

```

// JavaScript Document
var xmlloaded = false;

function getInternetExplorerVersion()
// Returns the version of Internet Explorer or a -1
// (indicating the use of another browser).
{
    var rv = -1; // Return value assumes failure.
    if (navigator.appName == 'Microsoft Internet Explorer') {
        var ua = navigator.userAgent;
        var re = new RegExp("MSIE ([0-9]{1,}[\.\.0-9]{0,})");
        if (re.exec(ua) != null) rv = parseFloat(RegExp.$1);
    }
    return rv;
}

function checkVersion() {
    var msg = "You're not using Internet Explorer.";
    var ver = getInternetExplorerVersion();
    if (ver > -1) {
        if (ver >= 8.0) msg = "You're using a recent copy of Internet Explorer."
        else msg = "You should upgrade your copy of Internet Explorer.";
    }
    //alert( msg );
}

function xMouseUp(elem) {
    if (elem.disabled == false) {
        elem.getElementsByTagName("img")[0].setAttribute("src", elem.getAttribute("images").split("@")[0]);
    }
}

function xMouseDown(elem) {
    if (elem.disabled == false) {
        elem.getElementsByTagName("img")[0].setAttribute("src", elem.getAttribute("images").split("@")[2]);
    }
}

function xMouseOut(elem) {
    if (elem.disabled == false) {
        elem.getElementsByTagName("img")[0].setAttribute("src", elem.getAttribute("images").split("@")[0]);
    }
}

function xMouseOver(elem) {
    if (elem.disabled == false) {
        elem.getElementsByTagName("img")[0].setAttribute("src", elem.getAttribute("images").split("@")[1]);
    }
}

```

```

}

function buildGui(setts) {
    var localProps = [];
    var myName = "";
    var firstSheet = document.styleSheets[0];
    var myGradient
    var defAppNormal = 'button { ';
    defAppNormal += 'background: radial-gradient(90% 300%, ' + setts.appearanceStyles.default.normal.colour1 + ', ' +
setts.appearanceStyles.default.normal.colour2 + '); ';
    defAppNormal += 'background-color: ' + setts.appearanceStyles.default.normal.colour1 + '; ';
    //
    defAppNormal += 'background: -webkit-gradient(radial, center center, 0, center center, 141, from(' +
setts.appearanceStyles.default.normal.colour1 + '), to(' + setts.appearanceStyles.default.normal.colour2 + '));';
    /* old WebKit Syntax */
    myGradient = 'center center, circle farthest-side, ' + setts.appearanceStyles.default.normal.colour1 + ' 0%, ' +
setts.appearanceStyles.default.normal.colour2 + ' 200%';
    defAppNormal += 'background: -webkit-radial-gradient(' + myGradient + ');';
    /* New WebKit syntax */
    defAppNormal += 'background: -moz-radial-gradient(' + myGradient + ');';
    defAppNormal += 'background: -ms-radial-gradient(' + myGradient + ');';
    /* IE10+ */
    defAppNormal += 'background: -o-radial-gradient(' + myGradient + ');';
    /* Opera (13?) */
    defAppNormal += 'width: ' + setts.appearanceStyles.default.dimensions.width + 'px; ';
    defAppNormal += 'height: ' + setts.appearanceStyles.default.dimensions.height + 'px; ';
    defAppNormal += '}';
    var defAppHover = 'button:hover { ';
    defAppHover += 'background: radial-gradient(90% 300%, ' + setts.appearanceStyles.default.hover.colour1 + ', ' +
setts.appearanceStyles.default.hover.colour2 + '); ';
    //
    defAppHover += 'background: -webkit-gradient(radial, center center, 0, center center, 141, from(' +
setts.appearanceStyles.default.hover.colour1 + '), to(' + setts.appearanceStyles.default.hover.colour2 + '));';
    /* old WebKit Syntax */
    myGradient = 'center center, circle farthest-side, ' + setts.appearanceStyles.default.hover.colour1 + ' 0%, ' +
setts.appearanceStyles.default.hover.colour2 + ' 200%';
    defAppHover += 'background: -webkit-radial-gradient(' + myGradient + ');';
    /* New WebKit syntax */
    defAppHover += 'background: -moz-radial-gradient(' + myGradient + ');';
    defAppHover += 'background: -ms-radial-gradient(' + myGradient + ');';
    /* IE10+ */
    defAppHover += 'background: -o-radial-gradient(' + myGradient + ');';
    /* Opera (13?) */
    //
    defAppHover += 'cursor: pointer; ';
    defAppHover += '}';
    var defAppActive = 'button:active { ';

```

```

defAppActive += 'background: radial-gradient(90% 300%, ' + setts.appearanceStyles.default.active.colour1 + ', ' +
setts.appearanceStyles.default.active.colour2 + ');';
//
defAppActive += 'background: -webkit-gradient(radial, center center, 0, center center, 141, from(' +
setts.appearanceStyles.default.active.colour1 + '), to(' + setts.appearanceStyles.default.active.colour2 + '));';
/* old WebKit Syntax */
myGradient = 'center center, circle farthest-side, ' + setts.appearanceStyles.default.active.colour1 + ' 0%, ' +
setts.appearanceStyles.default.active.colour2 + ' 200%';
defAppActive += 'background: -webkit-radial-gradient(' + myGradient + ');';
/* New WebKit syntax */
defAppActive += 'background: -moz-radial-gradient(' + myGradient + ');';
defAppActive += 'background: -ms-radial-gradient(' + myGradient + ');';
/* IE10+ */
defAppActive += 'background: -o-radial-gradient(' + myGradient + ');';
/* Opera (13?) */
//
defAppActive += 'cursor: pointer;';
defAppActive += '}'';
var defAppDisabled = 'button[disabled] {';
defAppDisabled += 'background: radial-gradient(90% 300%, ' + setts.appearanceStyles.default.disabled.colour1 + ', ' +
setts.appearanceStyles.default.disabled.colour2 + ');';
//
defAppDisabled += 'background: -webkit-gradient(radial, center center, 0, center center, 141, from(' +
setts.appearanceStyles.default.disabled.colour1 + '), to(' + setts.appearanceStyles.default.disabled.colour2 + '));';
/* old WebKit Syntax */
myGradient = 'center center, circle farthest-side, ' + setts.appearanceStyles.default.disabled.colour1 + ' 0%, ' +
setts.appearanceStyles.default.disabled.colour2 + ' 200%';
defAppDisabled += 'background: -webkit-radial-gradient(' + myGradient + ');';
/* New WebKit syntax */
defAppDisabled += 'background: -moz-radial-gradient(' + myGradient + ');';
defAppDisabled += 'background: -ms-radial-gradient(' + myGradient + ');';
/* IE10+ */
defAppDisabled += 'background: -o-radial-gradient(' + myGradient + ');';
/* Opera (13?) */
//
defAppDisabled += 'cursor: default;';
defAppDisabled += '}'';
var defAppDisabledHover = 'button[disabled]:hover {';
defAppDisabledHover += 'background: radial-gradient(90% 300%, ' + setts.appearanceStyles.default.disabled.colour1 + ', ' +
+ setts.appearanceStyles.default.disabled.colour2 + ');';
//
defAppDisabledHover += 'background: -webkit-gradient(radial, center center, 0, center center, 141, from(' +
setts.appearanceStyles.default.disabled.colour1 + '), to(' + setts.appearanceStyles.default.disabled.colour2 + '));';
/* old WebKit Syntax */
myGradient = 'center center, circle farthest-side, ' + setts.appearanceStyles.default.disabled.colour1 + ' 0%, ' +
setts.appearanceStyles.default.disabled.colour2 + ' 200%';
defAppDisabledHover += 'background: -webkit-radial-gradient(' + myGradient + ');';
/* New WebKit syntax */

```

```

defAppDisabledHover += 'background: -moz-radial-gradient(' + myGradient + ');';
defAppDisabledHover += 'background: -ms-radial-gradient(' + myGradient + ');';
/* IE10+ */
defAppDisabledHover += 'background: -o-radial-gradient(' + myGradient + ');';
/* Opera (13?) */
//
defAppDisabledHover += 'cursor: default;';
defAppDisabledHover += '}}';
var xAppNormal = '';
var xAppHover = '';
var xAppActive = '';
var xAppDisabled = '';
var xAppDisabledHover = '';
var additionalRule = '';
var imagesArray = ['', '', '', '']
firstSheet.insertRule(defAppNormal, firstSheet.cssRules.length);
firstSheet.insertRule(defAppHover, firstSheet.cssRules.length);
firstSheet.insertRule(defAppActive, firstSheet.cssRules.length);
firstSheet.insertRule(defAppDisabled, firstSheet.cssRules.length);
firstSheet.insertRule(defAppDisabledHover, firstSheet.cssRules.length);
// firstSheet.insertRule(defAppColours, firstSheet.cssRules.length);
for (var name in setts.buttonLabels) {
    localProps.push(name);
}
for (var i = 0; i < Object.size(setts.buttonLabels); i++) {
    imagesArray = ['', '', '', '']
    try {
        document.getElementsByName(localProps[i])[0].value = setts.buttonLabels[localProps[i]];
    }
    catch (err) {}
    try {
        /*
        additional SIZE rule for specific buttons
        */
        additionalRule = '';
        if ((setts.appearanceStyles[localProps[i]].dimensions.width) &&
(setts.appearanceStyles[localProps[i]].dimensions.width != '')) {
            additionalRule = 'width: ' + setts.appearanceStyles[localProps[i]].dimensions.width + 'px;';
        }
        if ((setts.appearanceStyles[localProps[i]].dimensions.height) &&
(setts.appearanceStyles[localProps[i]].dimensions.height != '')) {
            additionalRule += ' height: ' + setts.appearanceStyles[localProps[i]].dimensions.height + 'px;';
        }
        if (additionalRule != '') {
            additionalRule = '.' + localProps[i] + 'AddRule { ' + additionalRule + '}}';
            firstSheet.insertRule(additionalRule, firstSheet.cssRules.length);
            document.getElementsByName(localProps[i])[0].className += ' ' + localProps[i] + 'AddRule';
        }
    }
}

```

```

/*
additional colour rules and images for specific buttons
*/
//NORMAL
if ((setts.appearanceStyles[localProps[i]].normal.colour1) &&
(setts.appearanceStyles[localProps[i]].normal.colour1 != "")) {
    xAppNormal = '.' + localProps[i] + 'Appearance { '
    xAppNormal += 'background: radial-gradient(90% 300%, ' +
setts.appearanceStyles[localProps[i]].normal.colour1 + ', ' + setts.appearanceStyles[localProps[i]].normal.colour2 + ');' +
' background-color: ' + setts.appearanceStyles.default.normal.colour1 + '; '
    //
    xAppNormal += 'background: -webkit-gradient(radial, center center, 0, center center, 141, from(' +
setts.appearanceStyles[localProps[i]].normal.colour1 + '), to(' + setts.appearanceStyles[localProps[i]].normal.colour2 +
')));';
    /* old WebKit Syntax */
    myGradient = 'center center, circle farthest-side, ' + setts.appearanceStyles[localProps[i]].normal.colour1
+ ' 0%, ' + setts.appearanceStyles[localProps[i]].normal.colour2 + ' 200%';
    xAppNormal += 'background: -webkit-radial-gradient(' + myGradient + ');';
    /* New WebKit syntax */
    xAppNormal += 'background: -moz-radial-gradient(' + myGradient + ');';
    xAppNormal += 'background: -ms-radial-gradient(' + myGradient + ');';
    /* IE10+ */
    xAppNormal += 'background: -o-radial-gradient(' + myGradient + ');';
    /* Opera (13?) */
    //
    xAppNormal += '}';
    firstSheet.insertRule(xAppNormal, firstSheet.cssRules.length);
    document.getElementsByName(localProps[i])[0].className += ' ' + localProps[i] + 'Appearance';
}
if ((setts.appearanceStyles[localProps[i]].normal.image) && (setts.appearanceStyles[localProps[i]].normal.image
!= "")) {
    imagesArray[0] = setts.appearanceStyles[localProps[i]].normal.image;
    document.getElementsByName(localProps[i])[0].innerHTML = '<span>' + setts.buttonLabels[localProps[i]] +
'</span>' + '';
}
else {
    document.getElementsByName(localProps[i])[0].firstChild.firstChild.nodeValue =
setts.buttonLabels[localProps[i]];
}
//HOVER
if ((setts.appearanceStyles[localProps[i]].hover.colour1) &&
(setts.appearanceStyles[localProps[i]].hover.colour1 != "")) {
    xAppHover = '.' + localProps[i] + 'Appearance: hover { '
    xAppHover += 'background: radial-gradient(90% 300%, ' + setts.appearanceStyles[localProps[i]].hover.colour1
+ ', ' + setts.appearanceStyles[localProps[i]].hover.colour2 + '); '
    //

```

```

        xAppHover += 'background: -webkit-gradient(radial, center center, 0, center center, 141, from(' +
setts.appearanceStyles[localProps[i]].hover.colour1 + '), to(' + setts.appearanceStyles[localProps[i]].hover.colour2 +
'));';

        /* old WebKit Syntax */
        myGradient = 'center center, circle farthest-side, ' + setts.appearanceStyles[localProps[i]].hover.colour1 +
' 0%, ' + setts.appearanceStyles[localProps[i]].hover.colour2 + ' 200%';
        xAppHover += 'background: -webkit-radial-gradient(' + myGradient + ');';
        /* New WebKit syntax */
        xAppHover += 'background: -moz-radial-gradient(' + myGradient + ');';
        xAppHover += 'background: -ms-radial-gradient(' + myGradient + ');';
        /* IE10+ */
        xAppHover += 'background: -o-radial-gradient(' + myGradient + ');';
        /* Opera (13?) */
        //
        xAppHover += 'cursor: pointer;';
        xAppHover += '}}';
        firstSheet.insertRule(xAppHover, firstSheet.cssRules.length);
    }
    if ((setts.appearanceStyles[localProps[i]].hover.image) && (setts.appearanceStyles[localProps[i]].hover.image !=
"")) {
        imagesArray[1] = setts.appearanceStyles[localProps[i]].hover.image;
        document.getElementsByName(localProps[i])[0].addEventListener("mouseover", function () {
            xMouseOver(this);
        });
        document.getElementsByName(localProps[i])[0].addEventListener("mouseout", function () {
            xMouseOut(this);
        });
    }
    //ACTIVE
    if ((setts.appearanceStyles[localProps[i]].active.colour1) &&
(setts.appearanceStyles[localProps[i]].active.colour1 != "")) {
        xAppActive = '.' + localProps[i] + 'Appearance:active { ';
        xAppActive += 'background: radial-gradient(90% 300%, ' +
setts.appearanceStyles[localProps[i]].active.colour1 + ', ' + setts.appearanceStyles[localProps[i]].active.colour2 + '); ';
        //
        xAppActive += 'background: -webkit-gradient(radial, center center, 0, center center, 141, from(' +
setts.appearanceStyles[localProps[i]].active.colour1 + '), to(' + setts.appearanceStyles[localProps[i]].active.colour2 +
'));';

        /* old WebKit Syntax */
        myGradient = 'center center, circle farthest-side, ' + setts.appearanceStyles[localProps[i]].active.colour1
+ ' 0%, ' + setts.appearanceStyles[localProps[i]].active.colour2 + ' 200%';
        xAppActive += 'background: -webkit-radial-gradient(' + myGradient + ');';
        /* New WebKit syntax */
        xAppActive += 'background: -moz-radial-gradient(' + myGradient + ');';
        xAppActive += 'background: -ms-radial-gradient(' + myGradient + ');';
        /* IE10+ */
        xAppActive += 'background: -o-radial-gradient(' + myGradient + ');';
        /* Opera (13?) */

```

```

//
xAppActive += 'cursor: pointer;';
xAppActive += '>';
firstSheet.insertRule(xAppActive, firstSheet.cssRules.length);
}
if ((setts.appearanceStyles[localProps[i]].active.image) && (setts.appearanceStyles[localProps[i]].active.image
!= "")) {
    imagesArray[2] = setts.appearanceStyles[localProps[i]].active.image;
    document.getElementsByName(localProps[i])[0].addEventListener("mousedown", function () {
        xMouseDown(this);
    });
    document.getElementsByName(localProps[i])[0].addEventListener("mouseup", function () {
        xMouseUp(this);
    });
}
//DISABLED
if ((setts.appearanceStyles[localProps[i]].disabled.colour1) &&
(setts.appearanceStyles[localProps[i]].disabled.colour1 != "")) {
    xAppDisabled = '.' + localProps[i] + 'Appearance[disabled] { ';
    xAppDisabled += 'background: radial-gradient(90% 300%, ' +
setts.appearanceStyles[localProps[i]].disabled.colour1 + ', ' + setts.appearanceStyles[localProps[i]].disabled.colour2 + ');';
    //
    xAppDisabled += 'background: -webkit-gradient(radial, center center, 0, center center, 141, from(' +
setts.appearanceStyles[localProps[i]].disabled.colour1 + '), to(' + setts.appearanceStyles[localProps[i]].disabled.colour2 +
')));';
    /* old WebKit Syntax */
    myGradient = 'center center, circle farthest-side, ' +
setts.appearanceStyles[localProps[i]].disabled.colour1 + ' 0%, ' + setts.appearanceStyles[localProps[i]].disabled.colour2 +
' 200%';
    xAppDisabled += 'background: -webkit-radial-gradient(' + myGradient + ');';
    /* New WebKit syntax */
    xAppDisabled += 'background: -moz-radial-gradient(' + myGradient + ');';
    xAppDisabled += 'background: -ms-radial-gradient(' + myGradient + ');';
    /* IE10+ */
    xAppDisabled += 'background: -o-radial-gradient(' + myGradient + ');';
    /* Opera (13?) */
    //
    xAppDisabled += 'cursor: default;';
    xAppDisabled += '>';
    firstSheet.insertRule(xAppDisabled, firstSheet.cssRules.length);
    xAppDisabledHover = '.' + localProps[i] + 'Appearance[disabled]:hover { ';
    xAppDisabledHover += 'background: radial-gradient(90% 300%, ' +
setts.appearanceStyles[localProps[i]].disabled.colour1 + ', ' + setts.appearanceStyles[localProps[i]].disabled.colour2 + ');';
    //

```

```

        xAppDisabledHover += 'background: -webkit-gradient(radial, center center, 0, center center, 141, from(' +
setts.appearanceStyles[localProps[i]].disabled.colour1 + '), to(' + setts.appearanceStyles[localProps[i]].disabled.colour2 +
'));';

        /* old WebKit Syntax */
        myGradient = 'center center, circle farthest-side, ' +
setts.appearanceStyles[localProps[i]].disabled.colour1 + ' 0%, ' + setts.appearanceStyles[localProps[i]].disabled.colour2 +
' 200%';

        xAppDisabledHover += 'background: -webkit-radial-gradient(' + myGradient + ');';
        /* New WebKit syntax */
        xAppDisabledHover += 'background: -moz-radial-gradient(' + myGradient + ');';
        xAppDisabledHover += 'background: -ms-radial-gradient(' + myGradient + ');';
        /* IE10+ */
        xAppDisabledHover += 'background: -o-radial-gradient(' + myGradient + ');';
        /* Opera (13?) */
        //
        xAppDisabledHover = 'cursor: default; ';
        xAppDisabledHover = '}}';
        firstSheet.insertRule(xAppDisabledHover, firstSheet.cssRules.length);
    }
    if ((setts.appearanceStyles[localProps[i]].disabled.image) &&
(setts.appearanceStyles[localProps[i]].disabled.image != "")) {
        imagesArray[3] = setts.appearanceStyles[localProps[i]].disabled.image;
    }
    //alert(imagesArray.join("@"))
    document.getElementsByName(localProps[i])[0].setAttribute("images", imagesArray.join("@"));
}
catch (err) {
    // if(window.console && console.error("Error:" + err));
}
}

var defAppColours = ".isOn.isActive { border: 5px solid " + setts.appearanceColours.thumbBrdr + "};";
firstSheet.insertRule(defAppColours, firstSheet.cssRules.length);
defAppColours = "span#filterby {color : " + setts.appearanceColours.mainTxt + "};";
firstSheet.insertRule(defAppColours, firstSheet.cssRules.length);
defAppColours = "input#counter {color : " + setts.appearanceColours.mainTxt + "};";
firstSheet.insertRule(defAppColours, firstSheet.cssRules.length);
defAppColours = "div#wrapper {background-color: " + setts.appearanceColours.mainBg + "}; ";
firstSheet.insertRule(defAppColours, firstSheet.cssRules.length);
defAppColours = "nav.thumbsMain {background-color: " + setts.appearanceColours.thumbBg + "}; ";
firstSheet.insertRule(defAppColours, firstSheet.cssRules.length);
defAppColours = "div.designFill {background-color: " + setts.appearanceColours.thumbBg + "}; ";
firstSheet.insertRule(defAppColours, firstSheet.cssRules.length);
firstSheet.insertRule('button#hide {background: url("") repeat scroll 0 0 transparent;}', firstSheet.cssRules.length);
document.getElementById("units").children[0].value = setts.buttonLabels.unitAll;
document.getElementById("units").children[0].text = setts.buttonLabels.unitAll;
document.getElementById("lessons").children[0].value = setts.buttonLabels.lessonAll;
document.getElementById("lessons").children[0].text = setts.buttonLabels.lessonAll;
return setts;

```



```

}

function keepCorrectButtons(setts) {
    var localProps = [];
    for (var i = 0; i < Object.size(setts.buttonsOn); i++) {
        try {
            removeClass(document.getElementsByName(setts.buttonsOn[i])[0], 'bempty');
        }
        catch (err) {}
    }
}

function hasClass(el, name) {
    try {
        return new RegExp('(\\s|^)' + name + '(\\s|$)').test(el.className);
    }
    catch (err) {
        return false;
    }
}

function removeClass(el, name) {
    if (hasClass(el, name)) {
        el.className = el.className.replace(new RegExp('(\\s|^)' + name + '(\\s|$)'), ' ').replace(/^\s+|\s+$/g, '');
    }
}

var QueryString = function () {
    // This function is anonymous, is executed immediately and
    // the return value is assigned to QueryString!
    var query_string = {};
    var query = window.location.search.substring(1).toLowerCase();
    var vars = query.split("&");
    for (var i = 0; i < vars.length; i++) {
        var pair = vars[i].split("=");
        // If first entry with this name
        if (typeof query_string[pair[0]] === "undefined") {
            query_string[pair[0]] = pair[1];
            // If second entry with this name
        }
        else if (typeof query_string[pair[0]] === "string") {
            var arr = [query_string[pair[0]],
                pair[1]
            ];
            query_string[pair[0]] = arr;
            // If third or later entry with this name
        }
        else {
            query_string[pair[0]].push(pair[1]);
        }
    }
}

```

```

    }
}
query_string.dummy = 0;
return query_string;
}
();
/**
DOM XML handling is cumbersome and illogical. We create a Javascript object with a specific
structure that represents the XML's specific structure. We then do not need to access the XML
object, but the internal easy to handle JS object. If there is a change in the XML structure
or the XML is not valid, this (and the whole application) will break.
**/
function createObjectFromXML(xmlObjRef) {
    var thelocObjRef = {};
    var totalUnits = 0;
    try {
        totalUnits = xmlObjRef.children.length;
    }
    catch (err) {
        //totalUnits = xmlObjRef.childNodes.length; explorer
        totalUnits = xmlObjRef.getElementsByTagName("unit").length; //explorer - chrome
    }
    _fcPlayer.translateLessons[0] = [];
    _fcPlayer.translateLessons[0].push("");
    for (var u = 0; u < totalUnits; u++) {
        thelocObjRef["Unit" + (u + 1)] = {};
        _fcPlayer.translateLessons[u + 1] = [];
        _fcPlayer.translateLessons[u + 1].push("");
        var theCurrentUnitNumberOfLessons = 0;
        try {
            theCurrentUnitNumberOfLessons = xmlObjRef.children[u].children.length;
        }
        catch (isNotFireFoxErr) {
            theCurrentUnitNumberOfLessons = xmlObjRef.getElementsByTagName("unit")[u].getElementsByTagName("lesson").length;
        }
        try {
            _fcPlayer.translateUnits.push(xmlObjRef.children[u].attributes["no"].value);
        }
        catch (isNotFireFoxErr) {
            _fcPlayer.translateUnits.push(xmlObjRef.getElementsByTagName("unit")[u].attributes[0].value);
        }
        for (var l = 0; l < theCurrentUnitNumberOfLessons; l++) {
            thelocObjRef["Unit" + (u + 1)][
                "Lesson" + (l + 1)
            ] = [];
            var theCurrentNumberOfItems = 0;
            try {
                theCurrentNumberOfItems = xmlObjRef.children[u].children[l].children.length;
            }

```

```

    }
    catch (isNotFireFoxErr) {
        theCurrentNumberOfItems =
xmlObjRef.getElementsByTagName("unit")[u].getElementsByTagName("lesson")[1].getElementsByTagName("item").length;
    }
    try {
        _fcPlayer.translateLessons[u + 1].push(xmlObjRef.children[u].children[1].attributes["no"].value);
    }
    catch (isNotFireFoxErr) {
        fcPlayer.translateLessons[u +
1].push(xmlObjRef.getElementsByTagName("unit")[u].getElementsByTagName("lesson")[1].attributes.getNamedItem("no").value);
    }
    for (var i = 0; i < theCurrentNumberOfItems; i++) {
        thelocObjRef["Unit" + (u + 1)][
            "Lesson" + (l + 1)
        ][i] = [];
        //make the internal arrays
        for (var j = 0; j < 3; j++) {
            thelocObjRef["Unit" + (u + 1)][
                "Lesson" + (l + 1)
            ][i][j] = [];
        }
        //
        xmlObjRef.children[u].children[1].children[0].children[0].attributes["face"].value =
sideValue - use this to determine sides
        //
        xmlObjRef.children[u].children[1].children[0].children = both sides of the items. use [0]
for index
        //
        xmlObjRef.children[u].children[1].children[0] = level of item, can get audio from here.
        //
        xmlObjRef.children[u].children[1].children[0].attributes["audio"].value = audio file for
both sides

        //each array has 2 elements, do the loop
        for (var g = 0; g < 2; g++) {
            //text content
            try {
                thelocObjRef["Unit" + (u + 1)][
                    "Lesson" + (l + 1)
                ][i][0].push(xmlObjRef.children[u].children[1].children[i].children[g].textContent);
            }
            catch (isNotFireFoxErr) {
                if
(xmlObjRef.getElementsByTagName("unit")[u].getElementsByTagName("lesson")[1].getElementsByTagName("item")[i].getElementsByTagName("side")[g].textContent) {
                    thelocObjRef["Unit" + (u + 1)][
                        "Lesson" + (l + 1)
                    ][i][0].push(xmlObjRef.getElementsByTagName("unit")[u].getElementsByTagName("lesson")[1].getElementsByTagName("item")[i].getElementsByTagName("side")[g].textContent);
                }
                else {

```

```

        thelocObjRef["Unit" + (u + 1)][
            "Lesson" + (l + 1)

] [i] [0].push(xmlObjRef.getElementsByTagName("unit") [u].getElementsByTagName("lesson") [l].getElementsByTagName("item") [i].get
ElementsByTagName("side") [g].text);
    }
}
//node type
try {
    thelocObjRef["Unit" + (u + 1)][
        "Lesson" + (l + 1)
    ] [i] [2].push(xmlObjRef.children[u].children[l].children[i].children[g].attributes["daType"].value);
}
catch (isNotFireFoxErr) {
    thelocObjRef["Unit" + (u + 1)][
        "Lesson" + (l + 1)

] [i] [2].push(xmlObjRef.getElementsByTagName("unit") [u].getElementsByTagName("lesson") [l].getElementsByTagName("item") [g].get
ElementsByTagName("side") [g].attributes.getNamedItem("daType").value);
    }
}
var locEnvIsNotFirefox = false;
try {
    xmlObjRef.children[u];
}
catch (IsNotFireFox) {
    locEnvIsNotFirefox = true;
}
if (!locEnvIsNotFirefox) { //if this exists, we are running on firefox. kept the audio separately - it is
not possible to know whether a property is null
    //for it is not supported in the environment, or because it has not been included in the XML
    //audio - lets do for FF
    try {
        //try to see if there is an audio element above. if there is, keep that and overwrite any elements
included in the sides
        thelocObjRef["Unit" + (u + 1)][
            "Lesson" + (l + 1)
        ] [i] [1].push(xmlObjRef.children[u].children[l].children[i].attributes["audio"].value); //audio
attribute - try for single audio in item
    }
    catch (eitherAudioDoesNotExistOrPropertyIsNotSupported) {
        //there is no property for both sides. lets try to see if there is one for each side
        try {
            //check side 1
            thelocObjRef["Unit" + (u + 1)][
                "Lesson" + (l + 1)

] [i] [1].push(xmlObjRef.children[u].children[l].children[i].children[0].attributes["audio"].value);

```

```

    }
    catch (eitherAudioDoesNotExistOrPropertyIsNotSupported) {
        //we should should try for chrome/ie. if there is another exception here, that means definitely
no audio for side 1. now do the same for side 2
    }
    try {
        thelocObjRef["Unit" + (u + 1)][
            "Lesson" + (l + 1)

] [i] [1].push(xmlObjRef.children[u].children[l].children[i].children[1].attributes["audio"].value); //audio attribute for
side 2
    }
    catch (eitherAudioDoesNotExistOrPropertyIsNotSupported) {
        //we should should try for chrome/ie. if there is another exception here, that means definitely
no audio for side 2 also.
    }
}
}
else {
    //audio - lets do for chrome etc :)
    try {
        //try to see if there is an audio element above. if there is, keep that and overwrite any elements
included in the sides
        thelocObjRef["Unit" + (u + 1)][
            "Lesson" + (l + 1)

] [i] [1].push(xmlObjRef.getElementsByTagName("unit") [u].getElementsByTagName("lesson") [l].getElementsByTagName("item") [i].att
ributes.getNamedItem("audio").value); //audio attribute - try for single audio in item
    }
    catch (eitherAudioDoesNotExistOrPropertyIsNotSupported) {
        //there is no property for both sides. lets try to see if there is one for each side
        try {
            //check side 1
            thelocObjRef["Unit" + (u + 1)][
                "Lesson" + (l + 1)

] [i] [1].push(xmlObjRef.getElementsByTagName("unit") [u].getElementsByTagName("lesson") [l].getElementsByTagName("item") [i].get
ElementsByTagName("side") [0].attributes.getNamedItem("audio").value);
        }
        catch (eitherAudioDoesNotExistOrPropertyIsNotSupported) {
            //we should should try for chrome/ie. if there is another exception here, that means definitely
no audio for side 1. now do the same for side 2
        }
        try {
            thelocObjRef["Unit" + (u + 1)][
                "Lesson" + (l + 1)

```

```

] [i] [1].push(xmlObjRef.getElementsByTagName("unit")[u].getElementsByTagName("lesson")[l].getElementsByTagName("item")[i].getElementsByTagName("side")[1].attributes.getNamedItem("audio").value); //audio attribute for side 2
    }
    catch (eitherAudioDoesNotExistOrPropertyIsNotSupported) {
        //we should should try for chrome/ie. if there is another exception here, that means definitely
no audio for side 2 also.
    }
    }
    }
    //put data in the arrays, but also determine the type etc..
}
}
}
return thelocObjRef;
}

function GetFilename(url) {
    if (url) {
        var m = url.toString().match(/.*\/(.+)\.\/);
        if (m && m.length > 1) {
            return m[1];
        }
    }
    return "";
}

function getURLParameter(name) {
    return decodeURIComponent((new RegExp('[?|&]' + name + '=' + '([^&]+?)(&|#|;|$)').exec(location.search) || [,
    ""])[1].replace(/\+/g, '%20')) || null
}

Object.size = function (obj) {
    var size = 0,
        key;
    for (key in obj) {
        if (obj.hasOwnProperty(key)) size++;
    }
    return size;
};

Element.prototype.indexOf = function (el) {
    var nodeList = this.childNodes;
    var array = [].slice.call(nodeList, 0);
    return array.indexOf(el);
}

function insertAfter(newElement, targetElement) {
    //target is what you want it to go after. Look for this elements parent.
    var parent = targetElement.parentNode;

```

```

    //if the parents lastchild is the targetElement...
    if (parent.lastchild == targetElement) {
        //add the newElement after the target element.
        parent.appendChild(newElement);
    }
    else {
        // else the target has siblings, insert the new element between the target and it's next sibling.
        parent.insertBefore(newElement, targetElement.nextSibling);
    }
}

function hideLessonItems(theValue) {
    var ls = document.getElementsByTagName("li");
    if (theValue != 0) {
        for (var i = 0; i < ls.length; i++) {
            var theIdWeCare = extractLocation(ls[i].id).Lesson;
            var theOtherIdWeCare = extractLocation(ls[i].id).Unit;
            if ((theIdWeCare == theValue) && (theOtherIdWeCare == _fcPlayer.curentCardOnDisplay.Unit)) {
                document.getElementById(ls[i].id).className = "isOn";
            }
            else {
                document.getElementById(ls[i].id).className = "isOff";
            }
        }
    }
    else {
        hideItems(_fcPlayer.curentCardOnDisplay.Unit);
    }
}

function hideItems(theValue) {
    if (theValue != 0) {
        //issue#535507000004761031
        $("li[id^=" + theValue + "_]").removeClass("isOff").addClass("isOn");
        $("li:not([id^=" + theValue + "_])").removeClass("isOn").addClass("isOff");
    }
    else {
        $("li").removeClass("isOff").addClass("isOn");
    }
}

/**
Build the dropdowns for the units
**/
function createUnitDropDowns() {
    var theHtml = "";
    for (var u = 0; u < Object.size(_fcPlayer.allCards); u++) {
        theHtml = theHtml + '<option value="' + _fcPlayer.globalSettings.buttonLabels["unit"] + String(Number(u + 1)) + "'>'
+ _fcPlayer.globalSettings.buttonLabels["unit"] + " " + _fcPlayer.translateUnits[u + 1] + '</option>';
    }
}

```

```

    }
    return theHtml;
}
/**
Build the dropdowns for the lessons
**/
function createLessonDropDowns(theUnit) {
    var theHtml = '<option value="' + _fcPlayer.globalSettings.buttonLabels.lessonAll + '">' +
    _fcPlayer.globalSettings.buttonLabels.lessonAll + '</option>';
    try {
        for (var l = 0; l < Object.size(_fcPlayer.allCards["Unit" + theUnit]); l++) {
            theHtml = theHtml + '<option value="' + _fcPlayer.globalSettings.buttonLabels.lesson + String(Number(l + 1)) +
            '">' + _fcPlayer.globalSettings.buttonLabels.lesson + " " + _fcPlayer.translateLessons[theUnit]
            [l + 1] + '</option>';
        }
    }
    catch (e) {
        theHtml = '<option value="' + _fcPlayer.globalSettings.buttonLabels.lesson + '">' +
        _fcPlayer.globalSettings.buttonLabels.lesson + '</option>'; //LANGUAGE SPECIFIC
    }
    if (theHtml == "") {
        theHtml = '<option value="' + _fcPlayer.globalSettings.buttonLabels.lesson + '">' +
        _fcPlayer.globalSettings.buttonLabels.lesson + '</option>';
    } //LANGUAGE SPECIFIC
    return theHtml;
}
/**
Keep the internal sequential order, but associate with the actual unit number
**/
function transliterateUnits(theActualUnit) {
    var theReferencedUnit = _fcPlayer.translateUnits.indexOf(theActualUnit);
    if (theReferencedUnit < 0) {
        if (settings.allowDebugMessages) {
            alert("Wrong unit number as a paramater, selecting first unit as default.");
        }
        theReferencedUnit = 1;
    }
    return theReferencedUnit;
}
/**
Keep the internal sequential order, but associate with the actual lesson number
**/
function transliterateLesson(theActualLesson, theUnit) {
    var theReferencedLesson = _fcPlayer.translateLessons[theUnit].indexOf(theActualLesson);
    if (theReferencedLesson < 0) {
        if (settings.allowDebugMessages) {
            alert("Wrong lesson number as a paramater, selecting first lesson as default.");
        }
    }
}

```



```

        theReferencedLesson = 1;
    }
    return theReferencedLesson;
}
/**
If the player was launched with parameters, set the appropriate filters automatically
**/
function filterByURL() {
    var locUnitReference = QueryString.u;
    var transliteratedLesson = QueryString.l;
    if ((transliteratedLesson) && (!locUnitReference)) {
        locUnitReference = 1;
        QueryString.u = locUnitReference;
        if (settings.allowDebugMessages) {
            alert("Please check link. A lesson has to have a unit.\nFirst available unit will be selected by default");
        }
    }
    else {
        locUnitReference = transliterateUnits(locUnitReference);
    }
    _fcPlayer.curentCardOnDisplay.Unit = locUnitReference;
    _fcPlayer.curentCardOnDisplay.Card = 0;
    switch (Object.size(QueryString)) {
    case 3:
        document.getElementById("units").selectedIndex = locUnitReference;
        transliteratedLesson = transliterateLesson(transliteratedLesson, locUnitReference);
        _fcPlayer.curentCardOnDisplay.Lesson = transliteratedLesson;
        (function () {
            document.getElementById("lessons").selectedIndex = transliteratedLesson;
            hideLessonItems(transliteratedLesson);
        }) ($("#lessons").html(createLessonDropDowns(locUnitReference)));
        break;
    case 2:
        document.getElementById("units").selectedIndex = locUnitReference;
        $("#lessons").html(createLessonDropDowns(locUnitReference));
        hideItems(locUnitReference);
        break;
    default:
    }
}
/**
Builds the WHOLE thumbnails structure.
TODO: Change the fixed size for thumbs and images with something dynamic
**/
function addCurrentStateProperty(theObj) {
    var theHTML = "";
    //CONSTRUCT:
    // _fcPlayer.allCards.Unit[X].Lesson[X].item[K] [0 - DEFAULT FACE] [FLIP SIDE] [CURRENTSIDE]

```

```

//      _fcPlayer.allCards["Unit"+"1"] ["Lesson"+"1"] [K] [0] [0] - original side
//      _fcPlayer.allCards["Unit"+"1"] ["Lesson"+"1"] [K] [0] [1] - flip side
//      _fcPlayer.allCards["Unit"+"1"] ["Lesson"+"1"] [K] [0] [2] = o deikths tou cuurent card
//      _fcPlayer.allCards["Unit"+"1"] ["Lesson"+"1"] [0] [1] - audio
for (var u = 1; u <= Object.size(_fcPlayer.allCards); u++) {
    for (var l = 1; l <= Object.size(_fcPlayer.allCards["Unit" + u]); l++) {
        for (var k = 0; k < Object.size(_fcPlayer.allCards["Unit" + u][
            "Lesson" + l
        ]); k++) {
            var l1 = _fcPlayer.allCards["Unit" + u]["Lesson" + l][k][2][0];
            if (l1 == "t") {
                /*RESIZE TEXT - FULL*/
                document.getElementById("txt_test").style.fontSize = "24px";
                var myResults = calculateTextSize("txt_test", _fcPlayer.allCards["Unit" + u]["Lesson" + l][k][0]
                    [0], 175, 136, 0.1)
                var myAlignement = "center";
                if (myResults[0] > 70) {
                    myAlignement = "left"
                }
                document.getElementById("txt_test").value = _fcPlayer.allCards["Unit" + u]
                    ["Lesson" + l][k][0][
                        0
                    ];
                theHTML = theHTML + '<li class="isOn" id="' + String(Number(u)) + '_' + String(Number(l)) + '_' +
String(Number(k)) + '">' + '<span class="textThumb" style="' + 'height:' + myResults[0] + 'px;
font-size: ' + myResults[1] + '; text-align: ' + myAlignement + '" >' + _fcPlayer.allCards["Unit" + u][
    "Lesson" + l][k][0][0] + '</span></li>';
            }
            else if (l1 == "i") {
                theHTML = theHTML + '<li class="isOn" id="' + String(Number(u)) + '_' + String(Number(l)) + '_' +
String(Number(k)) + '">' + '<span class="empty">' + "" + '</span></li>';
            }
        }
    }
}
}
$("#units").append(createUnitDropDowns());
$("#left > ul").html(theHTML);
if (window.location.search != "") {
    filterByURL();
}
else {
    document.getElementById("lessons").selectedIndex = 0;
}
}
/**
This fires when the XML was loaded, thus begining the interaction

```

```

**/
function makePageChanges(theContent) {
    _fcPlayer.allCards = createObjectFromXML(theContent);
    addCurrentStateProperty(theContent);
}
**/
Begin to read the XML - W3C recommended method to read XML files
**/
function importXML(url) {
    var obj = {};
    var xmlDoc = {};
    try {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.open("GET", url, false);
    }
    catch (Exception) {
        var ie = (typeof window.ActiveXObject != 'undefined');
        if (ie) {
            xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
            xmlDoc.async = false;
            while (xmlDoc.readyState != 4) {};
            xmlDoc.load(url);
            obj = xmlDoc.documentElement;
            makePageChanges(obj);
            xmlloaded = true;
        }
        else {
            xmlDoc = document.implementation.createDocument("", "", null);
            obj = xmlDoc.documentElement;
            xmlDoc.onload = makePageChanges(obj);
            xmlDoc.load(url);
            xmlloaded = true;
        }
    }
    if (!xmlloaded) {
        xmlhttp.setRequestHeader('Content-Type', 'text/xml')
        xmlhttp.send("");
        xmlDoc = xmlhttp.responseXML;
        obj = xmlDoc.documentElement;
        makePageChanges(obj);
        xmlloaded = true;
    }
}

```