DOM(Document Object Model)

Document Object Model(Doküman Obje Modeli). Herhangi bir programlama dili veya script ile bir dökümana dinamik olarak erişmeyi ve bu dökümanın içeriğini, yapısını, stilini dinamik olarak değiştirmeyi sağlayan bir yapıdır. Bu W3C’nin DOM için yaptığı tanımdır.

* DOM bir W3C standardıdır.

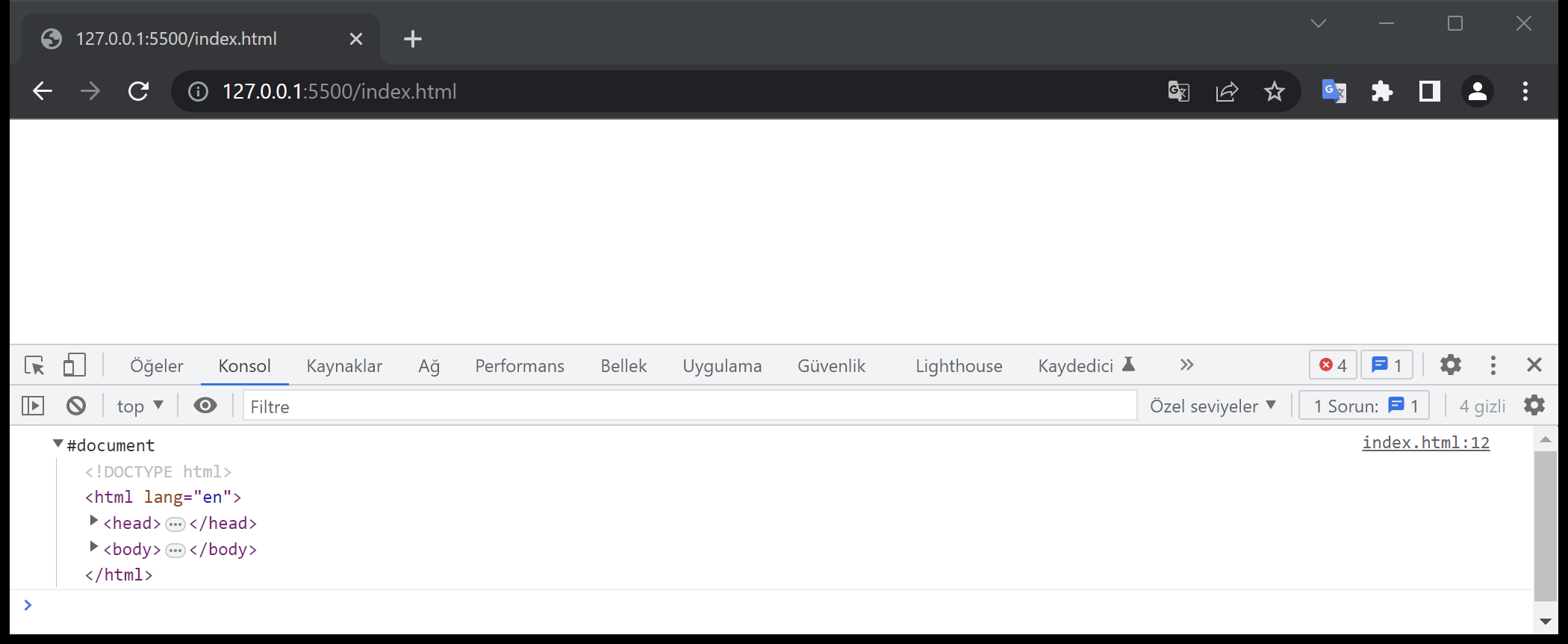
HTML DOM

İnternet tarayıcıları web sayfalarını bir belge, bu belge içerisinde bulunan tüm elemanları da birer nesne olarak kabul ederler. DOM bu nesneler üzerinde değişiklik yapma, yenilerini ekleme veya var olanları silme gibi işlemler yapmamıza olanak sağlamaktadır.

* Bir web sayfası yüklendiğinde tarayıcı tüm html öğelerini obje olarak tanımlar ve DOM’u yani sayfanın nesne modelini oluşturur.
* document objesi HTML DOM objesidir. document objesi window objesinin altında bulunmaktadır.

Example

console.log(document);



(document Object)

document.body

body elementini seçmek için kullanılmaktadır.

document.head

head elementini seçmek için kullanılmaktadır.

document.documentElement

Kök elementi seçmek için kullanılmaktadır.

document.characterSet

Sayfanın karakter kodlamasını döndürmektedir.

document.title

Sayfa başlığını almak ya da değiştirmek için kullanılmaktadır.

document.write()

Parametre olarak verilen değeri sayfa içerisine yazar. (Bu metod test amaçlı kullanılmaktadır)

document.getElementyById(id)

id değeri verilen elemanı seçer. Aynı id’ye sahip birden fazla eleman varsa sadece ilk eleman seçilir. id’si verilen eleman yoksa null değer döner.

document.querySelector(cssSelector)

Seçici ile eşleşen elemanı seçer. Seçici birden fazla elemanı hedef alıyorsa sadece ilk eleman seçilir. Seçici ile eşleşen eleman yoksa null değer döner.

document.getElementsByTagName(tagName)

Belirtilen etiket ismine sahip tüm elemanları bir obje içerisinde döndürür. Dönen obje HTMLCollection türündedir.

document.getElementsByClassName(className)

Belirtilen class isimlerine sahip tüm elemanları bir obje içerisinde döndürür. Dönen obje HTMLCollection türündedir.

* Birden fazla class adı için boşluk kullanılmaktadır. Elemanların parametre olarak verilen class isimlerinin tamamını içermesi gerekmektedir.

Example

//accordion-item class'ını uygulayan tüm elemanlar seçilir.

let elements = document.getElementsByClassName("accordion-item");

console.log(elements);

document.createElement(elementName)

Element node oluşturmak için kullanılmaktadır. Oluşturulan elementler appendChild(), insertBefore() ya da replaceChild() metodları ile sayfaya eklenebilmektedir. Elementlerin içerikleri innerHTML ve textContent özellikleri ile düzenlenebilmektedir.

document.createTextNode(text)

Text node oluşturmak için kullanılmaktadır. Oluşturulan text node’lar appendChild(), insertBefore() ya da replaceChild() metodları ile sayfaya eklenebilmektedir.

Example

let element = document.createElement("p");

let textNode = document.createTextNode("this is a text node.")

element.appendChild(textNode);

console.log(element); //<p>this is a text node.</p>

document.createAttribute(attributeName)

Attribute node oluşturmak için kullanılmaktadır. Oluşturulan attribute node’lar setAttributeNode() metodu ile elementlere eklenmektedir.

Example

let element = document.createElement("p");

let attribute = document.createAttribute("title");

attribute.nodeValue = "this is a title for p element."

element.setAttributeNode(attribute);

console.log(element); //<p title="this is a title for p element."></p>

document.addEventListener(eventName, function, true || false)

Dökümana olay tanımlamak için kullanılmaktadır. Browser olay hakkında bilgi içeren bir event objesini fonksiyona parametre olarak göndermektedir.

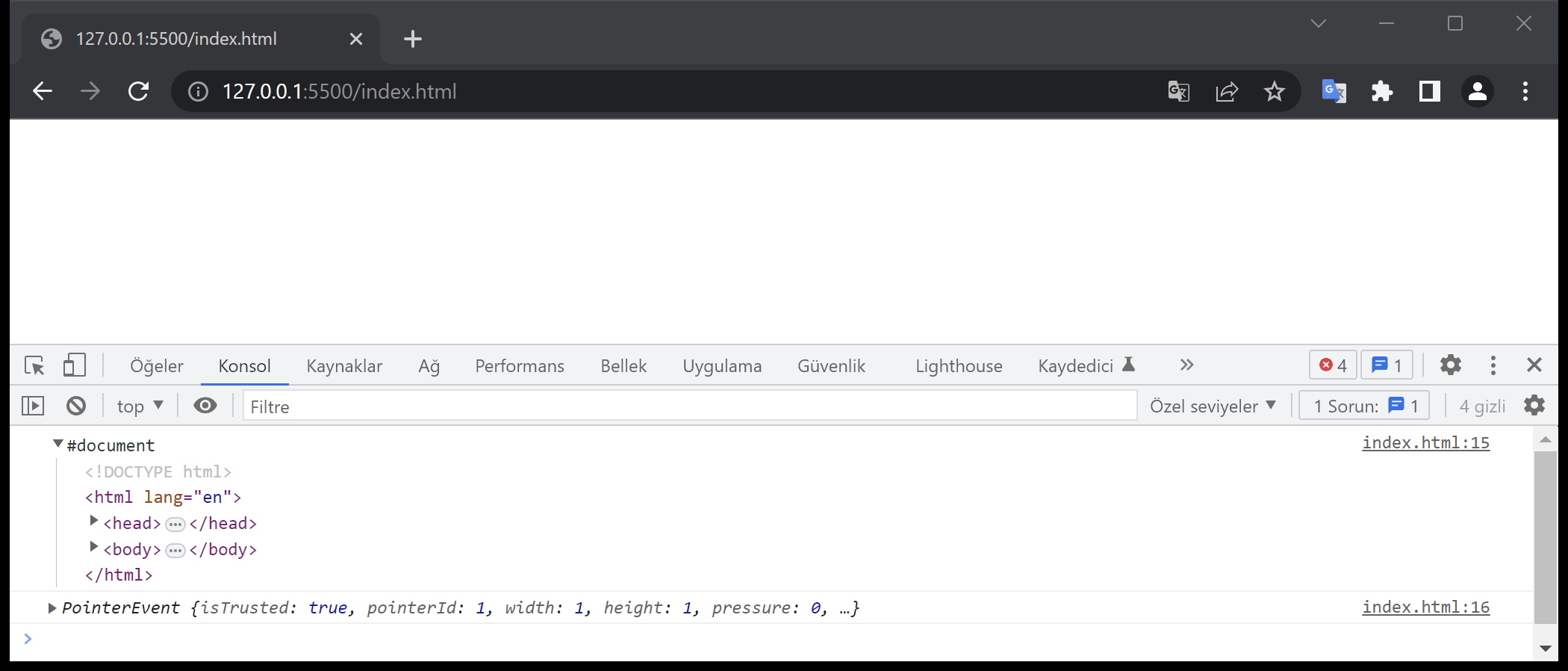
Example

document.addEventListener("click",function(e){

  console.log(this);

  console.log(e);

},false); //false : Event Bubbling



document.removeEventListener(eventName, function, true || false)

Dökümana addEventListener() fonksiyonu ile eklenmiş bir olayı kaldırmak için kullanılır. Olay hem bubbling hem de capturing olarak tanımlanmışsa her biri ayrı ayrı kaldırılmaktadır.

document.referrer

Bir bağlantıya tıklayarak sayfaya gelen ziyaretçilerin geldiği sayfanın url adresini döndürür. Bağlantıya tıklanmadan gelinmesi durumunda(Örneğin sayfa adresinin URL’ye yazılıp girilmesi durumu) boş dize döner.

* Web sayfasının ziyaretçi kaynağını öğrenmek için kullanılabilir.

document.cookie

Çerez oluşturmak ya da mevcut çerezleri okumak için kullanılmaktadır.

Example

document.cookie = "name=John;";

document.cookie = "age=30;";

console.log(document.cookie); //name=John; age=30

* Çerezler için süre belirtilmez ise tarayıcı kapatıldığında ilgili çerezler silinir.
* Çerezler için süre tanımlanırken toUTCString() metodu kullanılmaktadır.

<https://www.w3schools.com/jsref/jsref_toutcstring.asp>

Example

let date = new Date();

date.setDate(1);

date.setMonth(0);

date.setFullYear(2024);

console.log(date.toUTCString()); //Mon, 01 Jan 2024 08:54:43 GMT

document.cookie = "name=John; expires=" + date.toUTCString() + "; path/";

document.cookie = "age=30; expires=" + date.toUTCString() + "; path/";

console.log(document.cookie); //name=John; age=30

* Aynı isimde yeniden çerez tanımlarsanız değer değişikliği yapmış olursunuz.
* Bir çerezi silmek için aynı isimde yeniden çerez oluşturun ve tarih olarak geçmiş bir tarihi verin.(path parametresini unutmayın)
* Çerezler noktalı virgül gibi bazı sembolleri içeremezler. encodeURIComponent() metodu size bu konuda yardımcı olur.

document.createDocumentFragment()

Belge parçası oluşturmak için kullanılır.

* Sayfa içerisine birden fazla node eklenmesi gerekiyorsa bir belge parçası oluşturmak kullanışlıdır. Tüm node’lar fragment içerisine yerleştirilir ve tek seferde DOM render edilir.

Example

<ul class="menu"></ul>

<script>

let menu = document.querySelector(".menu");

let menuItemText = ["Home", "About Us", "Contact"];

let fragment = document.createDocumentFragment();

for(let i = 0; i < menuItemText.length; i++){

let element = document.createElement("li");

element.textContent = menuItemText[i];

fragment.appendChild(element);

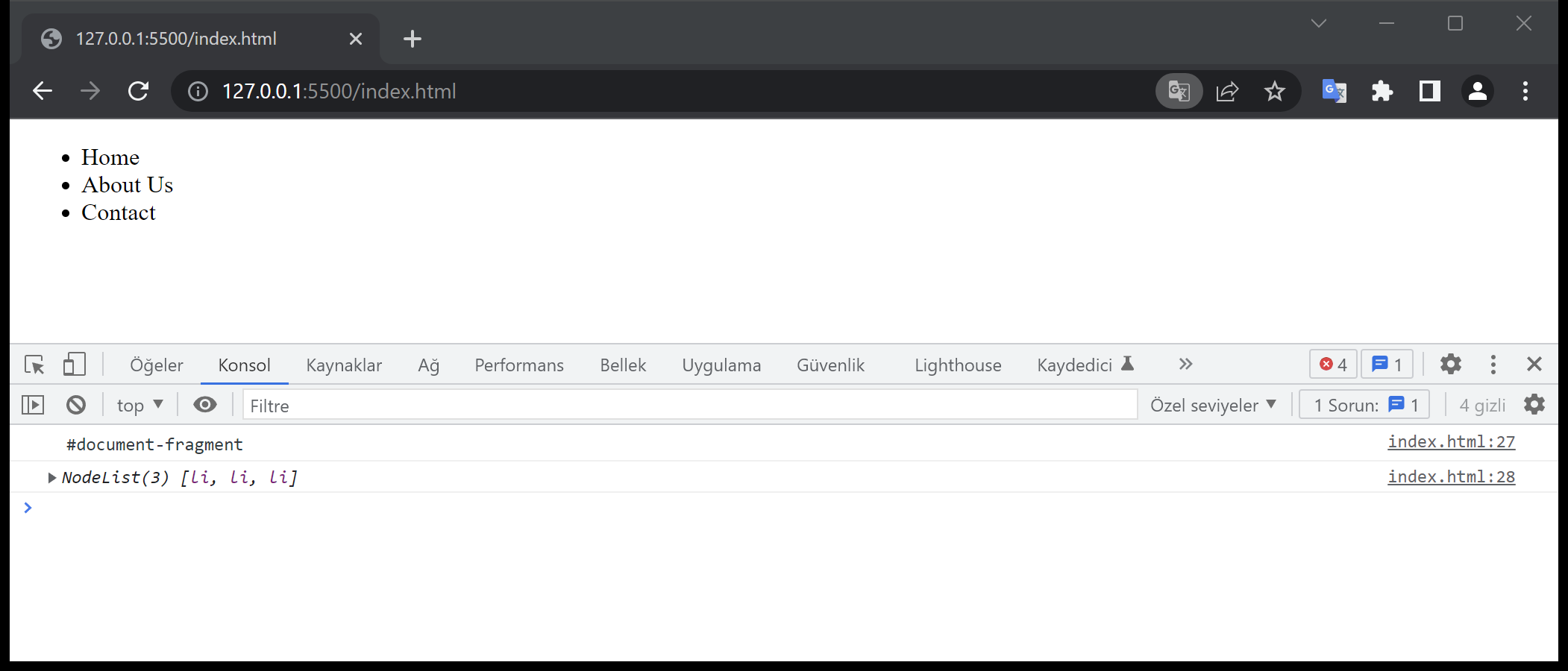
}

console.log(fragment);

console.log(fragment.childNodes);

menu.appendChild(fragment);

</script>



document.fullscreenEnabled

Belgenin tam ekran modunda görüntülenip görüntülenemeyeceğini gösteren boolean bir değer döndürür.

Example

console.log(document.fullscreenEnabled); //true

Example

<input type="button" value="Open Video in Fullscreen Mode" onclick="openFullScreen()"><br><br>

<video width="400" src="video.mp4" controls></video>

<script>

let element = document.getElementsByTagName("video")[0];

function openFullScreen(){

if(document.fullscreenEnabled){

element.requestFullscreen();

}

}

</script>

|  |
| --- |
| **NOT**  requestFullscreen() metodu bir öğeyi tam ekran modunda görüntülemek için kullanılmaktadır.  document.exitFullscreen() metodu tam ekran modunu iptal etmek için kullanılmaktadır. |

document.fullScreenElement

Bu özellik tam ekran modunda görüntülenen geçerli öğeyi, tam ekran modunda değilken null değer döndürmektedir.

(DOM Element)

tagName

Elementin adını verir.

Example

console.log(document.body.tagName); //BODY

nodeName

Düğümün adını verir.

nodeValue

Düğümün değerini almak ya da değiştirmek için kullanılır.

Example

<ul id="cities">

<li>Kocaeli</li>

<li>Ankara</li>

<li>İstanbul</li>

</ul>

<script>

let element = document.getElementById("cities");

console.log(element.attributes[0].nodeName); //id

console.log(element.attributes[0].nodeValue); //cities

</script>

Example

<div class="container"></div>

<script>

  let element = document.querySelector(".container");

  element.attributes[0].nodeValue = "container col-xs-12";

  console.log(element.attributes[0].nodeValue); //container col-xs-12

  console.log(element); //<div class="container col-xs-12"></div>

</script>

nodeType

Düğümün tipini verir.

1 : ELEMENT\_NODE

2 : ATTRIBUTE\_NODE

3 : TEXT\_NODE

4 : CDATA\_SECTION\_NODE

5 : ENTITY\_REFERANCE\_NODE

6 : ENTITY\_NODE

7 : PROCESSING\_INSTRUCTION\_NODE

8 : COMMENT\_NODE

9 : DOCUMENT\_NODE

10 : DOCUMENT\_TYPE\_NODE

11 : DOCUMENT\_FRAGMENT\_NODE

12 : NOTATION\_NODE

parentElement

Bir üst elementi seçmek için kullanılır.

childNodes

Elementin tüm çocuk düğümlerini döndürür. Dönen obje NodeList türündedir.

Example

<ul id="cities">

<li>Kocaeli</li>

<li>Ankara</li>

<li>İstanbul</li>

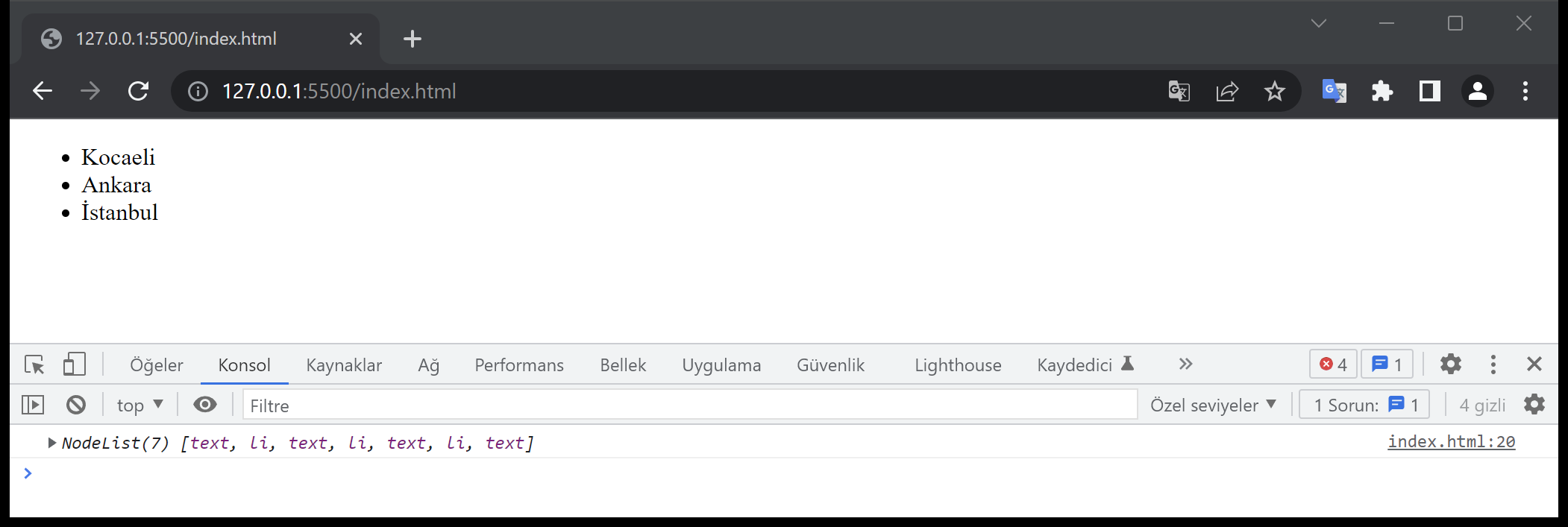
</ul>

<script>

let element = document.getElementById("cities");

console.log(element.childNodes);

</script>



children

Sadece element olan çocuk düğümleri döndürür. Dönen obje HTMLCollection türündedir.

nextSibling

Bir sonraki node’u seçmek için kullanılır.

previousSibling

Bir önceki node’u seçmek için kullanılır.

nextElementSibling

Bir sonraki element node’u seçmek için kullanılır.

previousElementSibling

Bir önceki element node’u seçmek için kullanılır.

firstChild

İlk alt node’u seçmek için kullanılır.

lastChild

Son alt node’u seçmek için kullanılır.

firstElementChild

Element olan ilk alt node’u seçmek için kullanılır.

lastElementChild

Element olan son alt node’u seçmek için kullanılır.

hasChildNodes()

Element bir alt düğüme sahipse true aksi halde false döner.

remove()

Seçilen elemanı siler. Parametresi yoktur. [Edge +]

removeChild()

Elementin alt düğümlerinden birini silmek için kullanılır.

Example

<ul id="cities">

<li>Kocaeli</li>

<li>Ankara</li>

<li>İstanbul</li>

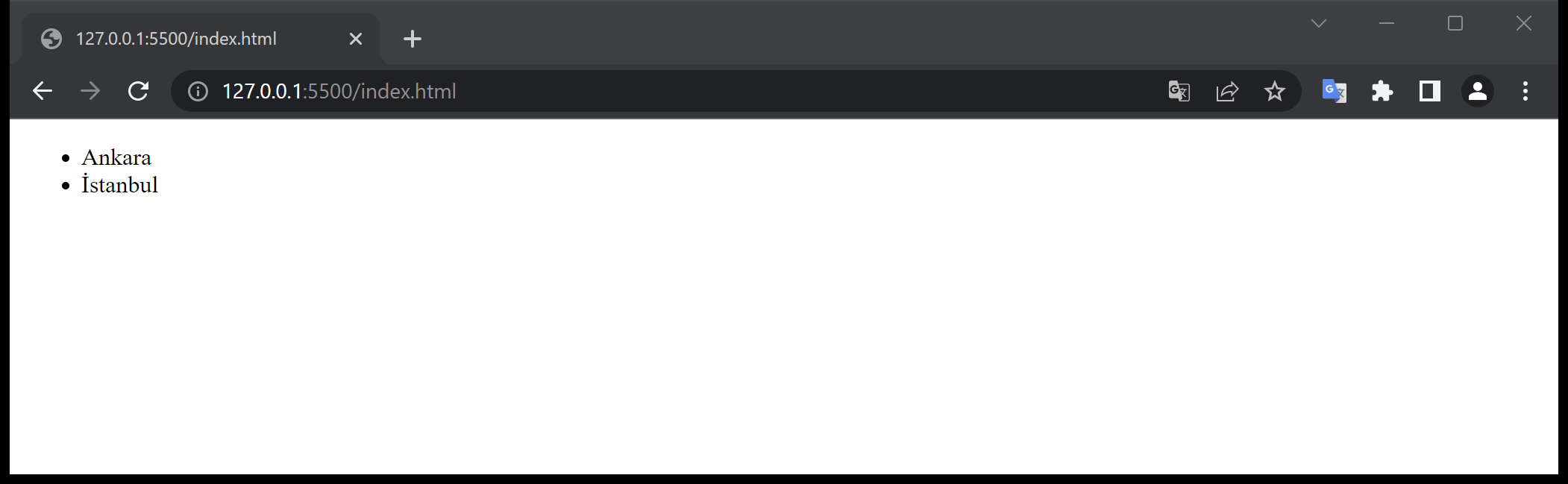
</ul>

<script>

let element = document.getElementById("cities");

element.removeChild(element.firstElementChild);

</script>



Example

<ul id="cities">

<li id="kocaeli">Kocaeli</li>

<li id="ankara">Ankara</li>

<li id="istanbul">İstanbul</li>

</ul>

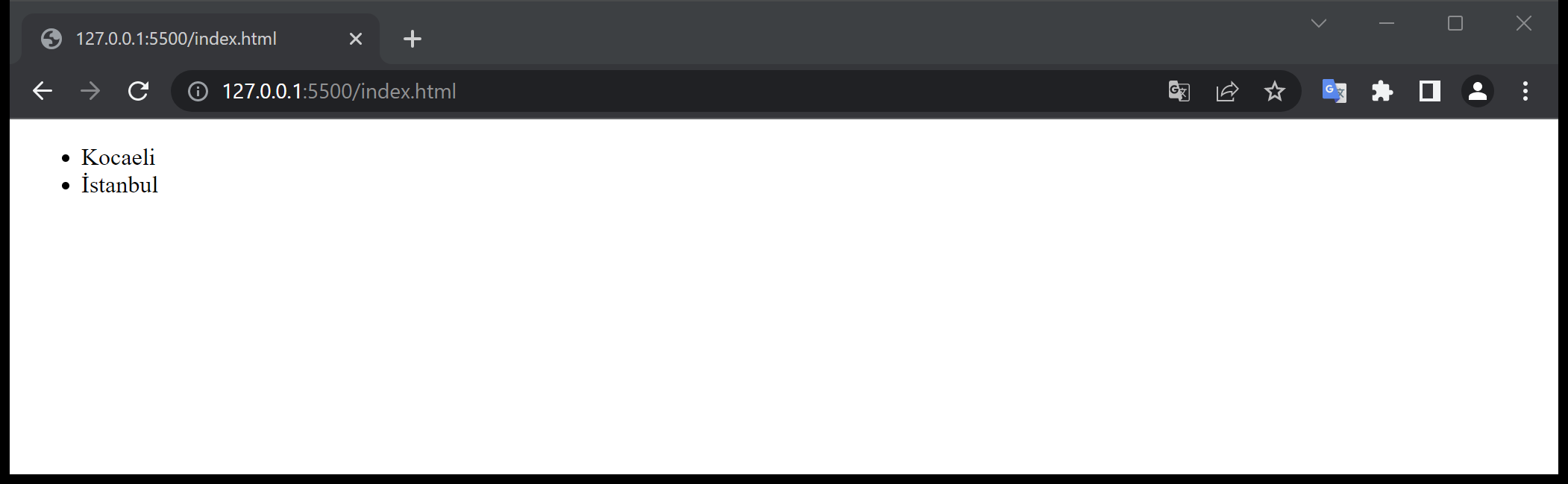
<script>

let element = document.getElementById("cities");

let childElement = document.getElementById("ankara");

element.removeChild(childElement);

</script>



replaceChild(newNode, oldNode)

Elementin bir alt düğümünü yenisi ile değiştirir.

appendChild()

Elementin alt düğümlerinin sonuna yeni bir eleman eklemek için kullanılır.

insertBefore(newElement, refElement)

Elementin bir alt düğümünün öncesine yeni bir eleman eklemek için kullanılır.

attributes

Elementin attribute’lerini bir obje içerisinde döndürür. Dönen obje NamedNodeMap türündedir.

Example

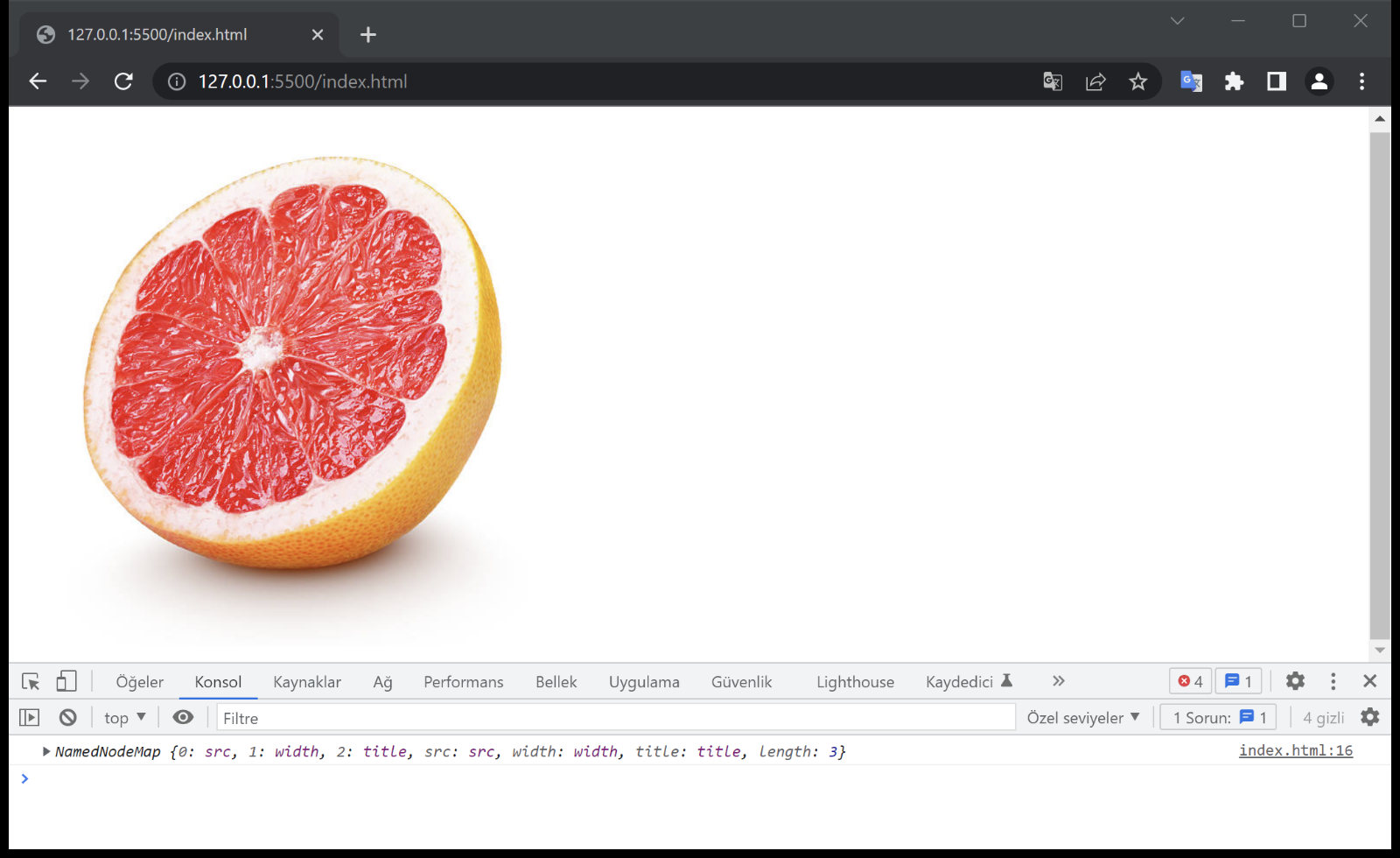
<img src="./grapefruit.png" width="400" title="grapefruit">

<script>

let element = document.getElementsByTagName("img")[0];

console.log(element.attributes);

</script>



Example

<img src="./grapefruit.png" width="400" title="grapefruit">

<script>

let element = document.getElementsByTagName("img")[0];

element.attributes[1].nodeValue = "600";

console.log(element.attributes[1].nodeValue); //600

</script>

getAttribute(attributeName)

Elementin belirtilen özelliğinin değerini verir. Özellik yoksa null değer döner.

Example

<img src="./grapefruit.png" width="400" title="grapefruit">

<script>

let element = document.getElementsByTagName("img")[0];

console.log(element.getAttribute("width")); //400

</script>

setAttribute(attributeName, attributeValue)

Elemente özellik eklemek için kullanılır. Zaten varolan bir özelliği eklerseniz değer değişikliği yapmış olursunuz.

* Bu fonksiyonla elementlere style ve event tanımlayabilirsiniz.

Example

<div class="container">div element</div>

<script>

let element = document.querySelector(".container");

element.setAttribute("style", "background-color: yellow");

</script>

removeAttribute(attributeName)

Elementin belirtilen özelliğini siler.

hasAttribute(attributename)

Element belirtilen özelliğe sahipse true aksi halde false değer döner.

hasAttributes()

Element bir özelliğe sahipse true aksi halde false değer döner.

setAttributeNode()

createAttribute() ile oluşturulan attribute node’lar bu fonksiyon ile elementlere eklenebilmektedir. Özellik zaten mevcutsa değerini değiştirir.

getAttributeNode(attributeName)

Elementin belirtilen özelliğini attribute node olarak döndürür. Belirtilen özellik yoksa null değer döner.

|  |
| --- |
| **NOT**  getAttribute() metodu sadece özelliğin değerini döndürür. getAttributeNode() ise özellik-değer ikilisini döndürmektedir. |

removeAttributeNode(attributeNode)

Attribute node silmek için kullanılır.

Example

<img src="./grapefruit.png" width="400" title="grapefruit">

<script>

let element = document.getElementsByTagName("img")[0];

element.removeAttributeNode(element.getAttributeNode("title"));

console.log(element); //<img src="./grapefruit.png" width="400">

</script>

getElementsByTagName()

getElementsByClassName()

querySelector()

addEventListener(event, function, true || false)

removeEventListener(event, function, true || false)

requestFullscreen()

focus()

Focus olayını javascript tarafında gerçekleştirir.

blur()

Blur olayını javascript tarafında gerçekleştirir.

Example

<input type="text" name="email" id="email">

<script>

let element = document.getElementById("email");

element.addEventListener("mouseover", function(){

this.focus();

});

element.addEventListener("mouseout", function(){

this.blur();

});

</script>

click()

Click olayını javascript tarafında gerçekleştirir. Bu metod ile sayfadaki herhangi bir elemana programsal olarak tıklayabilirsiniz.

id

id özelliğini almak ya da değiştirmek için kullanılır.

Example

<input type="text" name="email" id="email">

<script>

let element = document.getElementById("email");

console.log(element.id); //email

</script>

lang

lang özelliğini almak ya da değiştirmek için kullanılır.

dir

dir özelliğini almak ya da değiştirmek için kullanılır.

title

title özelliğini almak ya da değiştirmek için kullanılır.

className

Elementin class özelliğinin değerini string olarak döndürür.

Example

<div class="col-xs-12 col-sm-6">div element</div>

<script>

let element = document.getElementsByTagName("div")[0];

console.log(element.className); //col-xs-12 col-sm-6

</script>

* Bu özelliği kullanarak değer değişikliği yapabilirsiniz.

Example

<div class="col-xs-12 col-sm-6">div element</div>

<script>

let element = document.getElementsByTagName("div")[0];

element.className = "col-xs-12 col-sm-6 col-md-3";

console.log(element.className); //col-xs-12 col-sm-6 col-md-3

</script>

classList

Geriye DOMTokenList türünde bir obje döndürür. Bu objenin metodları ile elementin class değerlerine tek tek ulaşabilir, yeni değerler ekleyebilir veya var olanları silebiliriz.

length : Atanmış class sayısını döndürür.

item(index) : index numarasına göre class seçmek için kullanılır.

contains(className) : Parametre olarak verilen class DOMTokenList içerisinde mevcutsa true değilse false değer döner.

add(className) : Listeye class eklemek için kullanılır. Eklenecek eleman zaten mevcutsa ekleme yapılmaz.

remove(className) : İlgili class’ı listeden siler.

toggle(className) : İlgili class listede yoksa ekler varsa siler.

style

Geriye CSSStyleDecloration objesi döndürür. Bu ilgili elemanın CSS özelliklerini içeren bir objedir. Bu obje üzerinden sadece inline olarak tanımlanmış CSS stilleri görüntülenebilmektedir.

* Okuma yazma yapılırken **camelCase** sözdizimi kullanılmaktadır.

Example

<div style="background-color: yellow;">div element</div>

<script>

let element = document.getElementsByTagName("div")[0];

console.log(element.style.backgroundColor); //yellow

</script>

contains(element)

Bir düğümün başka bir düğümün soyundan gelip gelmediğini öğrenmek için kullanılır. (Parametre olarak verilen elemanı içeriyorsa anlamına gelmektedir)

Example

<div id="div">

div element

<p id="p">

p element

<span id="span">span element</span>

</p>

</div>

<script>

let element1 = document.getElementById("div");

let element2 = document.getElementById("span");

console.log(element1.contains(element2)); //true

</script>

offsetParent

static dışında konumlandırılmış en yakın atayı döndürür. Konumlandırılmış eleman yoksa body elementi döner.

Example

<div id="div1" style="position: relative">

div1

<div id="div2">

div2

<div id="div3">div3</div>

</div>

</div>

<script>

let element = document.getElementById("div3");

console.log(element.offsetParent); //div1 elementi döner.

</script>

cloneNode(true || false)

Seçilen elemanın bir kopyasını döndürür.

* true parametresi, elemanın tüm alt node’leriyle birlikte kopyalanmasını sağlar. Kopyalamaya elemanın ve alt node’lerinin içerikleri ve özellikleri dahildir. Parametre false yapılırsa içerik alınmaz sadece element ve özellikleri kopyalanır.

innerHTML

Elementin html içeriğini almak ya da değiştirmek için kullanılır.

Example

<ul id="cities">

<li>Kocaeli</li>

<li>Ankara</li>

<li>İstanbul</li>

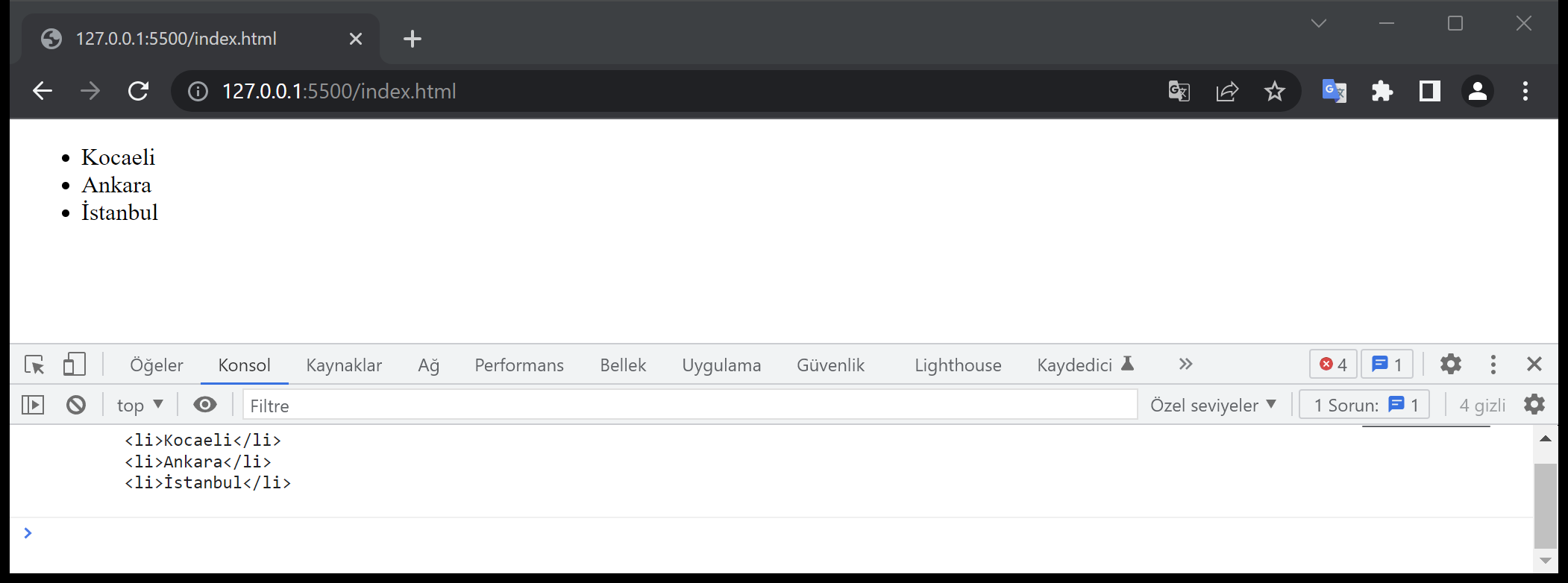
</ul>

<script>

let element = document.getElementById("cities");

console.log(element.innerHTML);

</script>



outerHTML

innerHTML’den farklı olarak hem elementin kendisini hem de içeriğini döndürmektedir. Değişiklik yapıldığında ilgili element silinip yerine yenisi gelir.

Example

<ul id="cities">

<li>Kocaeli</li>

<li>Ankara</li>

<li>İstanbul</li>

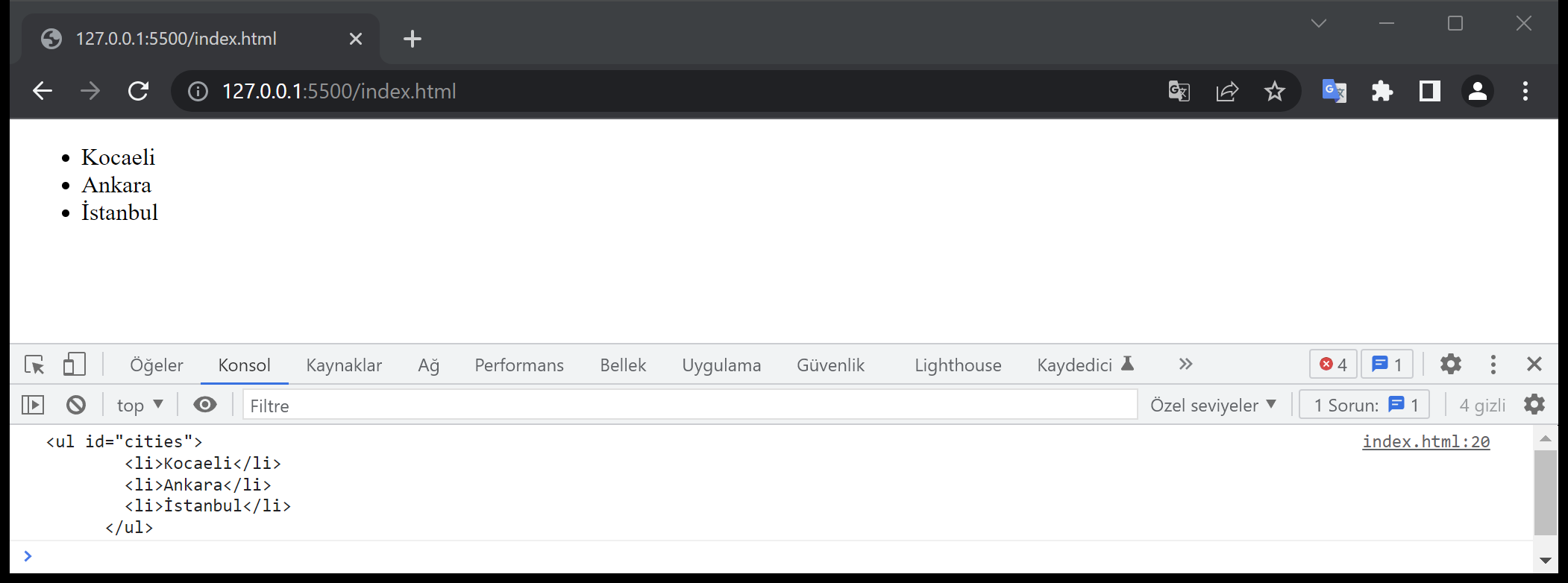
</ul>

<script>

let element = document.getElementById("cities");

console.log(element.outerHTML);

</script>



textContent

Öğenin ve tüm alt öğelerinin içeriğini ayarlar ya da döndürür. CSS ile gizlenmiş elemanların içerikleri de alınır. Değişiklik yaparken kullandığımız html kodları text olarak kabul edilmektedir.

scrollTop

Dikey kaydırma çubuğunun üstten uzaklığını almak ya da değiştirmek için kullanılır. (Elementler için)

* Element bir kaydırma çubuğuna sahip değilse 0 değeri döner ve değişiklik yapılamaz.
* Element bir kaydırma çubuğuna sahipse (scrollTop = number, scrollTop += number, scrollTop -= number) şeklinde değişiklik yapabilirsiniz.
* Negatif bir değer verirseniz scrollTop 0’a eşitlenir.
* Verdiğiniz değer maksimum kaydırma miktarından büyükse scrollTop maksimum değere eşitlenir.

scrollLeft

Yatay kaydırma çubuğunun soldan uzaklığını almak ya da değiştirmek için kullanılır. (Elementler için) (scrollTop ile benzer şekilde kullanılmaktadır)

|  |
| --- |
| **NOT**  scrollTop ve scrollLeft’de uzaklıklar piksel cinsindendir. |

(DOM Events)

onclick

Fare ile tıklandığında. (Sadece sol tuş için tetiklenmektedir)

odblclick

Fare ile çift tıklandığında. (Sadece sol tuş için tetiklenmektedir)

onmouseover

Fare ile üzerine gelindiğinde. (İçteki elemanlar için yeniden tetiklenir)

onmouseout

Fare üzerinden çekildiğinde. (İçteki elemanlar için yeniden tetiklenir)

onmouseenter

Fare ile üzerine gelindiğinde. (İçteki elemanlar için çalışmaz)

onmouseleave

Fare üzerinden çekildiğinde. (İçteki elemanlar için çalışmaz)

onmousedown

Farenin herhangi bir tuşuna basıldığında.

onmouseup

Farenin basılı tuşu bırakıldığında.

onkeydown

Klavyeden bir tuşa basıldığında.

onkeypress

onkeydown’dan sonra tetiklenmektedir.

onkeyup

Klavyenin basılı tuşu bırakıldığında.

* Tuş basılı olarak tutulursa keydown ve keypress olayları sürekli gerçekleşir bu sırada keyup çalışmaz. keyup basılı tuş bırakıldığında çalışır.

onfocus

onblur

onchange

Bu olay eleman focus özelliğini kaybettikten sonra çalışmaktadır.

onsubmit

Form gönderilmeden önce.

onreset

Form resetlenmeden önce.

DOMContentLoaded

DOM içeriği yüklendikten sonra bu olay tetiklenir. (document objesi üzerinde kullanılmaktadır)

onscroll

Kaydırma çubuğu hareket ettirildiğinde. (Bir element ya da window objesi üzerinde kullanılabilir)

onresize

Tarayıcı penceresi yeniden boyutlandırıldığında. (window objesi üzerinde kullanılmaktadır)

(DOM Event Object)

event.clientX

Farenin sayfadaki x kordinatını verir. Fare sayfanın en solundayken bu değer 0’dır.

event.clientY

Farenin sayfadaki y kordinatını verir. Fare sayfanın en üstündeyken bu değer 0’dır.

event.screenX

Farenin ekrandaki x kordinatını verir.

event.screenY

Farenin ekrandaki y kordinatını verir.

event.offsetX

Farenin gezdiği katmandaki x kordinatını verir.

event.offsetY

Farenin gezdiği katmandaki y kordinatını verir.

event.type

Olayın türünü başında **on** eki olmadan verir.

event.target

Olayı tetikleyen elemanı seçer.

event.stopPropagation()

Olayın yayılmasını önler. (event bubbling, event capturing)

event.stopImmediatePropagation()

Tanımlanmış olayın geri kalanının yürütülmesini durdurur. Ayrıca olayın DOM ağacında yayılmasını önler.

event.preventDefault()

Bu metodu bir formun gönderilmesini ya da bir bağlantının açılmasını önlemek için kullanabilirsiniz. (a elementinin **click** ya da form elementinin **submit** olayı içerisinde çağırılır)

event.key

Basılan tuşu döndürür.

event.shiftKey

SHIFT tuşuna basılmışsa true aksi halde false döndürür.

event.altKey

ALT tuşuna basılmışsa true aksi halde false döndürür.

event.ctrlKey

CTRL tuşuna basılmışsa true aksi halde false döndürür.

event.metaKey

META tuşuna basılmışsa true aksi halde false döndürür.

event.button

Hangi fare düğmesine basıldığını döndürür.

0 : Sol tuş.

1 : Tekerleğe basılmış.

2 : Sağ tuş.

* Genellikle **onmousedown** ve **onmouseup** olayları ile birlikte kullanılmaktadır.

(Event Bubbling)

Olay hedef elemandan DOM ağaç yapısı boyunca yukarıya doğru yayılır. addEventListener metodunun

üçüncü parametresine false değeri verilmelidir.

Example

<div>

<span></span>

</div>

<script>

//olay akışı aşağıdaki gibi olur.

//span'a tıklandığında : span -> div -> document

//div'e tıklandığında : div -> document

//sadece dökümana tıklanırsa : document

</script>

(Event Capturing)

Olay DOM ağaç yapısına göre kökten hedef elemana doğru ilerler. addEventListener metodunun

üçüncü parametresine true değeri verilmelidir.

Example

<div>

<span></span>

</div>

<script>

//olay akışı aşağıdaki gibi olur.

//span'a tıklandığında : document -> div -> span

//div'e tıklandığında : document -> div

//sadece dökümana tıklanırsa : document

</script>