BOM(Browser Object Model)

(location Object)

location.hash

URL içerisinden hash değerini almak ya da değişiklik yapmak için kullanılır.

Example

//http://127.0.0.1:5500/index.html#hash

console.log(window.location.hash); //hash

* Değişiklik yapılırken # sembolü kullanılmaz.

Example

window.location.hash = "hash";

location.search

URL içerisinden querystring değerlerini almak ya da değişiklik yapmak için kullanılır.

Example

//http://127.0.0.1:5500/index.html?name=John&age=30

console.log(window.location.search); //?name=John&age=30

* Değişiklik yapılırken ? sembolü kullanılmaz.

Example

window.location.search = "name=John&age=30";

location.href

URL adresinin tamamını döndürür. Farklı bir adrese yönlendirme yapmak için de kullanabilirsiniz.

Example

//URL : http://127.0.0.1:5500/index.html?name=John&age=30#hashValue

console.log(window.location.href);

//Result : http://127.0.0.1:5500/index.html?name=John&age=30#hashValue

location.hostname

Example

//URL : http://127.0.0.1:5500/index.html?name=John&age=30#hashValue

console.log(window.location.hostname); //127.0.0.1

location.host

Example

//URL : http://127.0.0.1:5500/index.html?name=John&age=30#hashValue

console.log(window.location.host); //127.0.0.1:5500

location.port

Example

//URL : http://127.0.0.1:5500/index.html?name=John&age=30#hashValue

console.log(window.location.port); //5500

location.protocol

Example

//URL : http://127.0.0.1:5500/index.html?name=John&age=30#hashValue

console.log(window.location.protocol); //http:

location.origin

URL içerisinden protokol, domain ve port numarasını birlikte almak için kullanılır.

Example

//URL : http://127.0.0.1:5500/index.html?name=John&age=30#hashValue

console.log(window.location.origin); //http://127.0.0.1:5500

location.pathname

URL içerisinden sayfa adını almak için kullanılır.

Example

//URL : http://127.0.0.1:5500/index.html?name=John&age=30#hashValue

console.log(window.location.pathname); ///index.html

(screen Object)

screen.width

Ekranın genişliğini verir.

screen.height

Ekranın yüksekliğini verir.

screen.availWidth

Ekranın kullanılabilir genişliğini verir.

screen.availHeight

Ekranın kullanılabilir yüksekliğini verir.

Example

console.log(window.screen.width); //1280

console.log(window.screen.height); //720

console.log(window.screen.availWidth); //1280

console.log(window.screen.availHeight); //672

(history Object)

history.back()

Geçmiş listesindeki bir önceki sayfayı yükler. Yalnızca önceki sayfa varsa çalışır.

history.forward()

Geçmiş listesindeki bir sonraki sayfayı yükler. Yalnızca sonraki sayfa varsa çalışır.

history.length

Geçmiş listesindeki URL sayısını döndürür. Liste geçerli sayfayı da içerdiğinden en az 1 değeri döner. Bu özellik kullanıcının oturum boyunca kaç sayfa ziyaret ettiğini öğrenmek için kullanışlıdır.

history.go(number)

Geçmiş listesinden bir sayfayı yükler. Parametre zorunludur ve pozitif-negatif değer alabilmektedir.

Example

//Bir önceki sayfaya gider. history.back() ile aynı sonucu verir.

window.history.go(-1);

history.pushState({}, “”, URL)

(window Object)

scrollTo(x, y)

Tarayıcının kaydırma çubuğunu belirtilen koordinatlara kaydırır. (Uzaklık px cinsindendir)

scrollBy(x, y)

Tarayıcının kaydırma çubuğunu belirtilen değerler kadar kaydırır. (Uzaklık px cinsindendir)

pageXOffset

Tarayıcının kaydırma çubuğunun soldan uzaklığını döndürür.

pageYOffset

Tarayıcının kaydırma çubuğunun üstten uzaklığını döndürür.

print()

Sayfanın çıktısını almak için kullanılır.

Example

window.print(); //Yazdırma penceresi açılır.

innerWidth

Sayfanın genişliğini döndürür.

innerHeight

Sayfanın yüksekliğini döndürür.

window.parent

Geçerli pencerenin bir üst penceresini döndürür (window objesi olarak). Üstte pencere yoksa geçerli pencere döner.

window.top

Geçerli pencerenin en üstteki tarayıcı penceresini döndürür (window objesi olarak). Üstte pencere yoksa geçerli pencere döner.

window.self

Geçerli pencereyi döndürür (window objesi olarak).

* self özelliği genellikle aşağıdaki gibi karşılaştırma yapılırken kullanılmaktadır.

Example

<!-- index.html içeriği -->

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

<iframe src="./frame.html" style="width:400px; height:400px"></iframe>

<!-- frame.html içeriği -->

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

<script>

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

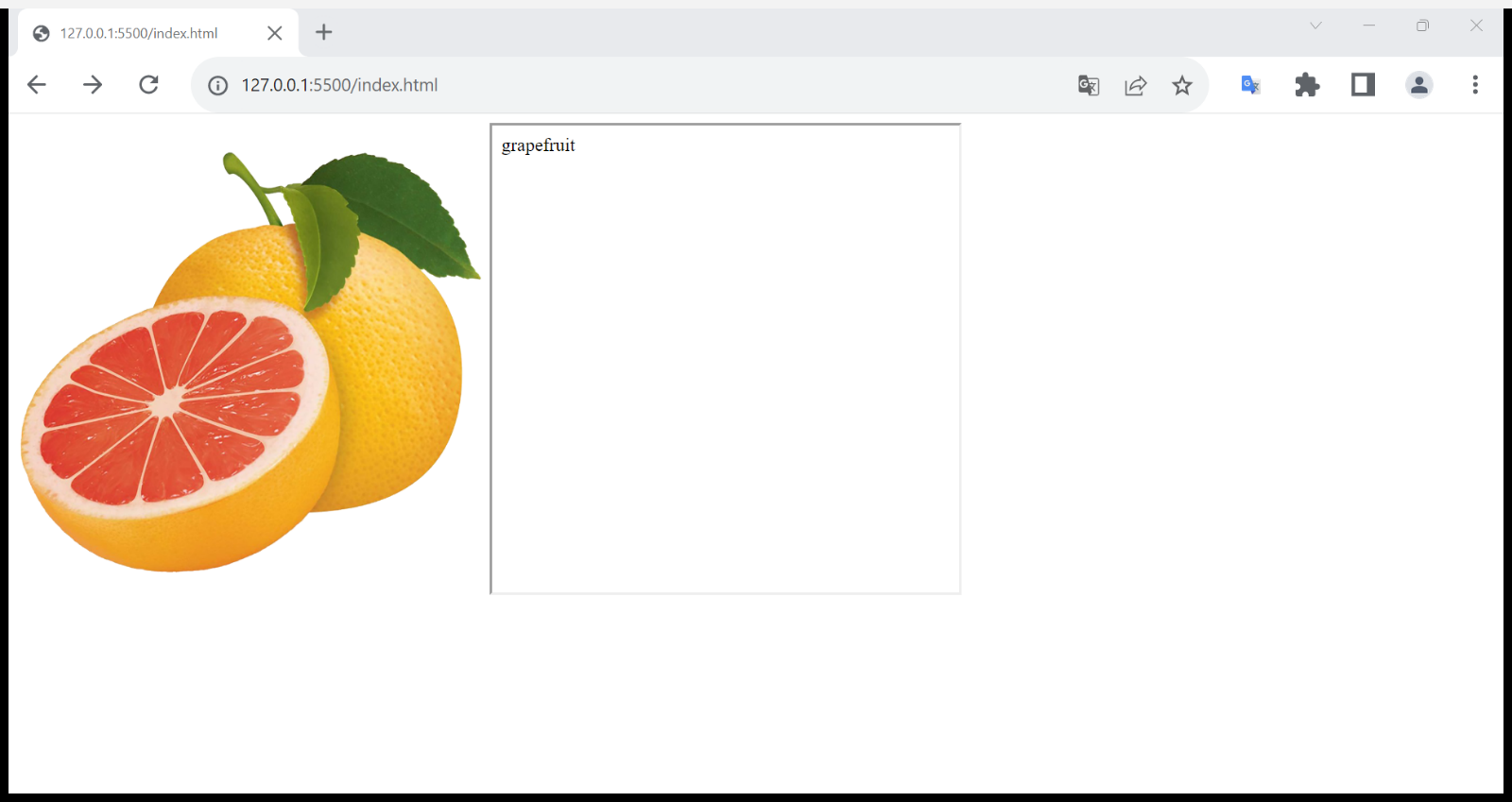
if(window.top != window.self){ //bir iframe içinde olduğumuzu anlıyoruz.

//en üst penceredeki img'nin title özelliğini elde ediyoruz :)

element.innerHTML = window.top.document.getElementsByTagName("img")[0].title;

}

</script>



getComputedStyle(element)

Parametre olarak verdiğimiz elemanın tüm CSS özelliklerini içeren bir CSSStyleDecloration objesi döndürür. Bu obje devralınan, dış stil sayfaları tarafından uygulanan ve tarayıcı tarafından uygulanan varsayılan stilleri de içerir.

Example

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

let style = window.getComputedStyle(element);

console.log(style); //CSSStyleDecloration objesi konsola yazılır.

Example

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

let style = window.getComputedStyle(element);

//getPropertyValue(string) : İlgili özelliğin değerini döndürür.

console.log(style.getPropertyValue("width")); //1264px

//item(index) : İlgili index'deki özelliğin adını string olarak döndürür.

console.log(style.item(0)); //accent-color

* item() metodu ile obje içerisindeki tüm değerleri okuyabilirsiniz.

Example

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

<script>

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

let style = window.getComputedStyle(element);

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

let property = style.item(i);

console.log(property + " : " + style.getPropertyValue(property));

}

</script>