlayer.curentCardOnDisplay.Lesson + "_" + _fcPlayer.curentCardOnDisplay.Card);
    checkButtonsState(0);
    updateVisuals();
    addAudioElement();

    resizeImgs();
}

/**
theDirection : show or hide
Handles the hide button click for both occasions.
**/

//Keep Hide Thumbs Status
var clickedHideThumbs = false;

//Used in Back next to update size of imgs
function resizeImgs() {

    var myHiddenThumbsImg = document.getElementById("centerfold");
    var myHiddenThumbsTheText = document.getElementById("centerfoldt");

    if (clickedHideThumbs) {

        if (myHiddenThumbsImg.width > myHiddenThumbsImg.height) {
            myHiddenThumbsImg.className += " imgThumbsHidden-width";

        } else {
            myHiddenThumbsImg.className += " imgThumbsHidden-height";
        }
    }
}

```

```

    }
}

window.onresize = function() {
    resizeImgs()
};

function dualAppearance(state) {
    var localProps = [];
    var myName = "";
    var imagesArray = ["", "", "", ""]
    try {
        if ((_fcPlayer.globalSettings.appearanceStyles[state].normal.image) && (_fcPlayer.globalSettings.appearanceStyles[state].normal.image != "")) {
            imagesArray[0] = _fcPlayer.globalSettings.appearanceStyles[state].normal.image;
            document.getElementsByName(state)[0].getElementsByTagName("img")[0].setAttribute("src", imagesArray[0]);
        }

        if ((_fcPlayer.globalSettings.appearanceStyles[state].hover.image) && (_fcPlayer.globalSettings.appearanceStyles[state].hover.image != "")) {
            imagesArray[1] = _fcPlayer.globalSettings.appearanceStyles[state].hover.image;
        }

        if ((_fcPlayer.globalSettings.appearanceStyles[state].active.image) && (_fcPlayer.globalSettings.appearanceStyles[state].active.image != "")) {
            imagesArray[2] = _fcPlayer.globalSettings.appearanceStyles[state].active.image;
        }

        if ((_fcPlayer.globalSettings.appearanceStyles[state].disabled.image) && (_fcPlayer.globalSettings.appearanceStyles[state].disabled.image != "")) {
            imagesArray[3] = _fcPlayer.globalSettings.appearanceStyles[state].disabled.image;
        }
        //alert(imagesArray.join("@"))
        document.getElementsByName(state)[0].setAttribute("images", imagesArray.join("@"));

    } catch (err) {
        // if(window.console && console.error("Error:" + err));
    }
}

//Updated 2019 resize img according to hide/show thumbs
function toggleHide(theDirection) {

    var myHiddenThumbsSt = document.getElementsByTagName("section")[0];
    var myHiddenThumbsImg = document.getElementById("centerfold");
    var myHiddenThumbsTheText = document.getElementById("centerfoldt");

    if (theDirection == "hide") {
        document.getElementById("hide").name = "show";
        document.getElementById("hide").value = _fcPlayer.globalSettings.buttonLabels.show;
        document.getElementById("hide").firstChild.firstChild.nodeValue = _fcPlayer.globalSettings.buttonLabels.show;

        document.getElementById("left").style.visibility = "hidden";
        //document.getElementById("left").style.display = "none";
    }
}

```



```
clickedHideThumbs = true;

document.getElementById("hide").className += " hideThumbsHidden";
document.getElementById("audiop").className += " audiopThumbsHidden";
document.getElementById("face").className += " faceThumbsHidden";
document.getElementById("shuffle").className += " shuffleThumbsHidden";
document.getElementById("flip").className += " flipThumbsHidden";
document.getElementById("previous").className += " previousThumbsHidden";
document.getElementById("counter").className += " counterThumbsHidden";
document.getElementById("next").className += " nextThumbsHidden";
document.getElementById("center").className += " sectionThumbsHidden";
document.getElementsByClassName("designFill")[0].className += " dFillThumbsHidden";
document.getElementById("filterby").className += " filterbyThumbsHidden";
document.getElementById("units").className += " unitsThumbsHidden";
document.getElementById("lessons").className += " lessonsThumbsHidden";

/*RESIZE TEXT - FULL*/
var myPerc = 583.5 / 490;
$(document.getElementById("centerfoldt")).css("font-size", parseFloat(document.getElementById("centerfoldt").style.fontSize) * myPerc + "px")
$(document.getElementById("centerfoldt")).css("width", "583.5px")
/*parseFloat(document.getElementById("centerfoldt").style.width)*myPerc+"px")*/
$(document.getElementById("centerfoldt")).css("height", parseFloat(document.getElementById("centerfoldt").style.height) * myPerc + "px")

if (myHiddenThumbsImg.width > myHiddenThumbsImg.height) {
    myHiddenThumbsImg.className += " imgThumbsHidden-width";
} else {
    myHiddenThumbsImg.className += " imgThumbsHidden-height";
}

dualAppearance("show");
} else {
    document.getElementById("hide").name = "hide";
    document.getElementById("hide").value = _fcPlayer.globalSettings.buttonLabels.hide;
    document.getElementById("hide").firstChild.firstChild.nodeValue = _fcPlayer.globalSettings.buttonLabels.hide;

    document.getElementById("left").style.visibility = "visible";
    clickedHideThumbs = false;
    document.getElementById("hide").className = document.getElementById("hide").className.replace(/(?:^|\s)hideThumbsHidden(?:\S)/g, '')
    document.getElementById("audiop").className = document.getElementById("audiop").className.replace(/(?:^|\s)audiopThumbsHidden(?:\S)/g, '')
    document.getElementById("face").className = document.getElementById("face").className.replace(/(?:^|\s)faceThumbsHidden(?:\S)/g, '')
    document.getElementById("shuffle").className = document.getElementById("shuffle").className.replace(/(?:^|\s)shuffleThumbsHidden(?:\S)/g, '')
    document.getElementById("flip").className = document.getElementById("flip").className.replace(/(?:^|\s)flipThumbsHidden(?:\S)/g, '')
    document.getElementById("previous").className = document.getElementById("previous").className.replace(/(?:^|\s)previousThumbsHidden(?:\S)/g, '')
    document.getElementById("counter").className = document.getElementById("counter").className.replace(/(?:^|\s)counterThumbsHidden(?:\S)/g, '')
    document.getElementById("next").className = document.getElementById("next").className.replace(/(?:^|\s)nextThumbsHidden(?:\S)/g, '')
    document.getElementById("center").className = document.getElementById("center").className.replace(/(?:^|\s)sectionThumbsHidden(?:\S)/g, '')
    document.getElementsByClassName("designFill")[0].className = document.getElementsByClassName("designFill")[0].className.replace(/(?:^|\s)dFillThumbsHidden(?:\S)/g, '')
    document.getElementById("filterby").className = document.getElementById("filterby").className.replace(/(?:^|\s)filterbyThumbsHidden(?:\S)/g, '')
    document.getElementById("units").className = document.getElementById("units").className.replace(/(?:^|\s)unitsThumbsHidden(?:\S)/g, '')
    document.getElementById("lessons").className = document.getElementById("lessons").className.replace(/(?:^|\s)lessonsThumbsHidden(?:\S)/g, '')
```

```
/*RESIZE TEXT - FULL*/
var myPerc = 490 / 583.5;
$(document.getElementById("centerfoldt")).css("font-size", parseFloat(document.getElementById("centerfoldt").style.fontSize) * myPerc + "px")
$(document.getElementById("centerfoldt")).css("width", "490px")
/*parseFloat(document.getElementById("centerfoldt").style.width)*myPerc+"px")*/
$(document.getElementById("centerfoldt")).css("height", parseFloat(document.getElementById("centerfoldt").style.height) * myPerc + "px")

if (myHiddenThumbsImg.width > myHiddenThumbsImg.height) {
    myHiddenThumbsImg.className = myHiddenThumbsImg.className.replace(/(?:^\|s)imgThumbsHidden-width(?:\S)/g, '')
} else {
    myHiddenThumbsImg.className = myHiddenThumbsImg.className.replace(/(?:^\|s)imgThumbsHidden-height(?:\S)/g, '')
}

dualAppearance("hide");

}

}

function getAllSelectors() {
    var ret = [];
    for (var i = 0; i < document.styleSheets.length; i++) {
        var rules = document.styleSheets[i].rules || document.styleSheets[i].cssRules;
        for (var x in rules) {
            if (typeof rules[x].selectorText == 'string')
                ret.push(rules[x].selectorText);
        }
    }
    return ret;
}

function selectorExists(selector) {
    var selectors = getAllSelectors();
    for (var i = 0; i < selectors.length; i++) {
        if (selectors[i] == selector)
            return true;
    }
    return false;
}

function disableMe(elem, disabled) {
    if (disabled == true) {
        elem.disabled = true;
        elem.style.color = "grey";
        elem.className += " opaque";
    } else {
        elem.disabled = false;
        elem.style.color = "#FAF3E4";
        removeClass(elem, "opaque");
    }
}
```

```

/**
thePosition : The subset's index of the current card.
This checks the index against the subset and activates/deactivates the
previous/next buttons
**/
function checkButtonsState(thePosition) {
    if (thePosition == 0) {
        disableMe(document.getElementById("previous"), true);
        disableMe(document.getElementById("next"), false);
    } else if (thePosition == $("#left > ul > li.isOn").length - 1) {
        disableMe(document.getElementById("previous"), false);
        disableMe(document.getElementById("next"), true);
    } else {
        disableMe(document.getElementById("previous"), false);
        disableMe(document.getElementById("next"), false);
    }
}

/**
theIdToSearch : element id
This sets the card's subset index
**/
function locateTheIndex(theIdToSearch) {
    var ls = $("#left > ul > li.isOn");
    var thePosition = 0;
    for (var i = 0; i < ls.length; i++) {
        if (ls[i].id == theIdToSearch) {
            thePosition = i;
            break;
        }
    }
    return thePosition
}

function determineCurrentCardLocation(theIndex) {
    var tempObj = {};
    tempObj = extractLocation($("#left > ul > li.isOn")[theIndex].id);
    _fcPlayer.curentCardOnDisplay.Unit = tempObj.Unit;
    _fcPlayer.curentCardOnDisplay.Lesson = tempObj.Lesson;
    _fcPlayer.curentCardOnDisplay.Card = tempObj.Card;
}

function toggleSequence(theDirection) {
    fcPlayer.curentCardOnDisplay.currentFace = fcPlayer.curentCardOnDisplay.setFace;
    if (theDirection == "next") {
        _fcPlayer.curentCardOnDisplay.SubsetIndex = _fcPlayer.curentCardOnDisplay.SubsetIndex + 1;
        setCurrentCard(_fcPlayer.curentCardOnDisplay.SubsetIndex);
    } else {
        _fcPlayer.curentCardOnDisplay.SubsetIndex = _fcPlayer.curentCardOnDisplay.SubsetIndex - 1;
        setCurrentCard(fcPlayer.curentCardOnDisplay.SubsetIndex);
    }
}

```

```

    }
    checkButtonsState(_fcPlayer.curentCardOnDisplay.SubsetIndex);
    determineCurrentCardLocation(_fcPlayer.curentCardOnDisplay.SubsetIndex);
    determineTheMediaType(_fcPlayer.curentCardOnDisplay.Unit + "_" + _fcPlayer.curentCardOnDisplay.Lesson + "_" + _fcPlayer.curentCardOnDisplay.Card);
    updateVisuals();
    addAudioElement();

    resizeImgs()
}

```

```

function crossBrowserSourceElement(evnt, theNodeType) {
    var fResult = {};
    fResult.condtn = false;
    fResult.theTargetType = "";
    fResult.theTargetName = "";
    try {
        if (evnt.srcElement) { //chrome property
            fResult.theTargetType = evnt.srcElement;
            //          fResult.theTargetName = evnt.srcElement.name;
        } else {
            fResult.theTargetType = evnt.originalTarget; //firefox
            //          fResult.theTargetName = evnt.originalTarget.name;
        }
        var locTest = "";
        if (fResult.theTargetType.nodeName.toLowerCase() == theNodeType) {
            locTest = fResult.theTargetType;
        } else {
            locTest = findUpTag(fResult.theTargetType, theNodeType);
        }
        if (locTest.nodeName.toLowerCase() != theNodeType) {
            evnt.preventDefault();
        } else {
            fResult.theTargetName = locTest.name;
            fResult.condtn = true;
            //allow bubbling
        }
    } catch (err) {
        //some error handling - usually the click was on the select and it has no name?
    }
    return fResult;
}

```

```

function flipCenterCard() {
    fcPlayer.curentCardOnDisplay.currentFace = Math.abs(fcPlayer.curentCardOnDisplay.currentFace - 1);
    var theOtherType = _fcPlayer.allCards["Unit" + _fcPlayer.curentCardOnDisplay.Unit]["Lesson" +
_fcPlayer.curentCardOnDisplay.Lesson][_fcPlayer.curentCardOnDisplay.Card][2][_fcPlayer.curentCardOnDisplay.currentFace]

    //we need to know what kind of media the other side has. This is defined in the XML, i for image and t for text. Depending on the type,
    //we know which child element will be visible, the text (2nd child or the image 1st child)
    if (theOtherType == "i") {

```

```

        document.getElementById("centerfold").src = _fcPlayer.mediaLocation.images + _fcPlayer.allCards["Unit" + _fcPlayer.curentCardOnDisplay.Unit]["Lesson" +
_fcPlayer.curentCardOnDisplay.Lesson][_fcPlayer.curentCardOnDisplay.Card][0][_fcPlayer.curentCardOnDisplay.currentFace];
        document.getElementById("centerfold").className = "theImages";
        document.getElementById("centerfoldt").className = "empty";
        if (document.getElementById("hide").name == "show") {
            if (document.getElementById("centerfold").width > document.getElementById("centerfold").height) {
                document.getElementById("centerfold").className += " imgThumbsHidden-width";
            } else {
                document.getElementById("centerfold").className += " imgThumbsHidden-height";
            }
        }
    } else {
        document.getElementById("centerfold").src = "#";
        document.getElementById("centerfold").className = "empty";
        document.getElementById("centerfoldt").innerHTML = _fcPlayer.allCards["Unit" + _fcPlayer.curentCardOnDisplay.Unit]["Lesson" +
_fcPlayer.curentCardOnDisplay.Lesson][_fcPlayer.curentCardOnDisplay.Card][0][_fcPlayer.curentCardOnDisplay.currentFace];
        document.getElementById("centerfoldt").className = "theText";

        /*RESIZE TEXT*/
        document.getElementById("txt2 test").style.fontSize = "67.2px";
        var myResults = calculateTextSize("txt2 test", _fcPlayer.allCards["Unit" + _fcPlayer.curentCardOnDisplay.Unit]["Lesson" +
_fcPlayer.curentCardOnDisplay.Lesson][_fcPlayer.curentCardOnDisplay.Card][0][_fcPlayer.curentCardOnDisplay.currentFace], 490, 381, 0.1 * (67.2 / 24))
        $(document.getElementById("centerfoldt")).css("text-align", "center");
        if (myResults[0] > 180) {
            $(document.getElementById("centerfoldt")).css("text-align", "left");
        }
        $(document.getElementById("centerfoldt")).css("height", myResults[0] + "px");
        $(document.getElementById("centerfoldt")).css("font-size", myResults[1]);
        /* I finally used jQuery because the following JS code doesn't work in IE...*/
        //document.getElementById("centerfoldt").style='font-size:'+myResults[1]+'; height:'+myResults[0]+'px';

        /*RESIZE TEXT - FULL*/
        if (document.getElementById("hide").name == "show") {
            var myPerc = 583.5 / 490;
            $(document.getElementById("centerfoldt")).css("font-size", parseFloat(document.getElementById("centerfoldt").style.fontSize) * myPerc + "px")
            $(document.getElementById("centerfoldt")).css("width", "583.5px")
            /*parseFloat(document.getElementById("centerfoldt").style.width)*myPerc+"px")*/
            $(document.getElementById("centerfoldt")).css("height", parseFloat(document.getElementById("centerfoldt").style.height) * myPerc + "px")
        }

    }
    addAudioElement();
}
/**
the function that runs when a button is selecetd. Depending on the button that
was clicked, appropriate action is taken
**/
function delegateButtonClicks(evt, that, theSenderName) {
    switch (theSenderName) {
        case "face":
            flipSides();
    }
}

```

```

        break;
    case "shuffle":
    case "unshuffle":
        toggleShuffle(theSenderName);
        break;
    case "hide":
    case "show":
        toggleHide(theSenderName);
        break;
    case "flip":
        flipCenterCard();
        break;
    case "audio":
        audioPlayDirectly();
        break;
    case "previous":
    case "next":
        toggleSequence(theSenderName);
        break;
    default:
    }
}

function determineTheMediaType(theItem) {
    try {
        var locImgRef = document.getElementById(theItem).children[0];
        var locSpanRef = document.getElementById(theItem).children[1];
        if (locImgRef.className != "empty") {
            document.getElementById("centerfold").src = locImgRef.src;
            document.getElementById("centerfold").className = "theImages";
            document.getElementById("centerfoldt").innerHTML = "";
            document.getElementById("centerfoldt").className = "empty";
        } else {
            document.getElementById("centerfold").src = "#";
            document.getElementById("centerfold").className = "empty";
            document.getElementById("centerfoldt").innerHTML = locSpanRef.innerHTML;
            document.getElementById("centerfoldt").className = "theText"

            /*RESIZE TEXT*/
            document.getElementById("txt2_test").style.fontSize = "67.2px";
            var myResults = calculateTextSize("txt2_test", locSpanRef.innerHTML, 490, 381, 0.1 * (67.2 / 24))
            $(document.getElementById("centerfoldt")).css("text-align", "center");
            if (myResults[0] > 180) {
                $(document.getElementById("centerfoldt")).css("text-align", "left");
            }
            $(document.getElementById("centerfoldt")).css("height", myResults[0] + "px");
            $(document.getElementById("centerfoldt")).css("font-size", myResults[1]);
            /* I finally used jQuery because the following JS code doesn't work in IE...*/
            //document.getElementById("centerfoldt").style='font-size:'+myResults[1]+'; height:'+myResults[0]+'px';

            /*RESIZE TEXT - FULL*/

```

```

        if (document.getElementById("hide").name == "show") {
            var myPerc = 583.5 / 490
            $(document.getElementById("centerfoldt")).css("font-size", parseFloat(document.getElementById("centerfoldt").style.fontSize) * myPerc + "px")
            $(document.getElementById("centerfoldt")).css("width", "583.5px")
            /*parseFloat(document.getElementById("centerfoldt").style.width)*myPerc+"px")*/
            $(document.getElementById("centerfoldt")).css("height", parseFloat(document.getElementById("centerfoldt").style.height) * myPerc + "px")
        }
    }
} catch (err) {
    //clicked on div probably
}
}

function findUpTag(el, tag) {
    while (el.parentNode) {
        el = el.parentNode;
        if (el.tagName.toLowerCase() === tag)
            return el;
    }
    return null;
}

/**
Do when the page has loaded and the DOM is ready.
**/
$(document).ready(function() {
    //remove unnecessary buttons, as defined in the settings
    keepCorrectButtons(settings);
    //label buttons, as defined in the settings
    _fcPlayer.globalSettings = buildGui(settings);

    //we keep one listener for the div instead one for each button, so the page runs faster
    //first we check if a button was clicked and if yes we proceed to handle that click
    document.getElementById("buttons").addEventListener("click", function(e) {
        var locObj = crossBrowserSourceElement(e, "button");
        if (locObj.condtn) {
            delegateButtonClicks(e, this, locObj.theTargetName);
        } else {
            //nothing
        }
    });

    //we keep one listener for the navigational part of the page. when a click
    //happens on that portion, that click is handled accordingly.
    document.getElementById("left").addEventListener("click", function(e) {
        var tempObj = {};
        var target = ""
        if (e.srcElement) {
            target = e.srcElement;
        } else {

```

the LI

```
        target = e.originalTarget;
    }
    //issue#535507000004761035
    if ((target.nodeName.toLowerCase() !== "nav") && (target.nodeName.toLowerCase() !== "div") && (target.nodeName.toLowerCase() !== "ul")) {

        if (target.nodeName.toLowerCase() === "li") {
            target = document.getElementById(target.id).childNodes[1]; //move the click to the span - next function ALWAYS expect the children as the target (ie images, span) and not
        } else if (target.parentNode.nodeName.toLowerCase() === "li") {
            //nothing, retain target but dont branch to the next case
        } else {
            target = findUpTag(target, "span");
        }
        determineTheMediaType(target.parentNode.id);
        _fcPlayer.curentCardOnDisplay.SubsetIndex = locateTheIndex(target.parentNode.id);
        try {
            removeClass(document.getElementById(_fcPlayer.curentCardOnDisplay.Unit + "_" + _fcPlayer.curentCardOnDisplay.Lesson + "_" + _fcPlayer.curentCardOnDisplay.Card), "isActive");
        } catch (err) {}
        tempObj = extractLocation(target.parentNode.id);
        _fcPlayer.curentCardOnDisplay.Unit = tempObj.Unit;
        _fcPlayer.curentCardOnDisplay.Lesson = tempObj.Lesson;
        _fcPlayer.curentCardOnDisplay.Card = tempObj.Card;
        document.getElementById(_fcPlayer.curentCardOnDisplay.Unit + "_" + _fcPlayer.curentCardOnDisplay.Lesson + "_" + _fcPlayer.curentCardOnDisplay.Card).className += " isActive";
        //reset face
        _fcPlayer.curentCardOnDisplay.currentFace = _fcPlayer.curentCardOnDisplay.setFace;
        checkButtonsState(_fcPlayer.curentCardOnDisplay.SubsetIndex);
        updateVisuals();
        addAudioElement();
    } else {
        //do nothing, the UL or the DIV was clicked
    }
});

//the listener for the flipping of the main card #obsolete - moved to buttons
/*document.getElementById("flip").addEventListener("click", function(e){

});*/

//listener for the units drop-down
document.getElementById("units").addEventListener("change", function(e) {
    selectTheUnits(this, e);

    resizeImgs();
});

//listener for the lessons drop-down
document.getElementById("lessons").addEventListener("change", function(e) {
    selectTheLessons(this, e)

    resizeImgs();
});
```



```
importXML( fcPlayer.mediaLocation.url);
updateVisuals();
setCurrentCard(0);
disableMe(document.getElementById("previous"), true);
determineTheMediaType(document.querySelector(".isOn").id); // #ISSUE:535507000004790031 - document.getElementById("left").children[0].children[0].id;
var tester = $('#center');
tester.swipeleft(function() {
    toggleSequence("next");
    // alert("swipeleft" called);
});
tester.swiperight(function() {
    toggleSequence("previous");
    // alert("swiperight" called);
});

//annoying safari ONLY bug
var ua = navigator.userAgent.toLowerCase();
if (ua.indexOf('safari') != -1) {
    if (ua.indexOf('chrome') > -1) {
        //alert("1") // chrome
    } else {
        //safari
        $('#left > ul').waitForImages(function() {
            // All descendant images have loaded, now slide up.
            document.getElementById("left").style.display = "none";
            //alert("ok");
            window.setTimeout(function() {
                document.getElementById("left").style.display = "block"
            }, 10);
            // alert("all loaded");
        });
    }
}
});
```