

# Dinamik HTML elementleri eklemek ve çıkartmak

## Yapıyı Anlatan Rapor

Yukarıda yazmış olduğum yapıları size kısa kısa anlatmaya çalışacağım burada.Öncelikle bir form yapısı oluşturdum inputları koydum ve tiplerini text olarak alıp birer buton yerleştirdim.Sonra bu butonları javascript eventleri olan onclick metodunu yerleştirdim.Buraya kadar her şey güzel.Bundan sonrası için DOM yapılarını kullandık.

DOM yapıları javascriptin olmazsa olmazı dinamik bir yapıdır ve text içerisine yazdığım metinleri formun altına ekledim ve aynı zamanda hangi tagları eklediğimde belirttim 2 tane farklı tag üzerinden göstermeye çalıştım.Ama diğer tagları da ekleyebilirsiniz ben yapı olarak aynı olduğu için 2 taneyi yeterli buldum.Şimdi biraz DOM yapılarına bakalım.Girmiş olduğum text yerini ilk önce script tagları içerisinde metin adlı bir değişkene atadım.Bu textin içindeki değeri o anki bulunduğu yerin id'sinin value değerini alarak yaptım.Sonra bu metini hangi taga ekleyeceğimi seçmek için yeni bir tag oluşturdum.Bunu da createElement metoduyla mesela bir p tagı oluşturmak istediğimi beyan ettim.Sonra bu girmiş olduğum metini bir textNode olduğunu belirtmek istedim çünkü bu metin ile p tagını bağdaştırmam gerekecekti.

Aslında burada şöyle bir şeye değinmek istiyorum burda atıyorum 4 tane elementimiz olsun ama bize javascript çıktısı ekrana 9 tane element olduğunu söylecek yani 9 tane childNode olduğunu belirtecek.Bu yapıları aslında konsoldan proto bölümünden görebilirsiniz.Yani bir tane tag oluşturuyoruz ama bu tagın içerisini ayrı bir çocuk oluşturuyor.Yani p tagı bir çocuk içindeki text ise ayrı bir çocuk elementini oluşturmaktadır.

Şimdi geri dönelim yapımızı anlatmaya.Bir textNode oluşturup bunu p tagıyla appendChild metoduyla bağdaştırıyoruz yani birleştiriyoruz. Yani p tagını createElement adlı değişkene atamıştık bu değişkeni de createElement.appendChild(text) diyerek ikisini birleştirdik. Sonra bu p tagını p tagları buraya eklenecek olan yerin altına eklemek için orayada appendChild metodunu kullaarak altına eklemiş olduk.

Gelelim silme işlemine.Silme işlemi eklemeye göre çok daha basit.İlk önce mesela p tagını eklediğimiz yere yeni bir createElement ile buton oluşturuyoruz.Bu butona bazı özellikler veriyoruz setAttribute ile sonra butona bir onclick özelliği veriyoruz yani butona tıkladığında silme işlemi gerçekleşmesi için.Sonra bu

fonksiyonun içine nereyi tıkladıysa o çocuğun parentini silmesi gerekecek yeri söylüyoruz.Bunuda şöyle yapıyoruz :

Fonksiyona bir e parametresi atadık.Bu parametre ile target adlı bir özelliğe eriştik.Bu target nereye tıkladıysak orayı bize verecek.Mesela butona tıkladıysak o butonun yerini bize vermiş olacak.Sonra bunun ParentElementine ulaştık target ile sonra removeChild fonksiyonuyla ve bu fonksiyonun içerisine target ile bulunduğumuz yerin parentNodu'na ulaştık ve bu sayede ekledimiz o patanın hem p tagına hem de içindeki text özelliğine ulaşp Node'ları kaldırmış olduk.

**e.target.parentNode.parentNode.removeChild(e.target.parentNode);**

Silme işlemini bu kod parçası ile yapabiliriz.

## Alışveriş Listesi Örneği

### Yapıyı Anlatan Rapor

Alışveriş listesi örneğinde aslında bu ilk yukarda anlattığım yapıdan hiçbir farkı yok tamamen aynı.Genede birkaç şeyden bahsedelim sizlere.İlk önce css yapılarını oluşturduk sonra bunları yapıya dökmeye karar verdik.Yapıya şöyle bir bakarsak ilk önce bir link oluşturduk .

**const li = document.createElement('li');**

Bu kod parçasıyla oluşturuyoruz.Sonra altına a elementlerini vereceiz.Tabi ondan önce bu li'lerin textNodelerini oluşturuyoruz.

**li.appendChild(document.createTextNode(input.value));**

Bu kod ile de textNode'leri altına eklemiş olduk.

Sonra a taglarını verdik altına link olduğu için.

**const a = document.createElement('a'); Bu şekilde.**

Sonra gerekli css.class özellikleri verdikten sonra bu link ile a tagını birbirine bağladık.

**li.appendChild(a); Bu şekilde olduğu gibi.**

Bu metod ile de oluşturduğumuz linki listemize kattık.

**liste.appendChild(li);**

Eklediğimiz bu taglarından istediğimizi kaldırmak için ;

**e.target.parentElement.parentElement.removeChild(e.target.parentElement); bu metodu kullandık.**

Tüm elementleri yani listeyi temizlemek için ise ;

**liste.innerHTML=""; Bu metodu kullanmamız yeterli oldu.**

Her sayfa yenilendiğinde scrolbar'ı sabit tutabilmek için

**e.preventDefault(); Bu metodu kullandık.**

## Kullanıcı Formu

### Yapıyı Anlatan Rapor

```
<div class="form-group clearfix">
```

```
<div class="col-25">
```

```
<label > TC Kimlik No: </label></div>
```

```
<div class="col-75">
```

```
<input type="text" name="tc" class="form-control" id="tc"
autocomplete="off">
```

```
<span id="tcids"> </span>
```

```
</div>
```

```
</div>
```

Yukarıda örneğini verdiğim gibi her oluşum bir textboxları veya diğer elementleri yerleştiriyoruz hepsinin yapısı bu şekildedir. Gerekli css özelliklerini vermek için class yapılarını ekledik.

```
<input type="text" name="tc" class="form-control" id="tc"
autocomplete="off">
```

Burada id değerini vererek script kodunda document.getElementById() metodu ile bu textin içindeki değere ulaşmış olacağız sonra bu değer ile gerekli kontrolleri yapacağız.

```
<span id="tcids"> </span>
```

Buraya ise eğer bir hata dönerse bu hatanın değerini buraya yazmasını sağlaması için kullandık.İçindeki id değeri ile buraya ulaşmış olacağız.

Aşağı yukarı tüm textler bu şekilde aynı css ve benzer id tip mantığıyla verilmiştir.

Şimdi biraz da script kodlarına bakalım beraber...

```
<form onsubmit="return dogrula()" >
```

Formu oluştururken butona tıklanırken onsubmit metoduyla dogrula adında bir fonksiyona erişmiş olacağız ve tüm kontrolleri bu dogrula() fonksiyonun içinde gerçekleştireceğiz.

Dogrula fonksiyonuna girdikten sonra ;

```
var user = document.getElementById('user').value;
```

```
var pass = document.getElementById('pass').value;
```

```
var confirmpass = document.getElementById('conpass').value;
```

```
var mobileNumber = document.getElementById('mobileNumber').value;
```

```
var emails = document.getElementById('emails').value;
```

```
var tc =document.getElementById('tc').value;
```

```
var il =document.getElementById('il').value;
```

tek tek yukarda html kısmında her textleri oluştururken verdiğimiz id değerlerini document.getElementById().value metoduyla bu textlerin içindeki değerlere ulaşmış olduk.

```
if(user == ""){
```

```
document.getElementById('username').innerHTML = " Lütfen kullanıcı  
adını boş geçmeyiniz.";
```

```
return false;
```

```
}
```

Örnek olarak şu kontrol yapısını incelersek id'sini user verdiğimiz textin içindeki değer boş olup olmadığını kontrol ediyor eğer boş ise false değerini dönderecektir.

```
if((user.length <= 2) || (user.length > 20)) {
```

```
document.getElementById('username').innerHTML = " Kullanıcı adı 2 ile  
20 karekter uzunluğunda olmalıdır.";
```

```
return false;
```

```
}
```

Burda ise girilen karakterlerin 2 ile 20 karakter uzunluğunda olduğunu kontrol etmektedir değilse sana uyarı verecektir.

```
if((pass.length <= 5) || (pass.length > 20)) {  
  
document.getElementById('passwords').innerHTML =" Password  
uzunluğu 5 ile 20 karakter uzunluğunda olmalıdır.";  
  
return false;  
  
}  
  
if(pass!=confirmpass){  
  
document.getElementById('confrmpass').innerHTML =" Password ile  
password doğrulama eşleşmiyor.";  
  
return false;  
  
}
```

Burda ise şifrelerin 5 ile 20 karakter arasında olduğunu kontrol ediyor ve iki şifrenin birbiriyle aynı olup olmadığını kontrol etmektedir.

Bu yapıların hepsi basit ve bir bakışta anlaşılabilir kodlar olduğu için daha fazla yazımı karmaşık hale getirmiyorum.

## Adam Asmaca Oyunu

## Yapıyı Anlatan Rapor

Aslında bu oyunu yaparken baya zahmet çektim ama tam olarak oldu mu bilmiyorum ^

İlk baş sayfa yapısını oluşturdum daha sonra script kodlarına geçtim.

İlk olarak tıklayaam butonların hepsini buton şekline oluşturup koydum.Aslında tek hazır olarak kullandığım yer burası diğerler her şeyde DOM yapısını kullanarak devam ettim.

```
var kelimeler = [  
  
"javascript",  
  
"php",
```

```
"angular",  
"asp",  
"bilgisayar",  
"elektronik",  
"yazılım",  
"yönetim bilişim"  
];
```

Burada oyunumuzda çıkacak kelimelerimizi yerleştirdik.

```
var kelime = kelimeler[Math.floor(Math.random() * kelimeler.length)];
```

Sonra bu kelimeleri random atayacak şekilde atadık.

```
for (var i = 0; i < kelime.length; i++) {  
    var harf = document.createElement("button");  
    harf.setAttribute("type","button")  
    harf.classList.add("btn2");  
    harf.style.cssText="margin-left: 2px";  
    harf.textContent="_";  
    burayaKoy.appendChild(harf);  
    answerArray[i] = "_";  
}
```

Burada oluşturduğumuz kelimeleri bir butona oluşturmamız gerekti ve her kelime için ve kelimenin uzunluğu kadar DOM yapılarını kullanarak butonlar oluşturduk ve textContent metoduyla bu butonların içini ilk önce “\_” yaptık.Aynı şekilde class ve css özelliklerini de verdik.

```
var kalan_boyut=kelime.length;
```

Kelimenin uzunluğunu değişkene atayıp daha sonra bu değer 0 olduğunda kelimelerin hepsini doğru bilmiş olacağız ve oyunu bitirmiş olacağız.

```
function f1(deger){  
    var tahmin = deger.value;  
    var harfsec= document.querySelectorAll(".btn2");  
    for (var j = 0; j < kelime.length; j++) {
```

```
if (kelime[j] === tahmin) {  
  
    harfsec[j].textContent=tahmin;  
    kalan_boyut --;  
}else {  
  
    harfler.textContent=tahmin+" ";  
}  
}  
if (kalan_boyut === 0) {  
    alert("Tebrikler Cevabı Bildiniz Tekrar Oynamak İster misin ? :)");  
}  
}
```

Asıl oyunun işleyişini veren yer burasıdır aslında. Her butona biz bir onclick metodu verdik ve bu metod sayesinde o butonun içindeki değeri alabilmiş olduk. Sonra bu değeri tahmin diye bir değişkene atadık sonra her kelimenin uzunluğu kadar o kelimeyi for ile gezdik.

Eğer dedik bu kelimenin içinde bizim butona bastığımız tahmin değeri varsa içi “\_” böyle olan butonların içini bulduğumuz tahmin değerinin text değerini içine yazacaktır.

Sonra kalan\_boyut – yaptık ki 0 olduğunda oyunun bitmesini anlayalım.

Harflerin hepsini bildiğimizde ekrana alert ile uyarı verecektir.