**JavaScript Seçiciler**

**Bu kısımda kullanılan html dökümanı --------------**

<!DOCTYPE html>

<html lang="tr">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Document</title>

        <style>

            .kirmizi{

                color: red;

            }

        </style>

    </head>

    <body>

        <h1 id="baslik">Lorem, ipsum dolor.</h1>

        <p class="paragraf">Lorem ipsum dolor sit amet.</p>

        <p class="paragraf">Nihil quidem harum ut eum non at laudantium tempora minima quam maiores!</p>

        <ul>

            <li>Lorem, ipsum dolor.</li>

            <li>Um yut eum non </li>

            <li>Minima quam maiores</li>

        </ul>

        <button type="submit">Bas</button>

        <script src="./degisken.js"></script>

    </body>

</html>

**Bu kısımda kullanılan html dökümanı --------------**

**TEKİL ELEMENT SEÇİCİ**

🡺ID Seçici

ID seçici yukarıdan aşağıya ilgili ID’ye sahip olan ilk elemanı seçer .

let baslik = document.getElementById('baslik')

console.log(baslik)//h1#baslik

baslik.style.backgroundColor='blue'

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

🡺Query Selector ile Genel Seçici

‘querySelector’ seçicisi ile element adı , class adı veya ID ile nesne seçilebilir . Eğer seçilmek istenen nesneden sayfada birden fazla varsa(aşağıdaki li gibi) yukarıdan ilkini seçer . QuerySelector ile direkt css seçicilerini seçebiliriz ; torun , çocuk ya da kardeş seçici vs.

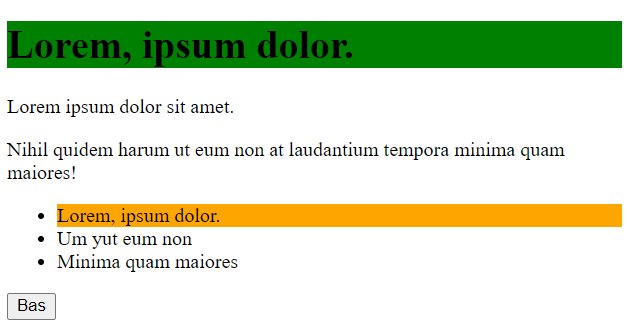
let baslik = document.querySelector('#baslik')

console.log(baslik)//h1#baslik

baslik.style.backgroundColor='green'

let listeElemani = document.querySelector('li')

listeElemani.style.backgroundColor='orange'



**ÇOĞUL ELEMENT SEÇİCİ**

🡺Class Seçici

Class Seçici ilgili class’a sahip sayfadaki tüm elementleri seçer ve geriye bir nesne döndürür . Bu nesne bir HTMLCollection’dur

let paragraflar = document.getElementsByClassName('paragraf')

console.log(paragraflar)//HTMLCollection(2) [p.paragraf, p.paragraf]

console.log(typeof paragraflar)//object

console.log(paragraflar[0])//p.paragraf

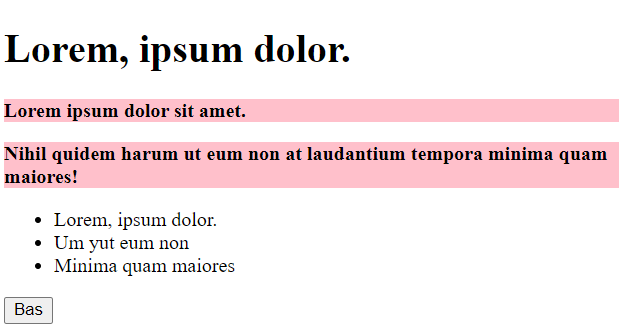
console.log(paragraflar[1])//p.paragraf

for (const iterator of paragraflar) {

    iterator.style.fontWeight = 'bold'

    iterator.style.backgroundColor= 'pink'

}



🡺QuerySelectorAll ile Çoklu Genel Seçici

Class seçici gibi bir obje döndürür . Bu nesne bir NodeList’tir .

let listeItemleri = document.querySelectorAll('li')

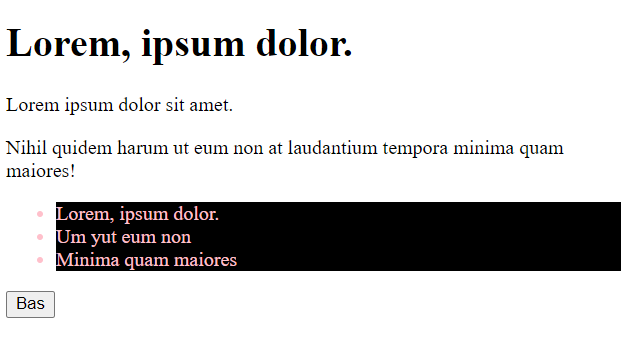
console.log(listeItemleri)//NodeList(3) [li, li, li]

for (const listeItemi of listeItemleri) {

    listeItemi.style.color='pink'

    listeItemi.style.backgroundColor='black'

}



Çoklu seçicilerde element adına göre vs. seçmede var ama onları eklemedim .

**DOM Elementleri Üzerinde Gezinmek**

**Bu kısımda kullanılan html dökümanı --------------**

    <body>

        <h1 class="header">Hello</h1>

        <ul id="liste">

            <li class="liste-item">Lorem, ipsum dolor.</li>

            <li class="liste-item">Um yut eum non </li>

            <li class="liste-item">Minima quam maiores</li>

        </ul>

        <button type="submit" id="myButton">Gönder</button>

        <script src="./degisken.js"></script>

    </body>

**Bu kısımda kullanılan html dökümanı --------------**

🡺**Child Etikete Erişmek**

Child etikete erişmek için ‘children’ , ‘firstElementChild’ ve ‘lastElementChild’ kullanılabilir .

‘children’ bir HTMLCollection dizisi döndürür .

‘firstElementChild’ ve ‘lastElementChild’ ise direkt tek bir element döndürür .

var ul = document.getElementById('liste')

console.log(ul.children)//HTMLCollection(3) [li.liste-item, li.liste-item, li.liste-item]

console.log(ul.children[0])//li.liste-item

console.log(ul.firstElementChild)//li.liste-item

console.log(ul.lastElementChild)//li.liste-item

🡺**Parent Etikete Erişmek**

Bunun için ise ‘parentElement’ kullanılabilir . Direkt Parent nesnenin kendisini döndürür .

var ul = document.getElementById('liste')

console.log(ul.parentElement)//body

🡺**Komşu Etikete Erişmek**

Komşu Etiketlere ise ‘nextElementSibling’ ve ‘previousElementSibling’ ile erişebiliriz , bunlarda direkt nesnein kendisini döndürürler .

var ul = document.getElementById('liste')

console.log(ul.nextElementSibling)//button#myButton

console.log(ul.previousElementSibling)//h1.header

**Element Oluşturma , Silme ve Güncelleme**

**Bu kısımda kullanılan html dökümanı --------------**

<body>

        <h1 class="header">Hello</h1>

        <ul id="liste">

            <li class="liste-item">1.Liste Elemanı</li>

            <li class="liste-item">2.Liste Elemanı</li>

            <li class="liste-item">3.Liste Elemanı</li>

        </ul>

        <button type="submit" id="myButton">Gönder</button>

        <script src="./degisken.js"></script>

    </body>

**Bu kısımda kullanılan html dökümanı --------------**

🡺Element Oluşturma : ‘insertAdjacentHTML’ metodu kullanarak eleman ekleyebiliriz :

var ul = document.querySelector('#liste')

console.log(ul.children.length) // 3

var eklenecekLi =

`

    <li class='liste-item'>${ul.children.length+1}.Liste Elemanı </li>

`

ul.insertAdjacentHTML('beforeend',eklenecekLi)

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Burada ‘insertAdjacentHTML’ metodunun ilk parametresi 4 farklı değer alır , biz burada eklenecek eleman sona gelsin diye ‘'beforeend'’ kullandık , bunu ezberlemek zorunda değiliz .

🡺Element ve Attribute Silme :

Element Silmek için ‘**remove**’ metodu kullanılabilir .

var ul = document.querySelector('#liste')

ul.children[0].remove()

Attr silmek içinse ‘**removeAttribute’** metodu kullanılır .

var ul = document.querySelector('#liste')

console.log(ul.children[0])//li.liste-item

ul.children[0].removeAttribute('class')

console.log(ul.children[0])//li

🡺Attr değiştirme ile Class ekleme/çıkarma

‘**setAttribute**’ ile nesnenin ilgili attribute’si değiştirilir . Aşağıdaki örnekte class bilgisi değiştirilmiş ancak dikkat edilmesi gereken nokta önceki class’ı silinmiş , yani bu güncellemeden ziyade direkt değiştirmedir .

var ul = document.querySelector('#liste')

ul.children[0].setAttribute('class','EklenenClass\_1 EklenenClass\_2')

console.log(ul.children[0])//li.EklenenClass\_1.EklenenClass\_2

Bir nesnede class işlemleri için ‘classList’ yapısı kullanılabilir .

Ekleme için ‘**.add**’

var ul = document.querySelector('#liste')

ul.children[0].classList.add('eklenen\_class')

console.log(ul.children[0])//li.liste-item.eklenen\_class

Silme için ‘**remove**’

var ul = document.querySelector('#liste')

ul.children[0].classList.remove('liste-item')

console.log(ul.children[0])//li

İlgili nesnenin bir class’a sahip olup olmadığını kontrol etmek için ‘**contains**’ kullanılabilir . Boolean değer döndürür

var ul = document.querySelector('#liste')

console.log(ul.children[0].classList.contains('liste-item'))//true

🡺**Child Etikete Erişmek**

asd