

Web Editör Kodları Açıklama Rehberi

```
function getUserCode() {  
    return htmlEditor.getValue() + "\n" + "<style>" + "\n" + cssEditor.getValue() + "\n" +  
    "</style>" + "\n" + "<script>" + "\n" + jsEditor.getValue() + "\n" + "</script>";  
}
```

Kullanıcının eklediği bilgilerin render edilmesi için bir araya getirdim

```
function update() {  
    //iframe alanı  
    var code = document.getElementById('iframe').contentWindow.document;  
    code.open();  
    //editörden gelen değeri iframe içeriğine ekleniyor.  
    code.write(getUserCode());  
    code.close();  
  
    document.getElementById("jsCode").value = jsEditor.getValue();  
    document.getElementById("htmlCode").value = htmlEditor.getValue();  
    document.getElementById("cssCode").value = cssEditor.getValue();  
}
```

I FRAME dataların yuklenmesini sağlar. Bu sayede js , html , css kodlarımız bir bütün olarak iframe üzerinde gösterilmiş olur.

```
function loadHTMLEditor() {  
    defaultHTMLValue =  
    "<!DOCTYPE html>\n\n<html>\n\n    <!-- Buraya Yazdığınız HTML Kod Otomatik  
derlenecektir -->\n\n</html>"  
    window.htmlEditor = ace.edit("htmlEditor");  
    htmlEditor.setTheme("ace/theme/dracula");  
    htmlEditor.getSession().setMode("ace/mode/html");  
    htmlEditor.setValue(defaultHTMLValue,1);  
    htmlEditor.getSession().on('change', function() {  
        update();  
    });  
}
```

loadhtmlEditor , loadcssEditor, loadJsEditor olmak üzere üç adet editor functionumuz var. Bu funtionlar ace editor framework kullanarak html,css,js kod geliştirebileceğimiz ortam oluşturur.

```
htmlEditor.focus();
htmlEditor.setOptions({
    fontSize: "12.5pt",
    showLineNumbers: true,
    vScrollBarAlwaysVisible:false,
    enableBasicAutocompletion: true,
    enableSnippets: true,
    enableLiveAutocompletion: false
});

htmlEditor.setShowPrintMargin(false);
}
function loadCSSEditor() {
    defaultCSSValue = "/*Buraya yazdığınız css code otomatik derlenecektir*/"

    window.cssEditor = ace.edit("cssEditor");
    cssEditor.resize();
    cssEditor.renderer.updateFull();
    cssEditor.setTheme("ace/theme/dracula");
    cssEditor.getSession().setMode("ace/mode/css");
    cssEditor.setValue(defaultCSSValue,1);
```

```
cssEditor.getSession().on('change', function() {
    update();
    cacheSet();
});
```

Herhangi bir değişiklikte update ve cache fonksiyonları çalışıyor. Bu fonksiyonlar aracılığı ile kod değişikliği anlık olarak iframe üzerinde görüntülenebiliyor. Hidden inputlar setleniyor. Session üzerine datalar yazılıyor.Bu sayede sayfa anlık kapansa bile session üzerinden gelen datalarla hayatına devam ediyor. Bu session üzerine eklenen datalar ise sayfa ready() fonksiyonunda tetikleniyor.

```
cssEditor.focus();
cssEditor.setOptions({
    fontSize: "12.5pt",
    showLineNumbers: true,
    vScrollBarAlwaysVisible:true,
    enableBasicAutocompletion: true,
```

```

        enableSnippets: true,
        enableLiveAutocompletion: false
    });
    cssEditor.setShowPrintMargin(false);
}

function loadJSEditor() {
    defaultJSValue = "/*Buraya yazdığınız JS code otomatik derlenecektir */"

    window.jsEditor = ace.edit("jsEditor");
    jsEditor.setTheme("ace/theme/dracula");
    jsEditor.getSession().setMode("ace/mode/javascript");
    jsEditor.setValue(defaultJSValue,1);
    jsEditor.getSession().on('change', function() {
        update();
    });
    jsEditor.focus();
    jsEditor.setOptions({
        fontSize: "12.5pt",
        showLineNumbers: true,
        vScrollBarAlwaysVisible:true,
        enableBasicAutocompletion: true,
        enableSnippets: true,
        enableLiveAutocompletion: false
    });
    jsEditor.setShowPrintMargin(false);
}

function setupEditor() {
    loadHTMLEditor();
    loadCSSEditor();
    loadJSEditor();
}

function ready() {
    setupEditor();
    storageToSelectBox();
    getCacheSessionData();
}

function minEditor() {
    let iframe = document.getElementById("iframe");
    iframe.style.height = "98%";
    iframe.style.width = "100%";
    let htmlEditor = document.getElementById("htmlEditor");
    htmlEditor.style.height = "%";
    htmlEditor.style.width = "0%";
    let cssEditor = document.getElementById("cssEditor");
    cssEditor.style.height = "0%";

```

```

cssEditor.style.width = "0%";
let jsEditor = document.getElementById("jsEditor");
jsEditor.style.height = "0%";
jsEditor.style.width = "0%";
let allEditors = document.getElementById("editors");
allEditors.style.height = "5%";
allEditors.style.width = "100%";
}
function maxEditor() {
    let editors = document.getElementById("editors");
    editors.style.height = "50%";
    editors.style.width = "100%";
    let htmlEditor = document.getElementById("htmlEditor");
    htmlEditor.style.height = "90%";
    htmlEditor.style.width = "32%";
    let cssEditor = document.getElementById("cssEditor");
    cssEditor.style.height = "90%";
    cssEditor.style.width = "32%";
    let jsEditor = document.getElementById("jsEditor");
    jsEditor.style.height = "90%";
    jsEditor.style.width = "32%";
    var iframe = document.getElementById("iframe");
    iframe.style.height = "50%";
    iframe.style.width = "100%";
}

function downloadCode() {
    const userCode = getUserCode();
    const blob = new Blob([userCode], {type: "text/html"});
    downloadFile(blob,"index.html");
}
function downloadFile(blob,fileName) {
    const url = window.URL.createObjectURL(blob);
    const a = document.createElement('a');
    a.href = url;
    a.download = fileName;
    a.click();
    a.remove();

    document.addEventListener("focus",w=>{window.URL.revokeObjectURL(url)})
}

```

Bu fonksiyon sayesinde yazdığımız kodlar sayfa download edilebiliyor

```
function saveCode(){
    var jsCode=document.getElementById("jsCode").value;
    var htmlCode=document.getElementById("htmlCode").value;
    var cssCode=document.getElementById("cssCode").value;
    var input;
    var jsonCode;
    input = prompt('Template Adını Giriniz');
    if (input === null) {
        alert("Ad Alanı Boş Olamaz.");
    }
    else if (input) {
        alert("Kayıt Ekleme Başarılı!");
        //json formatına cevirdim input datalarını
        jsonCode=JSON.stringify({ 'js': jsCode,
            'html': htmlCode ,'css':cssCode });
        //localStorage içerisine setledim. template_ attım başına seçmemiz kolay olsun diye.
        localStorage.setItem('template_'+input, jsonCode);
    }
}
```

Burası kaydet butonu ile tetikleniyor. Kaydet butonuna bastığımızda alan boş mu kontrol ediyorum. Daha sonra eğer boş değilse mevcut dataları json formatında key'in başında “template_” tagı olacak şekilde setliyorum.LocalStorage’de bulunan diğer datalar ile karışmaması ve kendine özgü set , get , list işlemlerinde daha rahat olurum diye başına tag ekledim.

```
function getCode(key){
    var JsonItem;
    JsonItem =localStorage.getItem('template_'+key);
    setFields(JsonItem);
}
```

Burada local storage üzerine kaydettiğim dataları getirdim ve ardından setField ile ilgili inputlara yolladım.

```
function getCacheSessionData(key){
    var JsonItem;
    JsonItem=sessionStorage.getItem('template_cache');
    setFields(JsonItem);
}
```

```
}
```

Burada sessionStorage üzerine kaydettiğim dataları getirdim , bu sayfa yüklendiğinde çalışıyor, eğer sessionStorage üzerinde data varsa (session storage sürekli olarak eventchanges ile setleniyor) bu dataları ilgili inputlara koyuyor. Bu sayede anlık kapanmalarda data kaybının vs. Önüne geçtim.

```
function setFields(JsonItem){
    const obj = JSON.parse(JsonItem);

    cssEditor.setValue(obj.css,1);
    jsEditor.setValue(obj.js,1);
    console.log(obj);
    //Kodu getirdikten sonra beatify ederek düzgün görünmesini sağladım.
    let beautify = window.ace.require('ace/ext/beautify');
    beautify.beautify(jsEditor.session);
    beautify.beautify(cssEditor.session);
    beautifyHtmlCodes = html_beautify(obj.html);
    htmlEditor.setValue(beautifyHtmlCodes,1);
    htmlEditor.getSession().setMode("ace/mode/html");
}
```

Burada ise local storage ya da session üzerinden gelen dataları editorlere setliyorum. Beautify ederek kodların düz text değilde , biçimlendirilmiş şekilde basılmasını sağladım.

```
function storageToSelectBox(){
    var onlyItemName;

    var select=select = document.getElementById('templateSelectBox');
    for (var key in localStorage){
        //data seçmem kolay olsun diye template_ ile kaydetmişim.Çünkü local storage da
        //sadece benim datalarım yok.Kullanıcı bunu görmesin diye geri listeye alırken sildim bu tagı.
        if(key.substring(0, 9)=== 'template_'){
            console.log(key);
            onlyItemName=key.substring(9);
            var opt = document.createElement('option');
            opt.value = onlyItemName;
            opt.innerHTML = onlyItemName;
            select.appendChild(opt);
        }
    }
}
```

Burada localstroge uzzerinden çektiğim storage keyleri checkbox'a koydum.Kullanıcı 'template_' dışında sadece yazdığı key'i görsün diye checkbox'a eklerken 'template_' keyini kaldırdım.

```
function cacheSet(){

    var jsCode=document.getElementById("jsCode").value;
    var htmlCode=document.getElementById("htmlCode").value;
    var cssCode=document.getElementById("cssCode").value;

    var jsonCode=JSON.stringify({ 'js': jsCode,
        'html': htmlCode , 'css':cssCode });

    sessionStorage.setItem('template_cache', jsonCode);
    console.log("CACHE SERVICE ÇALISTI : " + jsonCode);
}
```

Burada session üzerinden datalarımı ani kapanmalarda vs. kaybolmaması için tuttum.

```
function newTemplate(){  
  if (confirm('Yeni Çalışma Oluşturmak İstiyor musunuz ? Mevcut Çalışmalarınız Yok  
Olacak.')) {  
  
    sessionStorage.clear();  
    setupEditor();  
  
  } else {  
    console.log('Yes');  
  }  
  
}
```

Burada session temizleyerek yeni bir template oluşturma fırsatı sundum.

ZAMAN AYIRIP KODLARIMI INCELEDİĞİNİZ İÇİN TESEKKUR EDERİM...

-Osman ARAT