

HIZLI AÇIKLAMA KLAVUZU

abs() : Kendisine parametre olarak verilen herhangi bir sayı değerinin mutlak değerini bularak sonucu geriye döndürür. *(Matematiksel işlem)*

accessKey : Belge içerisindeki herhangi bir HTML elemanına atanmış olan accessKey değerini elde etmek veya ayarlamak için kullanılır. *(Document Object Model)*

accesskey : Nesneye klavyeden bir kısayol tuşu tanımlar. *(HTML Global Attributes)*

acos() : Kendisine parametre olarak verilen -1 ile 1 arasındaki sayı değerinin ark kosinüsünü radyan cinsinden bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

acosh() : Kendisine parametre olarak verilen herhangi bir sayı değerinin hiperbolik ark kosinüsünü bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

activeElement : Belge içerisindeki seçili veya odaklanılmış elemanın referansını elde etmek için kullanılır. *(Document Object Model)*

addEventListener() : Belgeye event işaretleyicisi / dinleyicisi eklemek için kullanılır. İşaretlenen / dinlenen event belgede gerçekleştirilecek olur ise belirtilecek olan fonksiyon veya fonksiyonlar çalışır. *(Document Object Model)*

adoptNode() : Belge içerisine başla bir belgeden node eklemek için kullanılır. Eklenecek olan node'un varsa tüm alt node'larında işleme dahil edilerek ilgili tüm node'lar diğer belgeden kaldırılır. *(Document Object Model)*

alert() : Sayfa içerisinde uyarı penceresi açar. Belirtilen uyarıyı ekranda gösterir. *(Uyarı penceresi)*

altKey : Belge içerisindeki herhangi bir event tetiklenirken alt tuşuna basılıp basılmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Document Object Model)*

anchor() : Değişkene çapa tanımlaması yapar. Aynı zamanda çapaya name değeri atar. *(Değişkenler)*

anchors : Belge içerisindeki çapa tag'ına sahip tüm elemanların referansını elde etmek için kullanılır. Elemanların referansları birden fazla ise her bir eleman döngü ile elde edilebilir. *(Document Object Model)*

appName : Browser'ın kod adı referansını elde etmek için kullanılır. *(Browser Object Model)*

appendChild() : Bir node'u belge içerisinde belirtilecek olan herhangi bir elemanın içerisine en sonda olmak üzere eklemek için kullanılır. *(Document Object Model)*

appName : Browser adının referansını elde etmek için kullanılır. *(Browser Object Model)*

appVersion : Browser'ın sürüm bilgisi referansını elde etmek için kullanılır. *(Browser Object Model)*

arguments : Fonksiyona herhangi bir parametre tanımlanmadan bile gelebilecek tüm parametre değerlerinin alınmasını sağlar. *(Fonksiyonlar)*

asin() : Kendisine parametre olarak verilen -1 ile 1 arasındaki sayı değerinin ark sinüsünü radyan cinsinden bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

asinh() : Kendisine parametre olarak verilen herhangi bir sayının değerinin hiperbolik ark sinüsünü bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

assign() : Browser penceresine yeni bir sayfa yüklemek için kullanılır. *(Browser Object Model)*

atan() : Kendisine parametre olarak belirtilen herhangi bir sayı değerinin ark tanjantını radyan cinsinden bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

atan2() : Kendisine parametre olarak verilen sayı değerlerinin ark tanjantını radyan cinsinden bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

atanh() : Kendisine parametre olarak verilen herhangi bir sayı değerinin -1 ile 1 arasındaki sayı değerinin hiperbolik ark tanjantını bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

atob() : Pencere içerisindeki herhangi bir metnin base64 kodlamasına dönüştürülmüş karakterlerini çözerek değeri geriye döndürür. *(Browser Object Model)*

availHeight : Kullanıcının windows görev çubuğu hariç ekranının yükseklik değerini elde etmek için kullanılır. *(Browser Object Model)*

availWidth : Kullanıcının windows görev çubuğu hariç ekranının genişlik değerini elde etmek için kullanılır. *(Browser Object Model)*

avarage() : Number veri türü içerikli dizinin her elemanı üzerinde tek tek olmak üzere çalışır ve tüm elemanların ortalamasını bularak, bulduğu değeri geriye döndürür. *(Diziler)*

back() : Pencerenin sayfa geçmişinde 1 geri gider. *(Browser Object Model)*

baseURI : Belgenin mutlak URL adresi referansını elde etmek için kullanılır. *(Document Object Model)*

big() : Değişken içerisindeki metni büyük fontlu metin yaparak değerini geriye döndürür. *(Değişkenler)*

blur() : Belge içerisinde odaklanılmış herhangi bir HTML elemanının odağından çıkmak için kullanılır. *(Document Object Model)*

body : Belge içerisindeki body tag'ının referansını elde etmek için kullanılır. *(Document Object Model)*

bold() : Değişken içerisindeki metni kalın fontlu metin yaparak değerini geriye döndürür. *(Değişkenler)*

break : Döngü işleminden çıkmak için kullanılır. *(Döngü)*

break : Koşul kontrollerini sonlandırmak için kullanılır. *(Koşul & Kontrol)*

btoa() : Pencere içerisindeki bir metnin tüm karakterlerini base64 kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür. *(Browser Object Model)*

button : Belge içerisinde mouse ile tıklama event'ı gerçekleştiğinde mouse'un hangi tuşuna basıldığı değerini geriye döndürür. *(Document Object Model)*

buttons : Belge içerisinde mouse ile tıklama event'ı gerçekleştiğinde mouse'un hangi tuşuna veya tuşlarına basıldığı değerini geriye döndürür. *(Document Object Model)*

caller : Fonksiyonu çağıran diğer bir fonksiyon referans olarak alınarak tüm bilgilerinin elde edilmesini sağlar. *(Fonksiyonlar)*

cancelable : Belge içerisinde tanımlı event'ın iptal edilebilir olup olmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Document Object Model)*

case : Ek koşul kontrolleri için kullanılır. *(Koşul & Kontrol)*

catch : Hata oluşursa çalıştırılacak kodlar için kullanılır. *(Koşul & Kontrol)*

cbqrt() : Kendisine parametre olarak verilen herhangi bir sayının küp kökünü bularak, bulduğu değeri geriye döndürür. *(Matematiksel işlem)*

ceil() : Kendisine parametre olarak verilen değeri yukarı yuvarlayarak değeri geriye döndürür. *(Matematiksel işlem)*

characterSet : Belgenin karakter kodlamasının referansını elde etmek için kullanılır. *(Document Object Model)*

charAt() : Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğindeki karşılığı olan karakteri bularak geriye döndürür. *(Değişkenler)*

charCode : Belge içerisindeki onkeypress event'ı gerçekleştiğinde klavyeden basılan tuşun unicode kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür. *(Document Object Model)*

charCodeAt() : Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğindeki karşılığı olan karakteri bularak, karakterin Unicode değerini geriye döndürür. *(Değişkenler)*

childElementCount : Belge içerisindeki herhangi bir HTML elemanının içerdiği alt elemanlarının sayısını elde etmek için kullanılır. *(Document Object Model)*

childNodes : Belge içerisindeki herhangi bir HTML elemanının içerdiği alt node'ları elde etmek için kullanılır. *(Document Object Model)*

children : Belge içerisindeki herhangi bir HTML elemanının içerdiği alt elemanları elde etmek için kullanılır. *(Document Object Model)*

class : Nesneye CSS ile oluşturulmuş bir veya daha fazla sınıf ataması tanımlar. *(HTML Global Attributes)*

classList : Belge içerisindeki herhangi bir HTML elemanına atanmış olan class listesi isimlerini elde etmek veya ayarlamak için kullanılır. *(Document Object Model)*

className : Belge içerisindeki herhangi bir HTML elemanının class adını elde etmek veya ayarlamak için kullanılır. *(Document Object Model)*

clearInterval() : setInterval() metodu kullanılarak oluşturulmuş olan zamanlayıcıyı kaldırmak / iptal etmek / geçersiz kılmak için kullanılır. *(Zamanlayıcı)*

clearTimeout() : setTimeout() metodu kullanılarak oluşturulmuş olan zamanlayıcıyı kaldırmak veya iptal etmek veya geçersiz kılmak için kullanılır. *(Zamanlayıcı)*

clientHeight : Belge içerisindeki herhangi bir HTML elemanının içerik yüksekliği değerini elde etmek için kullanılır. *(Document Object Model)*

clientLeft : Belge içerisindeki herhangi bir HTML elemanının sol kenarlık genişliği değerini elde etmek için kullanılır. *(Document Object Model)*

clientTop : Belge içerisindeki herhangi bir HTML elemanının üst kenarlık yüksekliği değerini elde etmek için kullanılır. *(Document Object Model)*

clientWidth : Belge içerisindeki herhangi bir HTML elemanının içerik genişliği değerini elde etmek için kullanılır. *(Document Object Model)*

clientX : Belge içerisindeki herhangi bir mouse event'ı gerçekleştiğinde browser penceresine göre mouse'un yatay konum koordinatının değerini geriye döndürür. *(Document Object Model)*

clientY : Belge içerisindeki herhangi bir mouse event'ı gerçekleştiğinde browser penceresine göre mouse'un dikey konum koordinatının değerini geriye döndürür. *(Document Object Model)*

cloneNode() : Belge içerisindeki herhangi bir HTML elemanının içerdiği node'u kopyalamak için kullanılır. *(Document Object Model)*

close() : Geçerli pencereyi kapatmak için kullanılır. *(Browser Object Model)*

close() : open() metodu ile açılan HTML çıktı akışını kapatmak / sonlandırmak için kullanılır. *(Document Object Model)*

closed : Bir pencerenin kapatılıp kapatılmadığını kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Browser Object Model)*

colorDepth : Kullanıcının ekranı için tanımlanmış olan renk derinliği / renk kalitesi değerini elde etmek için kullanılır. *(Browser Object Model)*

compareDocumentPosition() : Belge içerisinde bulunan node'ların birbirlerine göre konum değerlerini elde etmek için kullanılır. *(Document Object Model)*

compatMode : Belgenin browser tarafından hangi modda çalıştırıldığının referansını elde etmek için kullanılır. *(Document Object Model)*

compile() : Önceden oluşturulmuş düzenli ifade içerisindeki desenin değiştirilmesini sağlar. *(Düzenli ifadeler)*

concat() : Üzerinde çalışılan değişkene parametrik olarak verilen değişken veya değişkenleri dahil ederek yeni bir içerik oluşturur. Oluşturduğu değeri geriye döndürür. *(Değişkenler)*

concat() : Üzerinde çalışılan diziye parametrik olarak verilen eleman veya elemanları yada dizi veya dizileri dahil ederek yeni bir dizi oluşturur ve oluşturmuş olduğu dizi değerini geriye döndürür. *(Diziler)*

confirm() : Kullanıcı işlemlerinde onay penceresi oluşturmak için kullanılır. *(Browser Object Model)*

console.log() : Tarayıcı konsoluna değer ataması yapar veya kayıt atar. *(Konsol çıktısı)*

const : Sabit değişken tanımlamakta kullanılır. *(Değişkenler)*

constructor : Nesnenin constructor'ını elde etmek için kullanılır ve constructor'ın değerini geriye döndürür. Ayrıca aynı zamanda kontrol işlemleri içinde kullanılabilir. *(Nesneler)*

constructor : Object'in oluşturduğu yapıcı metoda erişmek için kullanılan özelliktir. *(Object Oriented Programming)*

contains() : Belge içerisindeki herhangi bir node'un, başka bir node'un alt node'u düğümü olup olmadığını kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Document Object Model)*

contentEditable : Belge içerisindeki herhangi bir HTML elemanının contentEditable değerini elde etmek veya ayarlamak için kullanılır. *(Document Object Model)*

contentEditable : Nesnenin düzenlenip düzenlenemeyeceğini tanımlar. *(HTML Global Attributes)*

contextMenu : Nesneyi bir menü ile ilişkilendirir. Nesnenin bağlam menüsü de denilebilir. Nesneye sağ tıklamalarda açılacak olan menü ilişkilendirilmesidir. *(HTML Global Attributes)*

continue : Döngünün o anki işlemini sonlandırarak döngünün başına döner ve işleme kaldığı yerden devam eder. *(Döngü)*

cookie : Çerez işlemleri için kullanılır. *(Document Object Model)*

cookieEnabled : Browser'ın cookie kullanımını destekleyip desteklemediğini kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Browser Object Model)*

copyWithin() : Kendisine verilecek olan parametreler ile tanımlanmış dizi aralığında bulunan değerleri kopyalayarak, parametreler doğrultusunda dizinin elemanlarını değiştirir ve dizinin değerini geriye döndürür. *(Diziler)*

cos() : Kendisine parametre olarak verilen herhangi bir sayı değerinin kosinüsünü bularak, bulduğu değeri geriye döndürür. *(Matematiksel işlem)*

cosh() : Kendisine parametre olarak verilen herhangi bir sayının hiperbolik kosinüsünü bularak, bulduğu değeri geriye döndürür. *(Matematiksel işlem)*

createAttribute() : Belge içerisindeki istenilen bir elemana yeni bir özellik / nitelik node'u oluşturur ve oluşturduğu bu node'u bir node object olarak geriye döndürür. *(Document Object Model)*

createComment() : Belge içerisine yeni bir açıklama node'u oluşturur ve oluşturduğu bu node'u bir node object olarak geriye döndürür. *(Document Object Model)*

createElement() : Belge içerisinde yeni bir HTML eleman node'u oluşturur ve oluşturduğu bu node'u bir node object olarak geriye döndürür. *(Document Object Model)*

createTextNode() : createElement() metodu ile oluşturulan HTML elemanına metin node'u oluşturur ve oluşturduğu bu node'u bir node object olarak geriye döndürür. *(Document Object Model)*

ctrlKey : Belge içerisindeki herhangi bir event tetiklenirken ctrl tuşuna basılıp basılmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Document Object Model)*

currentTarget : Belge içerisinde tanımlı event'ı tetikleyen herhangi bir HTML elemanının event işaretleyicileri / dinleyicileri tarafından tag adı referansını elde etmek için kullanılır. *(Document Object Model)*

data-* : Sayfa veya uygulamaya özel verileri saklamak, tutmak veya yakalamak için kullanılır. *(HTML Global Attributes)*

Date() : Kullanıldığı değişken içerisinde tarih ve saat bilgilerini depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

debugger : Script'leri bölüm bölüm durdurarak ve çalıştırarak kontrol işlemleri için kullanılır. *(Hata ayıklama & Kontrol)*

decodeURI() : Değişken içerisindeki metnin özel karakterlerinin UTF-8 kodlamasına dönüştürülmüş karakterlerini çözerek değeri geriye döndürür. *(Değişkenler)*

decodeURIComponent() : Değişken içerisindeki metnin özel karakterlerinin UTF-8 kodlamasına dönüştürülmüş karakterlerini çözerek değeri geriye döndürür. *(Değişkenler)*

default : Koşul olumsuzluklarının sonucu için kullanılır. *(Koşul & Kontrol)*

defaultPrevented : Belge içerisinde tanımlı event için preventDefault() metodunun çağrılıp çağrılmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Document Object Model)*

defaultView : Belgenin window referansını elde etmek için kullanılır. *(Document Object Model)*

delete : Dizi içerisindeki belirtilen indise / indexe ait elemanın silinmesini sağlar. İndis / index silinmez, sadece değeri boşaltılır yani tanımsız olur. *(Diziler)*

delete : Object'i veya object'in bir özelliğini silmek için kullanılır. *(Nesneler)*

delete : Herhangi bir object'i veya object'in bir özelliğini silmek için kullanılır. *(Özel operatör)*

designMode : Belgenin çalışma anında düzenlenebilir olup olmadığı referansını elde etmek için kullanılır. *(Document Object Model)*

detail : Belge içerisindeki mouse ile tıklama event'ı gerçekleştirildiğinde mouse'a kaç defa tıklandığı sayısı değerini geriye döndürür. *(Document Object Model)*

dir : Belge içerisindeki metinlerin yazılış yönü referansını elde etmek için kullanılır. Ayrıca belge içerisindeki metinlerin yazılış yönünün atamasını ayarlamak içinde kullanılır. *(Document Object Model)*

dir : Nesnedeki içeriğin yazılış yönünü tanımlar. *(HTML Global Attributes)*

do while : Döngü oluşturmak için kullanılır. *(Döngü)*

doctype : Belge doküman türü referansını elde etmek için kullanılır. *(Document Object Model)*

documentElement : HTML tag'ının referansını element nesnesi halinde elde etmek için kullanılır. *(Document Object Model)*

documentMode : Belgeyi oluşturmak için browser tarafından kullanılan modun referansını elde etmek için kullanılır. *(Document Object Model)*

documentURI : Belge sayfasının URI adresi referansını elde etmek için kullanılır. *(Document Object Model)*

domain : Belgenin domain referansını elde etmek için kullanılır. *(Document Object Model)*

draggable : Nesnenin sürüklenebilir olup olmadığını tanımlar. *(HTML Global Attributes)*

dropzone : Nesne sürüklenip bırak işlemi yapıldıktan sonra tarayıcının ne yapacağını tanımlar. Sürüklenen verilerin kopyalandığını da belirtir. *(HTML Global Attributes)*

E : Euler sayısı değerini geriye döndürür. *(Matematiksel işlem)*

else : Koşul olumsuzluklarının sonucu için kullanılır. *(Koşul & Kontrol)*

else if : Ek koşul kontrolleri için kullanılır. *(Koşul & Kontrol)*

embeds : Belge içerisindeki embed veya object tag'ına sahip tüm elemanların referansını elde etmek için kullanılır. Eleman referansları birden fazla ise her bir eleman döngü ile elde edilebilir. *(Document Object Model)*

encodeURIComponent() : Değişken içerisindeki metnin özel karakterlerini UTF-8 kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür. *(Değişkenler)*

encodeURIComponent() : Değişken içerisindeki metnin özel karakterlerini UTF-8 kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür. *(Değişkenler)*

endsWith() : Kendisine verilen parametre doğrultusunda değişken içeriğinin bitiş değerini kontrol ederek Boolean veri türünde sonucu geriye döndürür. *(Değişkenler)*

escape() : Değişken içerisindeki metnin özel karakterlerini Unicode kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür. *(Değişkenler)*

eval() : Kendisine parametre olarak verilen değerleri Javascript kodlamasına çevirerek komut gibi çalıştırır ve oluşan değeri geriye döndürür. *(Değişkenler)*

every() : Kendisine parametre olarak verilen fonksiyon false değeri döndürene kadar dizi elemanları üzerinde gezinilmesini sağlar. *(Diziler)*

exec() : Kendisine parametre olarak verilen değişken içeriğinde düzenli ifade ile aranan değer eşleşmesi durumunda eşleşmeyi sağlayan değişken içeriği değerinden yeni bir dizi oluşturularak, oluşturduğu dizi değerini geriye döndürür. *(Düzenli ifadeler)*

exp() : Kendisine parametre olarak verilen herhangi bir sayı değerinin euler sayısı üssünü bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

expm1() : Kendisine parametre olarak verilen herhangi bir sayının Euler sayısı üssünü bularak ve bir eksilterek, sonucu geriye döndürür. *(Matematiksel işlem)*

fill() : Kendisine verilecek olan parametreler ile tanımlanmış dizi aralığında bulunan değeri, parametre ile belirtilmiş olan değer doğrultusunda dizinin elemanlarını değiştirir ve dizinin değerini geriye döndürür. *(Diziler)*

filter() : Kendisine parametre olarak verilen fonksiyonu dizideki her eleman üzerine çalıştırır. Fonksiyonun geriye true değerini döndüren tüm elemanlarını kopyalarak yeni bir dizi oluşturur. Oluşturduğu dizinin değerini geriye döndürür. *(Diziler)*

finally : Hata oluşsada oluşmasada çalıştırılacak kodlar için kullanılır. *(Koşul & Kontrol)*

find() : Kendisine parametre olarak verilen fonksiyonu dizideki her eleman üzerine tek tek olmak üzere çalıştırır ve dizi içerisinde parametrik olarak verilen değeri arar. İlk eşleşmeyi sağlayan dizi elemanının değerini geriye döndürür. *(Diziler)*

findIndex() : Kendisine parametre olarak verilen fonksiyonu dizideki her eleman üzerine tek tek olmak üzere çalıştırır ve dizi içerisinde parametrik olarak verilen değeri arar. İlk eşleşmeyi sağlayan dizi elemanının indis / index değerini geriye döndürür. *(Diziler)*

firstChild : Belge içerisindeki herhangi bir HTML elemanının içerdiği ilk alt node'u elde etmek için kullanılır. *(Document Object Model)*

firstElementChild : Belge içerisindeki herhangi bir HTML elemanının içerdiği ilk elemanı elde etmek için kullanılır. *(Document Object Model)*

fixed() : Değişken içerisindeki metni teletype fontlu metin yaparak değeri geriye döndürür. *(Değişkenler)*

floor() : Kendisine parametre olarak verilen ondalıklı sayı değerini aşağı yuvarlayarak sonucu geriye döndürür. *(Matematiksel işlem)*

focus() : Belge içerisindeki herhangi bir HTML elemanına odaklanmak için kullanılır. *(Document Object Model)*

fontcolor() : Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğini renklendirerek değeri geriye döndürür. *(Değişkenler)*

fontsize() : Kendisine parametre olarak belirtilen değer doğrultusunda değişken içerisindeki metnin font boyutunu değiştirerek değeri geriye döndürür. *(Değişkenler)*

for : Döngü oluşturmak için kullanılır. *(Döngü)*

for in : Döngü oluşturmak için kullanılır. *(Döngü)*

forEach() : Kendisine parametre olarak verilen fonksiyonu dizideki her eleman üzerine tek tek çalıştırır ve geriye herhangi bir değer döndürmez. Fakat istenirse ilgili değerlere erişilebilir. *(Diziler)*

forms : Belge içerisindeki form tag'ına sahip elemanların referansını elde etmek için kullanılır. Eleman referansları birden fazla ise her bir eleman döngü ile elde edilebilir. *(Document Object Model)*

forward() : Pencerenin sayfa geçmişinde 1 ileri gider. *(Browser Object Model)*

frames : Pencere içerisindeki tüm çerçevelerin nesne referansını elde etmek veya ayarlamak için kullanılır. *(Browser Object Model)*

fromCharCode() : Kendisine parametre olarak verilen Unicode değerini karakter olarak geriye döndürür. Çoklu değer alabilir. *(Değişkenler)*

function : Fonksiyon oluşturmak için kullanılır. *(Fonksiyonlar)*

g : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriğinin tamamının kontrol edilmesini sağlar. *(Düzenli ifadeler)*

get : İstek verileri belirli bir kaynaktan gönderilir. Bir diğer deyiş ile form verilerini veya manuel olarak belirleyeceğimiz verileri adres çubuğundan veri gönderimi anlamına gelmektedir. *(HTML)*

getAttribute() : Belge içerisindeki herhangi bir HTML elemanının belirtilen özelliğinin değerini elde etmek için kullanılır. *(Document Object Model)*

getAttributeNode() : Belge içerisindeki herhangi bir HTML elemanının belirtilen özellik node'unun değerini elde etmek veya ayarlamak için kullanılır. *(Document Object Model)*

getDate() : Kullanıldığı değişken içeriğine ayın gün bilgisini 1 ila 31 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getDay() : Kullanıldığı değişken içeriğine haftanın gün bilgisini 0 ila 6 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getElementById() : Kendisine parametre olarak verilen değer doğrultusunda, belge içerisindeki belirtilen id özelliğine / niteliğine sahip elemanın referansını elde etmek için kullanılır. *(Document Object Model)*

getElementsByClassName() : Kendisine parametre olarak verilen değer doğrultusunda, belge içerisindeki belirtilen class adına sahip tüm elemanların referansını elde etmek için kullanılır. Eleman referansları birden fazla ise her bir eleman döngü ile elde edilebilir. *(Document Object Model)*

getElementsByName() : Kendisine parametre olarak verilen değer doğrultusunda, belge içerisinde belirtilen name özelliğine / niteliğine sahip tüm elemanların referansını elde etmek için kullanılır. Eleman referansları birden fazla ise her bir eleman döngü ile elde edilebilir. *(Document Object Model)*

getElementsByTagName() : Kendisine parametre olarak verilen değer doğrultusunda, belge içerisindeki belirtilen tag adına sahip tüm elemanların referansını elde etmek için kullanılır. Elemanların referansı birden fazla ise her bir eleman döngü ile elde edilebilir. *(Document Object Model)*

getFullYear() : Kullanıldığı değişken içeriğine yıl bilgisini 1000 ila 9999 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getHours() : Kullanıldığı değişken içeriğine günün saat bilgisini 0 ila 23 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getMilliseconds() : Kullanıldığı değişken içeriğine saniyenin milisaniye bilgisini 0 ila 999 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getMinutes() : Kullanıldığı değişken içeriğine saatin dakika bilgisini 0 ila 59 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getMonth() : Kullanıldığı değişken içeriğine yılın ay bilgisini 0 ila 11 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getNamedItem() : Belge içerisindeki HTML elemanına eklenmiş olan özellik node'u referansını elde etmek için kullanılır. *(Document Object Model)*

getSeconds() : Kullanıldığı değişken içeriğine dakikanın saniyesini 0 ila 59 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getTime() : Kullanıldığı değişken içeriğine tarihin zaman damgası bilgisini depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getTimezoneOffset() : Kullanıldığı değişken içeriğine yerel zaman ile 0 zaman dilimi arasındaki zaman farkı bilgisini dakika olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getUTCDate() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait ayın gün bilgisini 1 ila 31 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getUTCDay() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait haftanın gün bilgisini 0 ila 6 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getUTCFullYear() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait yıl bilgisini 1000 ila 9999 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getUTCHours() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait günün saat bilgisini 0 ila 23 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getUTCMilliseconds() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait saniyenin milisaniye bilgisini 0 ila 999 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getUTCMinutes() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait saatin dakika bilgisini 0 ila 59 arasında olmak üzere rakamsal olmak üzere depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getUTCMonth() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait yılın ay bilgisini 0 ila 11 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

getUTCSeconds() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait dakikanın saniye bilgisini 0 ila 59 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

global : Düzenli ifade içerisinde g ayarının kullanılıp kullanılmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Düzenli ifadeler)*

go() : Pencerenin sayfa geçmişinde belirtilen değer kadar ileri veya geri gider *(Browser Object Model)*

hasAttribute() : Belge içerisindeki herhangi bir HTML elemanının belirtilen özelliğe sahip olup olmadığını kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Document Object Model)*

hasAttributes() : Belge içerisindeki herhangi bir HTML elemanının herhangi bir özelliğe sahip olup olmadığını kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Document Object Model)*

hasChildNodes() : Belge içerisindeki herhangi bir HTML elemanının alt node'larının olup olmadığını kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Document Object Model)*

hasFocus() : Belge odağının olup olmadığı referansını boolean veri türünde elde etmek için kullanılır. *(Document Object Model)*

hasOwnProperty : Bir nesnenin parametrik olarak girilen özelliğinin kullanılıp kullanılmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Nesneler)*

head : Belge içerisindeki head tag'ının referansını elde etmek için kullanılır. *(Document Object Model)*

height : Kullanıcının toplam ekran yüksekliği değerini elde etmek için kullanılır. *(Browser Object Model)*

hidden : Nesnenin artık veya uygun olmadığını belirtir. Gizlemelerde veri tutma işlemlerinde kullanılır. *(HTML Global Attributes)*

history : Pencerenin geçmiş nesnesine erişim için kullanılır. Özellikleri sayesinde geçmiş işlemleri yapılabilir. *(Browser Object Model)*

i : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriğinin büyük harf / küçük harf ayrımı olmaksızın kontrol edilmesini sağlar. *(Düzenli ifadeler)*

id : Belge içerisindeki herhangi bir HTML elemanının id değerini elde etmek veya ayarlamak için kullanılır. *(Document Object Model)*

id : Nesneye benzersiz bir kimlik tanımlar. *(HTML Global Attributes)*

if : Koşul başlatmak için kullanılır. *(Koşul & Kontrol)*

ignoreCase : Düzenli ifade içerisinde i ayarının kullanılıp kullanılmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Düzenli ifadeler)*

images : Belge içerisindeki img tag'ına sahip tüm elemanların referansını elde etmek için kullanılır. Eleman referansları birden fazla ise her bir eleman döngü ile elde edilebilir. *(Document Object Model)*

implementation : Belge ile ilgili DOMImplementation arayüzünden türetilmiş bir DOMImplementation nesnesi referansını boolean veri türünde elde etmek için kullanılır. *(Document Object Model)*

importNode() : Belge içerisine başka bir belgeden node eklemek için kullanılır. Eklenecek olan node'un varsa tüm alt node'ları da işleme dahil edilir. *(Document Object Model)*

in : Herhangi bir object'in bir özelliğe sahip olup olmadığını sınamak için kullanılır. *(Özel operatör)*

includes() : Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğini kontrol ederek Boolean veri türünde sonucu geriye döndürür. *(Değişkenler)*

indexOf() : Değişken içerisinde parametrik olarak verilen değeri arar ve ilk eşleşmeyi sağlayan değişken içeriğinin indis / index değerini geriye döndürür. *(Değişkenler)*

indexOf() : Dizi içerisinde parametrik olarak verilen değeri arar ve ilk eşleşmeyi sağlayan elemanın indis / index değerini geriye döndürür. *(Diziler)*

innerHeight : Pencere içerik alanının yükseklik değerini elde etmek için kullanılır. *(Browser Object Model)*

innerHTML : HTML Dokümanı içerisinde bulunan HTML elemanına / elementine etki eder. Veri gönderir / veri değişimi yapar. *(Doküman Çıktısı)*

innerWidth : Pencere içerik alanının genişlik değerini elde etmek için kullanılır. *(Browser Object Model)*

input : Değişken içeriğinde düzenli ifade ile aranan değer eşleşmesi durumunda eşleşmeyi sağlayan değişken içeriğinin tüm değeri geriye döndürür. *(Düzenli ifadeler)*

inputEncoding : Belgenin karakter kodlamasının referansını elde etmek için kullanılır. *(Document Object Model)*

insertBefore() : Bir node'u belge içerisindeki belirtilecek olan herhangi bir node'un önüne eklemek için kullanılır. *(Document Object Model)*

instanceof : Bir nesnenin constructor'ının doğruluğunu kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Nesneler)*

instanceof : Herhangi bir object'in özel bir class örneği olup olmadığını sınamak için kullanılır. *(Özel operatör)*

isArray() : Bir değişkenin içeriğinin dizi olup olup olmadığını kontrol eder. Boolean türünde sonuç döndürür. *(Diziler)*

isContentEditable : Belge içerisindeki herhangi bir HTML elemanının düzenlenebilir olup olmadığını kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Document Object Model)*

isFinite() : Kendisine parametrik olarak verilen değişken içeriğinin en büyük pozitif sayı ile en küçük negatif sayı aralığında olup olmadığını kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Matematiksel işlem)*

isInteger() : Kendisine parametrik olarak verilen değişken içeriğinin tam sayı olup olmadığını kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Matematiksel işlem)*

isNaN() : Kendisine parametrik olarak verilen değişken içeriğinin NaN değerinde olup olmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Matematiksel işlem)*

isPrototypeOf : Parametrik olarak verilen object'in prototype zincirinde, bir constructor'ın bulunup bulunmadığını test etmek için kullanılır. *(Object Oriented Programming)*

isPrototypeOf() : Parametrik olarak girilen nesne prototype zincirinde, bir constructor bulunup bulunmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Nesneler)*

isSafeInteger() : Kendisine parametrik olarak verilen değişken içeriğinin güvenli bir tam sayı olup olmadığını kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Matematiksel işlem)*

italics() : Değişken içerisindeki metni italik (sağa yatık) fontlu metin yaparak değerini geriye döndürür. *(Değişkenler)*

item() : Belge içerisindeki HTML elemanına eklenmiş olan özellik node'u referansını elde etmek veya ayarlamak için kullanılır. HTML elemanı içerisindeki özellik sayısı birden fazla ise her bir özellik node' referansı döngü ile elde edilebilir. *(Document Object Model)*

javaEnabled() : Browser'da Java kodlamalarının etkinlik durumunu kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Browser Object Model)*

join() : Dizideki elemanların arasına kendisine parametre olarak verilen değeri ayraç olarak kullanarak string veri türünde düz bir metin oluşturur ve bu oluşturduğu değeri geriye döndürür. *(Diziler)*

key : Belge içerisindeki onkeypress, onkeydown veya onkeyup event'ı gerçekleştiğinde klavyeden basılan tuşun değerini geriye döndürür. *(Document Object Model)*

keyCode : Belge içerisindeki onkeypress, onkeydown veya onkeyup event'ı gerçekleştiğinde klavyeden basılan tuşun unicode kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür. *(Document Object Model)*

lang : Belge içerisindeki herhangi bir HTML elemanının lang değerini elde etmek veya ayarlamak için kullanılır. *(Document Object Model)*

lang : Nesne içeriğinin dil tanımlamasını yapar. *(HTML Global Attributes)*

language : Browser'ın dil bilgisi referansını elde etmek için kullanılır. *(Browser Object Model)*

lastChild : Belge içerisindeki herhangi bir HTML elemanının içerisindeki son alt node'u elde etmek için kullanılır. *(Document Object Model)*

lastElementChild : Belge içerisindeki herhangi bir HTML elemanının içerdiği son elemanı elde etmek için kullanılır. *(Document Object Model)*

lastIndex : Değişken içeriğinde düzenli ifade ile aranan değer eşleşmesi durumunda ilk eşleşmeyi sağlayan değişken içeriğinin bir sonraki indis / index değerini geriye döndürür. *(Düzenli ifadeler)*

lastIndexOf() : Değişken içerisinde parametrik olarak verilen değeri arar ve son eşleşmeyi sağlayan değişken içeriğinin indis / index değerini geriye döndürür. *(Değişkenler)*

lastIndexOf() : Dizi içerisinde parametrik olarak verilen değeri arar ve son eşleşmeyi sağlayan dizi elemanın indis / index değerini geriye döndürür. *(Diziler)*

lastMatch : Değişken içeriğinde düzenli ifade ile aranan değer eşleşmesi durumunda son eşleşmeyi sağlayan değişken içeriğinin değerini geriye döndürür. *(Düzenli ifadeler)*

lastModified : Belgenin en son güncelleme tarihinin ve saatinin referansını elde etmek için kullanılır. *(Document Object Model)*

leftContext : Değişken içeriğinde düzenli ifade ile aranılan değer eşleşmesi durumunda eşleşmeyi sağlayan değişken içeriği değerinin solunda kalan tüm değerleri geriye döndürür. *(Düzenli ifadeler)*

length : Pencere içerisindeki tüm çerçeveleri sayısını elde etmek için kullanılır. *(Browser Object Model)*

length : Pencerenin sayfa geçmişinde sayısını bulur. *(Browser Object Model)*

length : Değişken içeriğinin karakter sayısını geriye döndürür. *(Değişkenler)*

length : Dizi içeriğinin eleman sayısını veya dizi elemanının karakter sayısını geriye döndürür. *(Diziler)*

length : Belge içerisindeki HTML elemanına eklenmiş olan özelliklerini sayısını elde etmek için kullanılır. *(Document Object Model)*

length : Fonksiyona tanımlanan parametre sayısı değerini geriye döndürür. *(Fonksiyonlar)*

let : Blok kapsamlı yerel değişken tanımlamakta kullanılır. *(Değişkenler)*

link() : Değişkene çapa tanımlaması yapar. Aynı zamanda çapaya köprü değeri atar. *(Değişkenler)*

links : Belge içerisindeki tüm linklerin referansını elde etmek için kullanılır. Referanslar birden fazla ise her bir link referansı döngü ile elde edilebilir. *(Document Object Model)*

LN10 : 10 sayısının doğal logaritması değerini geriye döndürür. *(Matematiksel işlem)*

LN2 : 2 sayısının doğal logaritması değerini geriye döndürür. *(Matematiksel işlem)*

localeCompare() : Üzerinde çalışılan değişkene parametrik olarak verilen diğer bir değişken içeriğini tarayıcı diline göre karşılaştırarak sonucu geriye döndürür. *(Değişkenler)*

location : Belge URL adresi referansını elde etmek ve ayarlamak için kullanılır. *(Document Object Model)*

location : Belge içerisindeki onkeydown veya onkeyup event'ı gerçekleştiğinde ilgili tuşun klavyedeki lokasyon / konum bilgisi değerini geriye döndürür. *(Document Object Model)*

log() : Kendisine parametre olarak verilen herhangi bir sayının doğal logaritmasını bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

LOG10E : Euler sayısının 10 tabanına göre logaritmik değerini geriye döndürür. *(Matematiksel işlem)*

LOG2E : Euler sayısının 2 tabanına göre logaritmik değerini geriye döndürür. *(Matematiksel işlem)*

m : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriğinin birden fazla satırı varsa tüm satırların kontrol edilmesini sağlar. *(Düzenli ifadeler)*

map() : Kendisine parametre olarak verilen fonksiyonu dizideki her eleman üzerinde çalıştırır. Fonksiyonun her eleman için geriye döndürdüğü değerlerden yeni bir dizi oluşur ve yeni dizi değerini geriye döndürür. *(Diziler)*

match() : Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğini kontrol eder ve eşleşen değer varsa içeriğin eşleşen kısmından yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür. *(Değişkenler)*

match() : Kendisine parametre olarak verilen düzenli ifadeli değer doğrultusunda değişken içeriğini kontrol eder ve aranan değerlerin eşleşmesi durumunda eşleşmeyi sağlayan değişken içeriği değerlerinden yeni bir dizi oluşturarak, oluşturduğu dizi değerini geriye döndürür. *(Düzenli ifadeler)*

max() : Kendisine parametre olarak verilen sayı değerlerinin en büyüğünü bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

MAX_VALUE : Atanabilecek en büyük pozitif sayı değerini geriye döndür. *(Matematiksel işlem)*

metaKey : Belge içerisindeki herhangi bir event tetiklenirken pencere tuşuna veya emir tuşuna basılıp basılmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Document Object Model)*

MIN_VALUE : Atanabilecek en küçük sayı değerini geriye döndürür. *(Matematiksel işlem)*

min() : Kendisine parametre olarak verilen sayı değerlerinin en küçüğünü bularak, bulunduğu değeri geriye döndür. *(Matematiksel işlem)*

moveBy() : Pencerenin sol üst köşe koordinatı baz alınarak bilgisayar ekranına göre belirtilen konuma taşımak için kullanılır. *(Browser Object Model)*

moveTo() : Pencerenin sol üst köşe koordinatı baz alınarak bilgisayar ekranına göre belirtilen konuma taşımak için kullanılır. *(Browser Object Model)*

multiline : Düzenli ifade içerisinde m ayarının kullanılıp kullanılmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Düzenli ifadeler)*

name : Pencere adının referansını elde etmek veya ayarlamak için kullanılır. *(Browser Object Model)*

name : Belge içerisindeki HTML elemanına eklenmiş olan özellik adı referansını elde etmek için kullanılır. HTML elemanının içerisindeki özellik sayısı birden fazla ise her bir özellik adı referansı döngü ile elde edilebilir. *(Document Object Model)*

NaN : Değerin sayı olmaması durumunda geriye döner. *(Matematiksel işlem)*

NEGATIVE_INFINITY : Atanabilecek en küçük negatif sayı değerinden küçük sayılarda negatif sonsuzluk değerini geriye döndürür. *(Matematiksel işlem)*

new : Yeni bir object oluşturmak için kullanılır. *(Özel operatör)*

new Array() : Dizi tanımlamak için kullanılır. *(Diziler)*

new Object() : Nesne tanımlamak için kullanılır. *(Nesneler)*

newURL : Sayfa URL'inin bağlantı bölümünde değişiklik olduğundaki yeni URL adresini geriye döndürür. *(Document Object Model)*

nextElementSibling : Belge içerisindeki herhangi bir HTML elemanının içerisindeki aynı seviyede yani kardeş olan elemanlardan, belirtilecek olan elemandan sonra gelen ilk elemanı elde etmek için kullanılır. *(Document Object Model)*

nextSibling : Belge içerisindeki herhangi bir HTML elemanının içerisindeki aynı seviyede yani kardeş olan node'lardan, belirtilecek olan node'dan sonra gelen ilk node'u elde etmek için kullanılır. *(Document Object Model)*

nodeName : Belge içerisindeki herhangi bir node'un adını elde etmek için kullanılır. *(Document Object Model)*

nodeType : Belge içerisindeki herhangi bir node'un türünü elde etmek için kullanılır. *(Document Object Model)*

nodeValue : Belge içerisindeki herhangi bir node'un değerini elde etmek için kullanılır. *(Document Object Model)*

normalize() : Belge içerisindeki boş text node'larını kaldırarak bitişik node'a katılmasını sağlar. *(Document Object Model)*

noscript : Tarayıcının script kodlarını çalıştırmadığı zamanlarda bilgi vermek için kullanılır. *(HTML)*

now() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait tarihin zaman damgası bilgisini depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

Number() : Değişken içeriğini mutlak olarak Number veri türüne dönüştürerek oluşturduğu değeri geriye döndürür. *(Değişkenler)*

Number() : Değişken içeriğini mutlak olarak Number veri türüne dönüştürerek oluşturduğu içeriği geriye döndürür. *(Matematiksel işlem)*

offsetHeight : Belge içerisindeki herhangi bir HTML elemanının yükseklik değerini elde etmek için kullanılır. *(Document Object Model)*

offsetLeft : Belge içerisindeki herhangi bir HTML elemanının sol üst köşesi ile browser penceresinin sol üst köşesi arasındaki yatay uzaklık farkı değerini elde etmek için kullanılır. *(Document Object Model)*

offsetParent : Belge içerisindeki herhangi bir HTML elemanının içerisinde mutlak olarak konumlandırıldığı ilk elemanın referans nesnesini elde etmek için kullanılır. *(Document Object Model)*

offsetTop : Belge içerisindeki herhangi bir HTML elemanının sol üst köşesi ile browser penceresinin sol üst köşesi arasındaki dikey uzaklık farkı değerini elde etmek için kullanılır. *(Document Object Model)*

offsetWidth : Belge içerisindeki herhangi bir HTML elemanının genişlik değerini elde etmek için kullanılır. *(Document Object Model)*

oldURL : Sayfa URL'inin bağlantı bölümünde değişiklik olduğundaki eski URL adresini geriye döndürür. *(Document Object Model)*

onabort : Sayfanın, HTML elemanının veya medya elemanı verisinin yüklenmesi durdurulduğunda veya iptal edildiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onafterprint : Belge içeriği kullanıcı tarafından yazdırılmak istendiğinde yazdırma işlemi başladıktan sonra belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onbeforeprint : Belge içeriği kullanıcı tarafından yazdırılmak istendiğinde yazdırma işlemi başlamadan belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onbeforeunload : Sayfanın kullanıcı tarafından yenileme ve kapatma isteklerinde işlem gerçekleştirilmeden belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onblur : Form içerisindeki bir elemanın odağından çıkıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

oncanplay : Medya elemanı yürütülemeye hazır olduğunda belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

oncanplaythrough : Medya elemanı verisi tamamıyla arabelleğe yüklendiğinde ve yürütülmeye hazır olduğunda belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onchange : Form içerisindeki bir elemanın değeri veya seçimi değiştirildiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onclick : Belge içerisindeki herhangi bir elemana mouse ile sol tıklandığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

oncontextmenu : Belge içerisindeki herhangi bir elemana mouse ile sağ tımlandığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

oncopy : Form veya belge içerisindeki bir elemanın metin içeriği kullanıcı tarafından kopyalandığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

oncut : Form veya belge içerisindeki bir elemanın metin içeriği kullanıcı tarafından kesildiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ondblclick : Belge içerisindeki herhangi bir elemana mouse ile çift sol tıklandığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ondrag : Belge içerisindeki herhangi bir eleman sürüklendiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ondragend : Belge içerisindeki herhangi bir elemanın sürüklenmesi bittiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ondragenter : Belge içerisindeki herhangi bir elemanın hedef kapsama alanı üzerine giriş yaptığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ondragleave : Belge içerisindeki herhangi bir elemanın hedef kapsama alanından çıkış yaptığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ondragover : Belge içerisindeki herhangi bir elemanın hedef kapsama alanı üzerindeyken belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ondragstart : Belge içerisindeki herhangi bir eleman sürüklenmesi başladığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ondrop : Belge içerisindeki herhangi bir elemanın hedef kapsama alanı üzerine bırakıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ondurationchange : Medya elemanının nesnesinin süresi değiştirildiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onemptied : Medya elemanı kullanılamaz duruma geldiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onended : Medya elemanının oynatma süresi bittiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onerror : Sayfanın, HTML elemanının veya medya elemanı verisinin yüklenmesi browser tarafından durdurulduğunda veya hata oluştuğunda belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onfocus : Form içerisindeki bir elemana odaklanıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onfocusin : Form içerisindeki bir elemana odaklanılmak üzereyken belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onfocusout : Form içerisindeki bir elemanın odağından çıkılmak üzereyken belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onhashchange : Sayfa URL'inin bağlantı bölümünde değişiklik olur ise belirtilecek fonksiyon çalışır. *(Document Object Model)*

oninput : Form içerisindeki bir elemanın değeri veya seçimi değiştirilirken belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

oninvalid : Form içerisindeki bir elemanın değeri geçersiz ise belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onkeydown : Belge içerisindeki herhangi bir eleman üzerinde klavyeden herhangi bir tuşa basıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onkeypress : Belge içerisindeki herhangi bir eleman üzerinde klavyeden herhangi bir tuşa basıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onkeyup : Belge içerisindeki herhangi bir eleman üzerinde klavyeden basılı duran bir tuş bırakıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onLine : Browser'ın çevrimiçi modda çalıştırılıp çalıştırılmadığını kontrol ederek, boolean veri türünde sonucu geriye döndürür. *(Browser Object Model)*

onload : Sayfanın veya herhangi bir HTML elemanının yüklenmesi tamamlandığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onloadeddata : Medya elemanı verisi yüklendiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onloadedmetadata : Medya elemanı meta bilgileri verisi tamamen yüklendiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onloadstart : Medya elemanı verisi browser tarafından aranmaya başlandığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onmousedown : Belge içerisindeki herhangi bir elemana mouse ile sol veya sağ tıklanıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onmouseenter : Belge içerisindeki herhangi bir elemanın mouse ile kapsama alanına giriş yapıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onmouseleave : Belge içerisindeki herhangi bir elemanın mouse ile kapsama alanından çıkış yapıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onmousemove : Belge içerisindeki herhangi bir elemanın mouse ile kapsama alanında gezildiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onmouseout : Belge içerisindeki herhangi bir elemanın mouse ile üzerinden gidildiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onmouseover : Belge içerisindeki herhangi bir elemanın mouse ile üzerine gelindiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onmouseup : Belge içerisindeki herhangi bir elemana mouse ile sol basılı duran tıklama bırakıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onmousewheel : Belge içerisindeki herhangi bir eleman üzerindeyken mouse'un tekerleği yukarı veya aşağı yuvarlandığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onoffline : Browser sayfayı çevrimiçi modda çalıştırırken, çevrimdışı modda çalıştırmaya başladığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ononline : Browser sayfayı çevrimdışı modda çalıştırırken, çevrimiçi modda çalıştırmaya başladığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onpagehide : Sayfa kullanıcı tarafından görüntülenemiyorsa belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onpageshow : Sayfa kullanıcı tarafından görüntülenebiliyorsa belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onpaste : Form veya belge içerisindeki bir elemanın metin içeriğine kullanıcı tarafından herhangi bir içerik yapıştırıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onpause : Medya elemanının yürütülmesi durdurulduğunda belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onplay : Medya elemanı yürütülmeye başlandığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onplaying : Medya elemanı yürütülmeye başlandığında ve yürütme işlemi arabelleğe yükleme vs. gibi işlemleri için durup başlayınca belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onprogress : Browser medya verisini indirirken belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onratechange : Medya elemanının yürütme hızı değiştirildiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onreset : Belge içerisindeki form sıfırlanmak istenirse belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onresize : Kullanıcı tarafından browser'ın sayfa boyutu değiştirilmek istendiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onscroll : Sayfanın veya herhangi bir HTML elemanının kaydırma çubuğu yukarı veya aşağı kaydırıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onsearch : Form içerisindeki arama alanı işleme alındığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onseeked : Medya elemanının yürütme konumu kullanıcı tarafından yeni konuma getirildiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onseeking : Medya elemanının yürütme konumu kullanıcı tarafından hareket ettirildiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onselect : Form içerisindeki bir elemanın metni seçilirse belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onstalled : Browser medya elemanı verisini indirmeye çalışırken medya verisini alamıyorsa belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onsubmit : Belge içerisindeki form gönderilmek istenirse belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onsuspend : Browser medya elemanı verisini bilerek indirmediğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ontimeupdate : Medya elemanının yürütülme konumu browser tarafından / kullanıcı tarafından değiştirildiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

ontoggle : Belge içerisindeki HTML details elemanının içeriği açıldığında veya kapandığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onunload : Sayfanın veya herhangi bir HTML elemanının kapatıldığında veya kaldırıldığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onvolumechange : Medya elemanının ses seviyesi değiştirildiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onwaiting : Medya elemanının yürütülmesi arabelleğe daha fazla veri yüklenmesi için duraklatıldığında ve devam etmesi beklendiğinde belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

onwheel : Belge içerisindeki herhangi bir eleman üzerindeyken mouse'un tekerleği yukarı veya aşağı yuvarlandığında belirtilecek olan fonksiyon çalışır. *(Document Object Model)*

open() : Yeni browser penceresi açmak için kullanılır. *(Browser Object Model)*

open() : Belge içerisinden yeni bir HTML çıktı akışı açmak için kullanılır. *(Document Object Model)*

opener : Yeni pencereyi oluşturan pencereye belirtilecek olan referans değerini döndürür. *(Browser Object Model)*

outerHeight : Pencerenin tüm alanının yükseklik değerini elde etmek için kullanılır. *(Browser Object Model)*

outerWidth : Pencerenin tüm alanının genişlik değerini elde etmek için kullanılır. *(Browser Object Model)*

pageX : Belge içerisindeki herhangi bir mouse event'ı gerçekleştiğinde browser penceresine / sayfaya göre mouse'un yatay konum koordinatının değerini geriye döndürür. *(Document Object Model)*

pageXOffset : Belge içeriğinin yatay kaydırma değerini elde etmek için kullanılır. *(Browser Object Model)*

pageY : Belge içerisindeki herhangi bir mouse event'ı gerçekleştiğinde browser penceresine / sayfaya göre mouse'un dikey konum koordinatının değerini geriye döndürür. *(Document Object Model)*

pageYOffset : Belge içeriğinin dikey kaydırma değerini elde etmek için kullanılır. *(Browser Object Model)*

parent : Geçerli çerçevenin üst penceresini elde etmek veya ayarlamak için kullanılır. *(Browser Object Model)*

parentElement : Belge içerisindeki herhangi bir HTML elemanının bir üst elemanını elde etmek için kullanılır. *(Document Object Model)*

parentNode : Belge içerisindeki herhangi bir node'un bir üst node'unu elde etmek için kullanılır. *(Document Object Model)*

parse() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine tarihin zaman damgası başlangıcı ile arasındaki zaman farkı bilgisini depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

parseFloat() : Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğini ondalıklı sayı haline dönüştürerek, yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür. *(Matematiksel işlem)*

parseInt() : Kendisine parametre olarak verilen değerler doğrultusunda değişken içeriğini tam sayı haline dönüştürerek, yeni bir içerik oluşturur ve oluşturduğu değeri geriye döndürür. *(Matematiksel işlem)*

pathname : Bir URL'in path bilgisi referansını elde etmek veya ayarlamak için kullanılır. *(Browser Object Model)*

PI : Pi sayısı değerini geriye döndürür. *(Matematiksel işlem)*

pixelDepth : Kullanıcının ekran çözünürlüğü için tanımlanmış olan her pixel başına düşen bit değerini elde etmek için kullanılır. *(Browser Object Model)*

platform : Browser'ın hangi platform üzerinde çalıştırıldığı bilgisi referansını elde etmek için kullanılır. *(Browser Object Model)*

plugins : Browser'daki tüm eklentilerin referansını elde etmek için kullanılır. Eklenti referansları birden fazla ise her bir eklenti döngü ile elde edilebilir. *(Document Object Model)*

pop() : Dizi içerisindeki en son elemanı siler ve sildiği elemanın değerini geriye döndürür. *(Diziler)*

POSITIVE_INFINITY : Atanabilecek en büyük pozitif sayı değerinden büyük sayılarda sonsuzluk değerini geriye döndürür. *(Matematiksel işlem)*

post : İstek gönderdiğinde veriler belirtilen kaynaktan alınarak işlenir. Bir diğer deyiş ile form verilerini veya manuel olarak belirleyeceğimiz verileri, veri gönderim tag'ları aracılığı ile kapalı olarak gönderim anlamına gelmektedir. *(HTML)*

pow() : Kendisine parametre olarak verilen iki sayının üssünü bularak, bulduğu değeri geriye döndürür. *(Matematiksel işlem)*

preventDefault() : Belge içerisinde tanımlı event'ın işlevini kaldırmak / iptal etmek / geçersiz kılmak için kullanılır. *(Document Object Model)*

previousElementSibling : Belge içerisindeki herhangi bir HTML elemanının içerisindeki aynı seviyede yani kardeş olan elemanlardan, belirtilecek olan elemandan önce gelen ilk elemanı elde etmek için kullanılır. *(Document Object Model)*

previousSibling : Belge içerisindeki herhangi bir HTML elemanının içerisindeki aynı seviyede yani kardeş olan node'lardan, belirtilecek olan node'dan önce gelen ilk node'u elde etmek için kullanılır. *(Document Object Model)*

print() : Pencere içeriğini yazdırmak için kullanılır. *(Browser Object Model)*

product : Browser'ın hangi görüntüleme motorunu kullandığı bilgisini elde etmek için kullanılır. *(Browser Object Model)*

prompt() : Kullanıcıdan bilgi talep etmek için girdi penceresi oluşturmak için kullanılır. *(Browser Object Model)*

propertyIsEnumerable : Bir nesnenin parametrik olarak girilen özelliğinin kullanılıp kullanılmadığını ve numaralandırılabilir olup olmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Nesneler)*

propertyIsEnumerable : Bir object'in parametrik olarak girilen özelliğinin kullanılıp kullanılmadığını ve bu özelliğin numaralandırılabilir olup olmadığını test etmek için kullanılır. *(Object Oriented Programming)*

protocol : Bir URL'in protokol adı referansını elde etmek için kullanılır. *(Browser Object Model)*

prototype : Bir nesnenin prototype nesnesine etki etmek için kullanılır *(Nesneler)*

push() : Dizinin sonuna eleman veya elemanlar eklenebilmesini sağlar. *(Diziler)*

querySelector() : CSS seçicileri ile belgedeki bir elemanın referansını elde etmek için kullanılır. *(Document Object Model)*

querySelectorAll() : CSS seçicileri ile belgedeki eleman veya elemanların referansını elde etmek için kullanılır. Eleman referansları birden fazla ise her bir eleman döngü ile elde edilebilir. *(Document Object Model)*

random() : 0 ile 1 arasında rastgele bir sayı üreterek, ürettiği değeri geriye döndürür. *(Matematiksel işlem)*

readyState : Belge içeriğinin browser tarafından yüklenme durumu değeri referansını elde etmek için kullanılır. *(Document Object Model)*

reduce() : Kendisine parametre olarak verilen fonksiyonu dizideki her eleman üzerinde soldan sağa tek tek olmak üzere çalıştırır ve tek bir değer oluşturarak, oluşturduğu değeri geriye döndürür. *(Diziler)*

reduceRight() : Kendisine parametre olarak verilen fonksiyonu dizideki her eleman üzerinde sağdan sola tek tek olmak üzere çalıştırır ve tek bir değer oluşturarak, oluşturduğu değeri geriye döndürür. *(Diziler)*

RegExp() : Düzenli ifade oluşturmak için kullanılır. *(Düzenli ifadeler)*

reload() : Browser penceresinde bulunan sayfayı yenilemek için kullanılır. *(Browser Object Model)*

removeAttribute() : Belge içerisindeki herhangi bir HTML elemanının belirtilen özelliğini kaldırmak / silmek için kullanılır *(Document Object Model)*

removeAttributeNode() : Belge içerisindeki herhangi bir HTML elemanına getAttributeNode() metodu ile belirtilmiş özellik node'unu, node object'inden kaldırmak / silmek için kullanılır. *(Document Object Model)*

removeChild() : Belge içerisindeki herhangi bir HTML elemanının içerdiği node'u kaldırmak / silmek için kullanılır. *(Document Object Model)*

removeEventListener() : Belgeye addEventListener() metodu ile eklenmiş olan event işaretleyicisini / dinleyicisini silmek için kullanılır. *(Document Object Model)*

removeNamedItem() : Belge içerisindeki HTML elemanına eklenmiş olan özellik node'unu kaldırmak için kullanılır. *(Document Object Model)*

repeat() : Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğini tekrar edip kopyalayarak, yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür. *(Değişkenler)*

replace() : Browser penceresinde bulunan sayfayı yeni bir sayfa ile değiştirmek için kullanılır. *(Browser Object Model)*

replace() : Kendisine verilecek olan parametreler doğrultusunda değişken içeriğini kontrol eder ve eşleşen değerlerin değişimin sağlayarak yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür. *(Değişkenler)*

replace() : Kendisine verilecek olan düzenli ifadeli parametreler doğrultusunda değişken içeriğini kontrol eder ve aranan değer eşleşmesi durumunda eşleşmeyi sağlayan değerlerin değişimin sağlayarak yeni bir içerik oluşturur ve oluşturduğu değeri geriye döndürür. *(Düzenli ifadeler)*

replaceChild() : Belge içerisindeki herhangi bir HTML elemanının içerdiği node'u değiştirmek için kullanılır. *(Document Object Model)*

resizeBy() : Pencerenin mevcut genişlik ve yükseklik değerini belirtilen pixel'lere göre yeniden boyutlandırmak için kullanılır. *(Browser Object Model)*

resizeTo() : Pencerenin mevcut genişlik ve yükseklik değerini belirtilen pixel'lere göre yeniden boyutlandırmak için kullanılır. *(Browser Object Model)*

return : Fonksiyondan değer döndürmek için kullanılır. *(Fonksiyonlar)*

return : Nesneden değer döndürmek için kullanılır. *(Nesneler)*

reverse() : Dizi içerisindeki sıralamayı tersine çevirerek yeni bir dizi oluşturur ve oluşturduğu değeri geriye döndürür. *(Diziler)*

rightContext : Değişken içeriğinde düzenli ifade ile aranılan değer eşleşmesi durumunda eşleşmeyi sağlayan değişken içeriği değerinin sağında kalan tüm değerleri geriye döndürür. *(Düzenli ifadeler)*

round() : Kendisine parametre olarak verilen herhangi bir ondalıklı sayı değerini kendisine en yakın tam sayıyı kontrol ederek aşağı veya yukarı yuvarlama yaparak sonucu geriye döndürür. *(Matematiksel işlem)*

screenLeft : Pencerenin bilgisayar ekranına göre yatay konum koordinatının değerini geriye döndürür. *(Browser Object Model)*

screenTop : Pencerenin bilgisayar ekranına göre dikey konum koordinatının değerini geriye döndürür. *(Browser Object Model)*

screenX : Pencerenin bilgisayar ekranına göre yatay konum koordinatının değerini geriye döndürür. *(Browser Object Model)*

screenX : Belge içerisindeki herhangi bir mouse event'ı gerçekleştiğinde bilgisayar ekranına göre mouse'un yatay konum koordinatının değerini geriye döndürür. *(Document Object Model)*

screenY : Pencerenin bilgisayar ekranına göre dikey konum koordinatının değerini geriye döndürür. *(Browser Object Model)*

screenY : Belge içerisindeki herhangi bir mouse event'ı gerçekleştiğinde bilgisayar ekranına göre mouse'un dikey konum koordinatının değerini geriye döndürür. *(Document Object Model)*

scripts : Belge içerisindeki script tag'ına sahip tüm elemanların referansını elde etmek için kullanılır. Eleman referansları birden fazla ise her bir eleman döngü ile elde edilebilir. *(Document Object Model)*

scrollBy() : Belge içeriğini belirtilen değere göre kaydırmak için kullanılır. *(Browser Object Model)*

scrollHeight : Belge içerisindeki herhangi bir HTML elemanının toplam dik kaydırma yüksekliği değerini elde etmek için kullanılır. *(Document Object Model)*

scrollLeft : Belge içerisindeki herhangi bir HTML elemanının yatay kaydırılma genişliği pixel değerini elde etmek için kullanılır. *(Document Object Model)*

scrollTo() : Belge içeriğini belirtilen değerlere göre kaydırmak için kullanılır. *(Browser Object Model)*

scrollTop : Belge içerisindeki herhangi bir HTML elemanının dikey kaydırılma genişliği pixel değerini elde etmek için kullanılır. *(Document Object Model)*

scrollWidth : Belge içerisindeki herhangi bir HTML elemanının toplam yatay kaydırma genişliği değerini elde etmek için kullanılır. *(Document Object Model)*

scrollX : Belge içeriğinin yatay kaydırma değerini elde etmek için kullanılır. *(Browser Object Model)*

scrollY : Belge içeriğinin dikey kaydırma değerini elde etmek için kullanılır. *(Browser Object Model)*

search : Bir URL'in sorgu dizesi bilgisi referansını elde etmek veya ayarlamak için kullanılır. *(Browser Object Model)*

search() : Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğini arayarak ilk eşleşmeyi sağlayan değişken içeriğinin indis / index değerini geriye döndürür. *(Değişkenler)*

search() : Kendisine parametre olarak verilen düzenli ifadeli değer doğrultusunda değişken içeriğini kontrol eder ve ilk eşleşmeyi sağlayan değişken içeriğinin indis / index değerini geriye döndürür. *(Düzenli ifadeler)*

self : İçerisinde bulunan pencerenin en üst seviye pencere olup olmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Browser Object Model)*

setAttribute() : Belge içerisindeki herhangi bir HTML elemanına belirtilen özelliği ve değerini eklemek için kullanılır. *(Document Object Model)*

setAttributeNode() : Belge içerisindeki herhangi bir HTML elemanına createAttribute() metodu ile oluşturulan özellik node'unu ve değerini, node object'ine uygulamak için kullanılır. *(Document Object Model)*

setDate() : Kendisine parametre olarak belirtilmiş değer doğrultusunda değişken içeriğine ayın gün bilgisini 1 ila 31 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setFullYear() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine yıl bilgisini 1000 ila 9999 arasında olmak üzere rakamsal olarak ayarlayarak depolayarak ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setHours() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine günün saat bilgisini 0 ila 23 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setInterval() : Belge içerisinde tanımlanan herhangi bir kodlamayı belirtilen süre aralıklarında tekrar tekrar çalıştırmak için kullanılır. *(Zamanlayıcı)*

setMilliseconds() : Kendisine parametre olarak belirtilmiş değer doğrultusunda saniyenin milisaniye bilgisini 0 ila 999 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setMinutes() : Kendisine parametre olarak belirtilmiş değer doğrultusunda değişken içeriğine saatin dakika bilgisini 0 ila 59 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setMonth() : Kendisine parametre olarak belirtilmiş değer doğrultusunda değişken içeriğine yılın ay bilgisini 0 ila 11 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setNamedItem() : Belge içerisindeki HTML elemanına özellik node'u ve değeri eklemek için kullanılır. *(Document Object Model)*

setSeconds() : Kendisine parametre olarak belirtilmiş değer doğrultusunda değişken içeriğine dakikanın saniye bilgisini 0 ila 59 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setTime() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine zaman damgasına ait tarihin bilgisini depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setTimeout() : Belge içerisinde tanımlanan herhangi bir kodlamayı belirtilen süre sonunda sadece bir defa olmak üzere çalıştırmak için kullanılır. *(Zamanlayıcı)*

setUTCDate() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait ayın gün bilgisini 1 ila 31 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setUTCFullYear() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait yıl bilgisini 1000 ila 9999 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setUTCHours() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait günün saat bilgisini 0 ila 23 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setUTCMilliseconds() : Kendisine parametre olarak belirtilmiş değer doğrultusunda değişken içeriğine evrensel zaman dilimine ait saniyenin milisaniye bilgisini 0 ila 999 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setUTCMinutes() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait saatin dakikasını 0 ila 59 arasında olmak üzere rakamsal olarak ayarlayarak depolayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setUTCMonth() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait yılın ay bilgisini 0 ila 11 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

setUTCSeconds() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait dakikanın saniye bilgisini 0 ila 59 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

shift() : Dizi içerisindeki ilk elemanı siler ve sildiği elemanın değerini geriye döndürür. *(Diziler)*

shiftKey : Belge içerisindeki herhangi bir event tetiklenirken shift tuşuna basılıp basılmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Document Object Model)*

shuffle() : Dizinin her elemanı üzerinde tek tek çalışarak tüm elemanları karıştırır ve yeni bir dizi oluşturarak oluşturduğu değeri geriye döndürür. *(Diziler)*

sign() : Kendisine parametre olarak verilen herhangi bir sayı değerinin işaretini bularak sonucu geriye döndürür. *(Matematiksel işlem)*

sin() : Kendisine parametre olarak verilen herhangi bir sayı değerinin sinüsünü bularak, bulduğu değeri geriye döndürür. *(Matematiksel işlem)*

sinh() : Kendisine parametre olarak verilen herhangi bir sayı değerinin hiperbolik sinüsünü bularak, bulduğu değeri geriye döndürür. *(Matematiksel işlem)*

slice() : Kendisine verilecek olan parametreler ile tanımlanmış değişken içeriği aralığında bulunan değerleri kopyalayarak yeni bir içerik oluşturur ve oluşturduğu değeri geriye döndürür. *(Değişkenler)*

slice() : Kendisine verilecek olan parametreler ile tanımlanmış dizi aralığında bulunan değerleri kopyalayarak yeni bir dizi oluşturur ve oluşturduğu dizinin değerini geri döndürür. *(Diziler)*

small() : Değişken içerisindeki metni küçük fontlu metin yaparak değerini geriye döndürür. *(Değişkenler)*

some() : Kendisine parametre olarak belirtilen fonksiyon true değeri döndürene kadar dizi elemanları üzerinde sıra ile gezinilmesini sağlar. *(Diziler)*

sort() : Dizi içeriğinin sıralanmasını sağlar. Metot Unicode olarak çalışır. *(Diziler)*

source : Düzenli ifade içerisinde kullanılan desen değerini geriye döndürür. *(Düzenli ifadeler)*

specified : Belge içerisindeki HTML elemanına herhangi bir özelliğin eklenip eklenmediği referansını boolean veri türünde elde etmek için kullanılır. *(Document Object Model)*

spellcheck : Nesne içeriğinin dil bilgisi ve yazım kurallarının denetlenmesini sağlar. *(HTML Global Attributes)*

splice() : Diziden eleman veya elemanlar silinebilmesini sağlar ve sildiği eleman veya elemanların değerini geri döndürür. Ayrıca diziye eleman veya elemanlar eklenebilmesini sağlar. *(Diziler)*

split() : Kendisine parametre olarak tanımlanmış değer doğrultusunda değişken içeriğini bölümleyerek yeni bir dizi oluşturur ve oluşturduğu değeri geriye döndürür. *(Değişkenler)*

split() : Kendisine parametre olarak verilen düzenli ifadeli değer doğrultusunda değişken içeriğini kontrol eder ve aranan değer eşleşmesi durumunda eşleşmeyi sağlayan değişken içeriği değerlerini bölümleyerek yeni bir dizi oluşturur ve oluşturduğu dizi değerini geriye döndürür. *(Düzenli ifadeler)*

sqrt() : Kendisine parametre olarak verilen herhangi bir sayı değerinin karekökünü bularak, bulduğu değeri geriye döndürür. *(Matematiksel işlem)*

SQRT1_2 : 1/2 sayısının karekök değerini geriye döndürür. *(Matematiksel işlem)*

SQRT2 : 2 Sayısının karekök değerini geriye döndürür. *(Matematiksel işlem)*

startsWith() : Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğinin başlangıç değerini kontrol ederek Boolean veri türünde sonucu geriye döndürür. *(Değişkenler)*

stop() : Pencerenin yüklenmesini durdurmak için kullanılır. *(Browser Object Model)*

strike() : Değişken içerisindeki metni üzeri çizili metin yaparak değeri geriye döndürür. *(Değişkenler)*

String() : Değişken içeriğini mutlak olarak string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Değişkenler)*

String() : Değişken içeriğini mutlak olarak string veri türüne çevirerek oluşturduğu değeri geriye döndürür. *(Matematiksel işlem)*

style : Belge içerisindeki HTML elemanına veya elemanlarına CSS özellikleri ve değerleri eklemek veya referansını elde etmek için kullanılır. *(Document Object Model)*

style : Nesneye satır içi CSS tanımlar. *(HTML Global Attributes)*

styleSheets : Belge üzerinde etkisi olan tüm stil sayfalarının nesne referansını elde etmek için kullanılır. Stil sayfası referansları birden fazla ise her stil sayfası referansı döngü ile elde edilebilir. *(Document Object Model)*

sub() : Değişken içerisindeki metni alt simge metni yaparak değerini geriye döndürür. *(Değişkenler)*

substr() : Kendisine verilecek olan parametreler ile tanımlanmış değişken içeriği aralığında bulunan değerleri kopyalayarak yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür. *(Değişkenler)*

substring() : Kendisine verilecek olan parametreler ile tanımlanmış değişken içeriği aralığında bulunan değerleri kopyalayarak yeni bir içerik oluşturur ve oluşturduğu değeri geriye döndürür. *(Değişkenler)*

sup() : Değişken içindeki metni üst simge metni yaparak değerini geriye döndürür. *(Değişkenler)*

switch : Koşul başlatmak için kullanılır. *(Koşul & Kontrol)*

tabIndex : Belge içerisindeki herhangi bir HTML elemanının tabindex değerini elde etmek veya ayarlamak için kullanılır. *(Document Object Model)*

tabindex : Nesnelerin sekme sırasını tanımlar. *(HTML Global Attributes)*

tagName : Belge içerisindeki herhangi bir HTML elemanının tag adını elde etmek için kullanılır. *(Document Object Model)*

tan() : Kendisine parametre olarak verilen herhangi bir sayının tanjantını bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

tanh() : Kendisine parametre olarak verilen herhangi bir sayının hiperbolik tanjantını bularak, bulunduğu değeri geriye döndürür. *(Matematiksel işlem)*

target : Belge içerisinde tanımlı event'ı tetikleyen herhangi bir HTML elemanının tag adı referansını elde etmek için kullanılır. *(Document Object Model)*

ternary : Koşul oluşturmak ve sonuçlarını döndürmek için kullanılır. *(Koşul & Kontrol)*

test() : Kendisine parametre olarak verilen değişken içeriğinde düzenli ifade ile aranan değer eşleşmesinin olup olmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Düzenli ifadeler)*

textContent : Belge içerisindeki herhangi bir node'un, text node'larının değerini elde etmek veya ayarlamak için kullanılır. *(Document Object Model)*

this : Oluşturulmuş olan bir object'i işaret etmek için kullanılır. *(Özel operatör)*

throw : Hata oluşturabilecek kodlar için istisnai durumlarda kullanılır. *(Koşul & Kontrol)*

timeStamp : Belge içerisinde tanımlı event'ın oluşturulduğu zaman ile çalıştırıldığı zaman arasındaki fark süresinin mikro saniye olarak referansını elde etmek için kullanılır. *(Document Object Model)*

title : Belge içerisindeki title tag'ının referansını elde etmek için kullanılır. *(Document Object Model)*

title : Belge içerisindeki herhangi bir HTML elemanının title değerini elde etmek veya ayarlamak için kullanılır. *(Document Object Model)*

title : Nesneye başlık tanımlar. *(HTML Global Attributes)*

toDateString() : Kullanıldığı değişken içeriğine atanmış olan tarih ve saat bilgilerinin sadece tarih kısmını alarak string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Tarih saat işlemi)*

toExponential() : Kendisine parametre olarak belirlenen değer doğrultusunda değişken içeriğindeki sayının ondalık hanesinde kaç basamak olacağını belirleyerek, kalan değeri üstel hale getirerek yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür. *(Matematiksel işlem)*

toFixed() : Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğindeki sayının ondalık hanesinde kaç basamak olacağını belirleyerek yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür. *(Matematiksel işlem)*

toISOString() : Kullanıldığı değişken içeriğine atanmış olan tarih ve saat bilgilerini ISO standartını kullanarak string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Tarih saat işlemi)*

toJSON() : Kullanıldığı değişken içeriğine atanmış olan tarih ve saat bilgilerini JSON formatını kullanarak string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Tarih saat işlemi)*

toLocaleDateString() : Kullanıldığı değişken içeriğine atanmış olan tarih ve saat bilgilerinin tarayıcı lokasyonuna / diline göre sadece tarih kısmını alarak string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Tarih saat işlemi)*

toLocaleLowerCase() : Değişken içerisindeki metnin tamamını tarayıcı lokasyonuna / dili göre küçük harfe dönüştürerek değeri geriye döndürür. *(Değişkenler)*

toLocaleString : Bir object'in karakter dizesi olarak temsil eden halini döndürmek için kullanılır. *(Object Oriented Programming)*

toLocaleString() : Nesnenin tarayıcı lokasyonuna / diline göre karakter dizesi olarak temsil edilen halinin değerini geriye döndürür. Nesne değeri üzerinden tarayıcı lokasyonuna / diline göre özel işlemler yapılabilir. *(Nesneler)*

toLocaleString() : Kullanıldığı değişken içeriğine atanmış olan tarih ve saat bilgilerini tarayıcı lokasyonuna / diline göre alarak string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Tarih saat işlemi)*

toLocaleTimeString() : Kullanıldığı değişken içeriğine atanmış olan tarih ve saat bilgilerinin tarayıcı lokasyonuna / diline göre sadece saat kısmını alarak string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Tarih saat işlemi)*

toLocaleUpperCase() : Değişken içerisindeki metnin tamamını tarayıcı lokasyonuna / diline göre büyük harfe dönüştürerek değeri geriye döndürür. *(Değişkenler)*

toLowerCase() : Değişken içerisindeki metnin tamamını küçük harfe dönüştürerek değeri geriye döndürür. *(Değişkenler)*

top : İçerisinde bulunulan pencerenin en üst seviye pencere olup olmadığını kontrol ederek boolean veri türünde sonucu geriye döndürür. *(Browser Object Model)*

toFixed() : Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğindeki sayının kaç basamak olacağını belirleyerek yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndür. *(Matematiksel işlem)*

toSource : Bir object'in kaynak kodunu temsil eden halini döndürmek için kullanılır. *(Object Oriented Programming)*

toSource() : Nesnenin kaynak kodunu temsil eden halinin değerini geriye döndürür. *(Nesneler)*

toString : Bir object'in karakter dizesi olarak temsil eden halini döndürmek için kullanılır. *(Object Oriented Programming)*

toString() : Değişken içeriğini string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Değişkenler)*

toString() : Dizideki tüm elemanları içeren string veri türünde düz bir metin oluşturur ve oluşturduğu değeri geriye döndürür. *(Diziler)*

toString() : Düzenli ifade içeriğini string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Düzenli ifadeler)*

toString() : Değişken içeriğini string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Matematiksel işlem)*

toString() : Nesnenin karakter dizesi olarak temsil edilen halinin değerini geriye döndürür. Nesne prototype özelliği kullanılarak değer ataması yapılabilir. *(Nesneler)*

toString() : Kullanıldığı değişken içeriğine atanmış olan tarih ve saat bilgilerini alarak string veri türünde düz bir metne dönüştürerek, dönüştürdüğü değeri geriye döndürür. *(Tarih saat işlemi)*

toTimeString() : Kullanıldığı değişken içeriğine atanmış olan tarih ve saat bilgilerinin sadece saat kısmını alarak string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Tarih saat işlemi)*

toUpperCase() : Değişken içerisindeki metnin tamamını büyük harf yaparak değeri geriye döndürür. *(Değişkenler)*

toUTCString() : Kullanıldığı değişken içeriğine atanmış olan tarih ve saat bilgilerini evrensel zaman diliminde alarak string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür. *(Tarih saat işlemi)*

translate : Nesnenin başka dile çevrilip çevrilmeyeceğini tanımlar. *(HTML Global Attributes)*

trim() : Değişken içeriğinin varsa başındaki ve sonundaki tüm boşlukları temizleyerek değeri geriye döndürür. *(Değişkenler)*

trunc() : Kendisine parametre olarak verilen herhangi bir ondalıklı sayı değerinin ondalık kısmını kaldırarak, tam sayı değerini geriye döndürür. *(Matematiksel işlem)*

try : Hata oluşturabilecek kodlar için kullanılır. *(Koşul & Kontrol)*

type : Belge içerisinde tanımlı herhangi bir event türü adının referansını elde etmek için kullanılır. *(Document Object Model)*

typeof : Değişkenin veri türünü geri döndürür. *(Değişkenler)*

typeof : Herhangi bir ifadenin veri türünü bildirmek için kullanılır. *(Özel operatör)*

unescape() : Değişken içerisindeki metnin özel karakterlerinin Unicode kodlamasına dönüştürülmüş karakterlerini çözerek değeri geriye döndürür. *(Değişkenler)*

unshift() : Dizinin başına eleman veya elemanlar eklenebilmesini sağlar. *(Diziler)*

unwatch : Belirtilen bir özelliğinin değeri değiştiğinde, eklenmiş olan herhangi bir işlevi kaldırmak için kullanılır. *(Object Oriented Programming)*

unwatch() : Nesnenin belirtilen bir özelliği değiştiğinde, eklenmiş olan bir işlevi kaldırmak için kullanılır. *(Nesneler)*

URL : Belgenin tam URL adresi referansını elde etmek için kullanılır. *(Document Object Model)*

userAgent : Browser'ın tam model referansını elde etmek için kullanılır. *(Browser Object Model)*

UTC() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait tarihin zaman damgası bilgisini depolayarak, depoladığı değeri geriye döndürür. *(Tarih saat işlemi)*

value : Belge içerisindeki HTML elemanına eklenmiş olan özellik değerinin referansını elde etmek için kullanılır. HTML elemanı içerisindeki özellik sayısı birden fazla ise her bir özellik değeri referansı döngü ile elde edilebilir. *(Document Object Model)*

valueOf : Bir object'in temel değerini elde etmek için kullanılır. *(Object Oriented Programming)*

valueOf() : Üzerinde çalışılan değişken değerinin tam anlamıyla aktarımını sağlayarak yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür. *(Değişkenler)*

valueOf() : Üzerinde çalışılan dizi değerinin tam anlamıyla aktarımını sağlayarak yeni bir dizi oluşturur ve oluşturduğu dizi değerini geriye döndürür. *(Diziler)*

valueOf() : Üzerinde çalışılan değişken değerinin tam anlamıyla aktarımını sağlayarak yeni bir içerik oluşturur ve oluşturduğu değeri geriye döndürür. *(Matematiksel işlem)*

valueOf() : Nesnenin temel değerini elde etmek için kullanılır. *(Nesneler)*

valueOf() : Üzerinde çalışılan değişken değerinin tam anlamıyla aktarımını sağlayarak yeni bir içerik oluşturur ve oluşturduğu değeri geriye döndürür. *(Tarih saat işlemi)*

var : Standart değişken tanımlamakta kullanılır. *(Değişkenler)*

void : Herhangi bir ifadenin geriye değer döndürmesini engellemek için kullanılır. *(Özel operatör)*

watch : Belirtilen bir özelliğinin değeri değiştiğinde, çalıştırılacak herhangi bir işlevi eklemek için kullanılır. *(Object Oriented Programming)*

watch() : Nesnenin belirtilen bir özelliği değiştiğinde, çalıştırılacak olan herhangi bir işlev eklemek için kullanılır. *(Nesneler)*

which : Belge içerisindeki onkeypress, onkeydown veya onkeyup event'ı gerçekleştiğinde klavyeden basılan tuşun unicode kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür. *(Document Object Model)*

while : Döngü oluşturmak için kullanılır. *(Döngü)*

width : Kullanıcının toplam ekran genişliği değerini elde etmek için kullanılır. *(Browser Object Model)*

write() : Dokümana çıktı yapar. *(Document Object Model) (Doküman Çıktısı)*

writeln() : Dokümana yeni satırbaşı yaparak çıktı yapar. *(Document Object Model) (Doküman Çıktısı)*

200 : Server, isteği başarıyla işlemiştir ve genellikle bu, server'ın istenen sayfayı sağladığı anlamına gelir. *(Http durum kodu)*

301 : İstenen sayfa kalıcı olarak yeni bir konuma taşındı anlamını ifade etmektedir. *(Http durum kodu)*

302 : Geçici yönlendirme çeşidi olarak bilinir. Eğer bir sayfa adresini arama motorunda göstermek isteyip kullanıcının eriştiği yerde başka adres kullanmak istiyorsanız bu yöntemi kullanabilirsiniz. *(Http durum kodu)*

404 : Server istenen sayfayı bulamıyor. *(Http durum kodu)*

500 : Server hatayla karşılaştı ve isteği gerçekleştiremiyor. *(Http durum kodu)*

502 : Server hatayla karşılaştı ve bad gateway sorunu. *(Http durum kodu)*

503 : Server şu anda kullanılamıyor anlamına gelir. *(Http durum kodu)*

- : Değerler arasında çıkartma işlemi için kullanılır. *(Aritmetik işlem operatörü)*

-- : Değerin önüne veya arkasına eklenerek değerın önceden veya sonradan azaltılması işlemi için kullanılır. *(Aritmetik işlem operatörü)*

! : Operandın önüne eklenerek, eklendiği operandın mantıksal olarak ters değerini döndürür. *(Mantıksal operatör)*

!= : Değerler arasındaki eşitsizlik sınaması için kullanılır. *(Karşılaştırma operatörü)*

!== : Değerler arasındaki denk olmama sınaması için kullanılır. *(Karşılaştırma operatörü)*

\$: Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminin sonu ile düzenli ifade arasında bir eşleşme aramak için kullanılır. *(Düzenli ifadeler)*

% : İlk değerın ikinci değere bölünmesi sonucunda kalan değeri elde etmek için kullanılır. *(Aritmetik işlem operatörü)*

%= : Değişkene değer / veri atamak için kullanılır. *(Atama operatörü)*

& : Operandların sayısal değerlerinin bitset şablonları üzerinde karşılıklı olarak bitset karşılaştırma için kullanılır. *(Bit işlem operatörü)*

&& : İki veya daha fazla operandın oluşturduğu şartlar ayrı ayrı değerlendirilerek oluşan boolean veri türündeki değerler incelenir. *(Mantıksal operatör)*

&= : Değişkene değer / veri atamak için kullanılır. *(Bitset atama operatörü)*

() : Düzenli ifadelerde grup tanımlamak için kullanılır. *(Düzenli ifadeler)*

(?:) : Düzenli ifadelerde alt grup oluşturmak için kullanılır. *(Düzenli ifadeler)*

(x|y) : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin, belirtilen x yada y karakterlerinden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

* : Değerler arasında çarpma işlemi için kullanılır. *(Aritmetik işlem operatörü)*

* : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin 0 defa yada daha fazla tekrarlanmış olması gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

****** : İlk değeri ikinci değer kadar üssü işlemine tabi tutmak için kullanılır. *(Aritmetik işlem operatörü)*

****=** : Değişkene değer / veri atamak için kullanılır. *(Atama operatörü)*

***=** : Değişkene değer / veri atamak için kullanılır. *(Atama operatörü)*

, : Birbiri ardına sırayla çalışacak deyimler bildirmek için kullanılır. *(Özel operatör)*

. : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, yeni satır karakteri hariç herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

/ : İlk değeri ikinci değere bölme işlemi için kullanılır. *(Aritmetik işlem operatörü)*

/* ... */ : Çok satılı yorum / gizli açıklama metinleri eklemek için kullanılır. *(Javascript genel)*

// : Tek satılı yorum / gizli açıklama metni eklemek için kullanılır. *(Javascript genel)*

/= : Değişkene değer / veri atamak için kullanılır. *(Atama operatörü)*

? : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer 0 defa yada 1 defa tekrarlanmış olması gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

?! : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter dizilimi içerisindeki belirtilen bir referans değer ile takip edilmeyen değerleri düzenli ifade ile eşleşebilecek değerleri aramak için kullanılır. *(Düzenli ifadeler)*

?: : Koşullu operatör aynı zamanda ternary operatörü olarak da adlandırılır. Program akışında bir şarta bağlı olarak iki ifadeden birisi çalıştırılmak istendiği durumlarda kullanılır. *(Özel operatör)*

?= : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter dizilimi içerisindeki belirtilen bir referans değer önünde düzenli ifade ile eşleşebilecek değerleri aramak için kullanılır. *(Düzenli ifadeler)*

[^abc] : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, belirtilen karakterler dışındaki karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

[^a-z] : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, belirtilen karakter aralığı dışındaki karakterlerin herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

[abc] : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, belirtilen karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

[a-z] : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, belirtilen karakter aralığından herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

**** : Düzenli ifadelerde ardından gelecek olan karakterin özel bir karakter olup olmadığını belirtmek için kullanılır. *(Düzenli ifadeler)*

**** : Kaçırma operatörü kullanım esnasında karakterin önüne yazıldığında ilgili karaktere işlev kazandırılması için kullanılır. *(Özel operatör)*

\0 : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, sadece Null karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\000 : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, octal numeral system tabanında ASCII karşılığı olan karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\b : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriğinin düzenli ifadeye uygun olabilecek metin parçalarını, karakter dizilimi içerisindeki kelimelerin başında yada sonunda aramak için kullanılır. *(Düzenli ifadeler)*

\B : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriğinin düzenli ifadeye uygun olabilecek metin parçalarını, karakter dizilimi içerisindeki kelimelerin içerisinde aramak için kullanılır. *(Düzenli ifadeler)*

\d : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, sadece rakam karakterlerinin herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\D : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, rakam karakterleri dışındaki karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\f : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, sadece form besleme karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\n : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, sadece yeni satır karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\r : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, sadece paragraf sonu karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\s : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, sadece boşluk karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\S : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, boşluk karakteri dışındaki karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\t : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, sadece yatay sekme karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\u0000 : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, 16 Bit'lik Unicode formatındaki karşılığı olan karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\v : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, sadece dikey sekme karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\w : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, harf, rakam veya _ karakterlerinden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\W : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, harf, rakam veya _ karakterleri dışındaki karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

\x00 : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değer, hexadecimal numeral system tabanında ASCII karşılığı olan karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

^ : Operandların sayısal değerlerinin bitset şablonları üzerinde karşılıklı olarak bitset karşılaştırma için kullanılır. *(Bit işlem operatörü)*

^ : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminin başlangıcı ile düzenli ifade arasında eşleşme aramak için kullanılır. *(Düzenli ifadeler)*

^= : Değişkene değer / veri atamak için kullanılır. *(Bitset atama operatörü)*

__proto__ : Bir nesnenin prototype nesnesini elde etmek için kullanılır ve prototype nesnesi içerisindeki özelliklere ulaşarak değeri geriye döndürür. *(Nesneler)*

{x,} : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin x defa yada daha fazla tekrarlanmış olması gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

{x,y} : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin en az x defa yada en fazla y defa tekrarlanmış olması gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

{x} : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin x defa tekrarlanmış olması gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

| : Operandların sayısal değerlerinin bitset şablonları üzerinde karşılıklı olarak bitset karşılaştırma için kullanılır. *(Bit işlem operatörü)*

|| : İki veya daha fazla operandın oluşturduğu şartlar ayrı ayrı değerlendirilerek oluşan boolean veri türündeki değerler incelenir. *(Mantıksal operatör)*

|= : Değişkene değer / veri atamak için kullanılır. *(Bitset atama operatörü)*

~ : Eklendiği operandın sayısal değerinin bitset şablonu üzerinde bitset değerinin mantıksal olarak ters değerini döndürür. *(Bit işlem operatörü)*

+ : Değerler arasında toplama işlemi için kullanılır. *(Aritmetik işlem operatörü)*

+ : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin 1 defa yada daha fazla tekrarlanmış olması gerektiğini belirtmek için kullanılır. *(Düzenli ifadeler)*

+ : Birden fazla ifadeyi birleştirmek için kullanılır. *(Özel operatör)*

++ : Değerin önüne veya arkasına eklenerek değerın önceden veya sonradan arttırılması işlemi için kullanılır. *(Aritmetik işlem operatörü)*

+= : Değişkene değer / veri atamak için kullanılır. *(Atama operatörü)*

< : İlk değerin ikinci değerden küçük olma sınaması için kullanılır. *(Karşılaştırma operatörü)*

<!-- ... --> : yorum / gizli açıklama metinleri eklemek için kullanılır. *(HTML)*

<< : İlk operandın sayısal değerinin bitset şablonu üzerinde ikinci operand ile belirlenecek değer kadar sola kaydırma işlemi yapar. *(Bit işlem operatörü)*

<= : Değişkene değer / veri atamak için kullanılır. *(Bitset atama operatörü)*

<= : İlk değerin ikinci değerden küçük olma sınaması veya değerler arasında eşitlik sınaması için kullanılır. *(Karşılaştırma operatörü)*

= : Değişkene değer / veri atamak için kullanılır. *(Atama operatörü)*

-= : Değişkene değer / veri atamak için kullanılır. *(Atama operatörü)*

== : Değerler arasında eşitlik sınaması için kullanılır. *(Karşılaştırma operatörü)*

=== : Değerler arasında denklik sınaması için kullanılır. *(Karşılaştırma operatörü)*

> : İlk değerin ikinci değerden büyük olma sınaması için kullanılır. *(Karşılaştırma operatörü)*

>= : İlk değerin ikinci değerden büyük olma sınaması veya değerler arasında eşitlik sınaması için kullanılır. *(Karşılaştırma operatörü)*

>> : İlk operandın sayısal değerinin bitset şablonu üzerinde ikinci operand ile belirlenecek değer kadar sağa kaydırma işlemi yapar. *(Bit işlem operatörü)*

>>= : Değişkene değer / veri atamak için kullanılır. *(Bitset atama operatörü)*

>>> : İlk operandın sayısal deęerinin bitset řablonu üzerinde ikinci operand ile belirlenecek deęer kadar saęa kaydırma işlemleri yapar. (*Bit işlem operatörü*)

>>>= : Deęişkene deęer / veri atamak için kullanılır. (*Bitset atama operatörü*)