

Derleyen Düzenleyen :

SERCAN LÜLLECİ (Matematik Mezunu)...

JAVASCRIPT

DEĞİŞKENLER İÇİN ÖN TANIMLI METODLARI:

- **Typeof()** = Değişkenin veri türünü geri döndürür.
- **Length()** = Değişkenin içeriğinin karakter sayısını geriye döndürür.İndex sayısını verir.
- **eval()** = Kendisine paremetre olarak verilen değerleri Javascript kodlamasına çevirerek komut gibi çalıştırır ve oluşan değeri geriye döndürür.
- **Trim()** = Değişken içeriğinin varsa başındaki ve sonundaki boşlukları temizleyerek değeri geriye döndürür.
- **Slice()** = Kendisine verilecek olan parametreler ile tanımlanmış değişken içeriği aralığında bulunan değerleri kopyalayarak yeni bir içerik oluşturur ve oluşturulan değeri geriye döndürür.
- **substring()** = Kendisine verilecek olan parametreler ile tanımlanmış değişken içeriği aralığında bulunan değerleri kopyalarayarak yeni bir içerik oluşturur ve oluşturduğu değeri geriye döndürür.
- **substr()** = Kendisine verilecek olan parametreler ile tanımlanmış değişken içeriği aralığında bulunan değerleri kopyalayarak yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür.
- **concat()** = Üzerinde çalışılan değişkene parametrik olarak verilen değişken veya değişkenleri dahil ederek yeni bir içerik oluşturur.Oluşturduğu değeri geriye döndürür.
- **repeat()** = Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğini tekrar edip kopyalayarak,yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür.
- **to string()** = Değişken içeriğini string (metin) veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür.
- **String()** = Değişken içeriğini mutlak olarak string veri türünde düz bir metne dönüştürerek oluşturduğu değeri geriye döndürür.
- **Number()** = Değişken içeriğini mutlak olarak Number (sayı) veri türüne dönüştürerek oluşturduğu değeri geriye döndürür.
- **valueOf()** = Üzerinde çalışılan değişken değerinin tam anlayla aktarımını sağlayarak yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür.
- **indexOf()** = Değişken içerisinde parametrik olarak verilen değeri arar ve ilk eşleşmeyi sağlayan değişken içeriğinin indis/index değerini geriye döndürür.
- **lastIndexOf()** = Değişken içerisinde parametrik olarak verilen değeri arar ve son eşleşmeyi sağlayan değişken içeriğinin indis/index değerini geriye döndürür.
- **Search()** = Kendisine parametre olarak verilen değer doğrultusunda değişkeni içeriğini arayarak ilk eşleşmeyi sağlayan değişkenin içeriğinin indis/index değerini geriye döndürür.(Düzenli ifade alabilir).

- **Includes()** = Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğini kontrol ederek Boolean (mantıksal) veri türünde sonucu geriye döndürür.
- **startWith()** = Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğinin başlangıç değerini kontrol ederek Boolean (Mantıksal) veri türünde sonucu geriye döndürür.
- **endWith()** = Kendisine verilen parametre doğrultusunda değişken içeriğinin bitiş değerini kontrol ederek Boolean (mantıksal) veri türünde sonucu geriye döndürür.
- **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.(Düzenli ifade alabilir.)
- **Replace()** = Kendisine verilecek parametreler doğrultusunda değişken içeriğini kontrol eder ve eşleşen değerlerin değişimin sağlayarak yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür.(Düzenli ifade alabilir.) Not: 2 parametre alır.1.parametre değişecek içerik.2.parametre güncel içerik..
- **Split()** = Kendisine parametre olarak tanımlanmış değer doğrultusunda değişken içeriğini bölümleyerek yeni bir dizi oluşturur ve oluşturduğu değeri geriye döndürür.
- **localeCompare()** = Üzerinde çalışılan değişkene parametrik olarak verilen diğer bir değişken içeriğini tarayıcı diline göre karşılaştırarak sonucunu geriye döndürür.(Unicode-Evrensel kod)

-1 = Birinci değişken içeriği,ikinci değişken içeriğinden önce geliyorsa...

0 = Birinci değişken içeriği,ikinci değişken içeriğine eşitse...

1 = Birinci değişken içeriği,ikinci değişken içeriğinden sonra geliyorsa...

- **charAt()** = Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğindeki karşılığı olan karakteri bularak geriye döndürür.
- **charCodeAt()** = Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğindeki karşılığı olan karakteri bularak,karakterin Unicode (evrensel kod) değerini geriye döndürür.
- **formCharCode()** = Kendisine parametre olarak verilen Unicode (Evrensel kod) değerini karakter olarak geriye döndürür.Çoklu değer alabilir.
- **encodeURIComponent()** = Değişken içerisindeki metnin özel karakterlerini UTF-8 (Unicode dönüşüm formatı) kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür.
- **decodeURI()** = Değişken içerisindeki metnin özel karakterlerinin UTF-8 (Unicode dönüşüm formatı) kodlamasına dönüştürülmüş karakterini çözerek değeri geriye döndürür.
- **encodeURIComponentComponent()** = Değişken içerisindeki metnin özel karakterlerini UTF-8 (Unicode dönüşüm formatı) kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür.
- **decodeURIComponent()** = Değişken içerisindeki metnin özel karakterlerinin UTF-8 kodlamasına dönüştürülmüş karakterlerini çözerek değeri geriye döndürür.
- **Escape()** = Değişken içerisindeki metnin özel karakterlerini Unicode (Evrensel kod) kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür.

- **Unescape()** = Değişken içerisindeki metnin özel karakterlerinin Unicode (evrensel kod) dönüştürülmüş karakterlerini çözerek değeri geriye döndürür.
- **toUpperCase()** = Değişken içerisindeki metnin tamamını büyük harf yaparak değeri geriye döndürür.
- **toLowerCase()** = Değişken içerisindeki metnin tamamını küçük harfe dönüştürerek değeri geriye döndürür.
- **toLocaleUpperCase()** = Değişken içerisindeki metnin tamamını tarayıcı lokasyonuna / diline göre büyük harfe dönüştürerek değeri geriye döndürür.
- **toLocaleLowerCase()** = Değişken içerisindeki tamamını tarayıcı lokasyonuna / dili göre küçük harfe dönüştürerek değeri geriye döndürür.
- **sub()** = Değişken içerisindeki metni alt simge metni yaparak değerini geriye döndürür.Örnek= H2O metnini 2 tabanda yazar.
- **sup()** = Değişken içerisindeki metni üst simge yaparak değerini geriye döndürür.Örnek= m2 metnin m kare olarak yazar.
- **big()** = Değişken içerisindeki metni büyük fontlu metin yaparak değerini geriye döndürür.
- **small()** = Değişken içerisindeki metni küçük fontlu metin yaparak değerini geriye döndürür.
- **bold()** = Değişken içerisindeki metni kalın fontlu metin yaparak değerini geriye döndürür.
- **italics()** = Değişken içerisindeki metni italik (sağa yatık) fontlu metin yaparak değerini geriye döndürür.
- **strike()** = Değişken içerisindeki metni üzeri çizgili metin yaparak değeri geriye döndürür.
- **fixed()** = Değişken içerisindeki metni teletype (daktilo) fontlu metin yaparak değeri geriye döndürür.
- **fontcolor()** = Kendisine parametre olarak verilen değer doğrultusunda değişken içeriğini renklendirerek değeri geriye döndürür.Varsayılan renk green(yeşil).
- **fontsize()** = Kendisine parametre olarak belirtilen değer doğrultusunda değişken içerisindeki metnin font boyutunu değiştirerek değeri geriye döndürür.
- **anchor()** = Değişkene çapa (<a>...) tanımlaması yapar.Aynı zamanda çapaya name değeri atar.
- **link()** = Değişkene çapa tanımlaması yapar.Aynı zamanada çapaya köprü değeri atar.

KOŞUL YAPILARI:

İF-ELSE İF-ELSE :

İf = Eğer , **Else if** = Değilse Eğer , **Else** = Değilse

Kullanılışı;

```
İf (koşul/koşullar){  
    İfade/ifadeler  
}else if (koşul/koşullar){  
    İfade/ifadeler  
}else{  
    İfade/ifadeler  
}
```

SWITCH-CASE-BREAK VE DEFAULT :

Switch = Koşul Başlatır , **Case** = Koşul Kontrolü , **Break** = Koşul Kontrolü Sonlandırması ,
Default = Koşul Olumsuzlukları Sonucu

Kullanılışı:

```
switch ( koşul ){  
    case koşul kontrolü:  
        .....  
    Break;  
    case koşul kontrolü:  
        .....  
    Break;  
    case koşul kontrolü:  
        .....  
    Break;  
    Default:  
        .....  
}
```

TERNARY OPERATÖRÜ :

? : = Ternary operatörür , koşul ? çalıştırılacak ifade 1 : çalıştırılacak ifade 2;

Kullanılışı:

```
Koşul ? çalıştırılacak ifade 1 : çalıştırılacak ifade 2;
```

TRY-CATCH-FINALLY-THROW :

try = Hata oluşturabilecek kodlar için kullanılır ,

catch = Hata oluşursa çalıştırılacak kodlar için kullanılır ,

finally = Hata oluşsada oluşmasada çalıştırılacak kodlar için kullanılır ,

throw = Hata oluşturabilecek kodlar için istisnai durumlarda kullanılır ,

Kullanılışı:

```
try {  
    Var isim = "sercan";  
    document.write ( isim + "<br/>" );  
    document.write ( soyisim );  
    throw ( err );  
}  
catch ( err ){  
    document.write ( "Hata oluştu <br/>" );  
    document.write ( err );  
    document.write ( "<br/>" );  
}  
finally{  
    document.write ( "Hata olsada olmasada çalıştı." );  
}
```

DÖNGÜLER:

WHİLE DÖNGÜSÜ :

while = Döngü uzunluğunda belirli bir koşul doğru olarak kod bloğu döndürür.

Kullanılışı:

```
While ( koşul ){  
    Çalışacak kodlar  
}
```

DO-WHİLE DÖNGÜSÜ :

do-while = Koşulun doğru olup olmadığını kontrol etmeden önce kod bloğunu bir kez çalıştıracak, ardından koşul doğru olduğu sürece döngüyü tekrarlayacaktır.

Kullanılışı:

```
do {  
    Çalışacak kodlar  
} while (koşul) ;
```

FOR DÖNGÜSÜ :

for = For döngüsü genellikle koşul ile belirtilen tekrar sayısı kadar kod bloğunun çalışması için kullanılır.

Kullanılışı:

```
for ( komut 1 ; koşul ; komut 2 ){  
    Çalışacak kodlar  
}
```

Not : Komut 1 döngü başlamadan çalıştırılacak komutları içerir.

Not : Koşul döngü kod bloğunun çalışma şartını belirtmek için kullanılır.

Not : Komut 2 döngü kod bloklarından sonra çalıştırılacak komutları içerir.

FOR-IN DÖNGÜSÜ :

for-in = For in döngüsü genellikle eleman sayısını bilmediğimiz diziler veya elemanlarını bilmediğimiz nesnelerde kullanılır.

Kullanılışı:

NOT : 1.parametre : Değişken

NOT : 2.parametre : Dizi / Nesne

```
for ( 1.parametre in 2. Parametre ){  
    Çalışacak kodlar  
}
```

DÖNGÜLERDE BREAK VE CONTINUE İFADELERİ:

BREAK :

break = Döngüden çıkmak için kullanılır.

Kullanılışı:

ÖRNEK:

```
Var baslangıç = 1;  
While ( baslangıç <=1000 ){  
    Document.write ( baslangıç + "<br/>" );  
    If ( baslangıç == 100){  
        Break ;  
    }  
    Baslangıç ++ ;  
}
```


CONTINUE :

continue = Döngünün o anki işlemini sonlandırarak döngünün başına döner ve işleme kaldığı yerden devam eder.

Kullanılışı:

ÖRNEK:

```
Var deger = 1 ;
While ( deger <=10 ){
    İf ( deger == 4 ){
        Deger ++ ;
        Continue ;
    }
    Document.write ( deger + "<br/>" );
    Deger ++ ;
}
```

FONKSİYONLAR VE KULLANIMLARI :

Javascript ile yazılmış bir fonksiyon, belirli bir işlemi yapmak için tasarlanmış kod bloğudur.

Fonksiyon ile yazılmış bir fonksiyon çalıştırıldığında (çağrıldığında) işlem yapar.

Kullanılışı:

```
function fonksiyonismi ( parametre / parametreler ) {
    çalıştırılacak kodlar
}
```

Fonksiyon isimlendirme kuralları :

- 1- Fonksiyon isimleri her zaman anlamlı olarak kullanılmalıdır.
- 2- Fonksiyon isimleri harfler ile başlayabilir.(Türkçe karakterler kullanılmaz.)
- 3- Fonksiyon isimleri _ (alt çizgi) işareti ile başlayabilir.
- 4- Fonksiyon isimleri \$ (dolar) işareti ile başlayabilir.

- 5- Fonksiyon isimleri hiçbir zaman rakam ile başlayamaz.(Fakat rakamlar ilk karakterden sonra kullanılabilir.)
- 6- Fonksiyon isimleri javascript tarafından rezerve kelimeleri içeremez.
- 7- Fonksiyon isimleri hiçbir zaman birden fazla kullanılamaz.
- 8- Fonksiyon isimleri küçük harf / büyük harf duyarlıdır.

Fonksiyon değer döndürme (return) :

Fonksiyonlar **return** anahtar kelimesine ile değer döndürür.

Kullanılışı:

```
<script>

    var sonuc;

    function Topla(sayi1, sayi2) {

        return sayi1 + sayi2;    // sayi1 ve sayi2 toplamını döndürür

    }

    sonuc = Topla(10, 5);

    alert(sonuc);

</script>
```

Fonksiyon çağırma :

Javascript ile yazılmış fonksiyon kodlarının çalışması için fonksiyon çağırma veya fonksiyon çalıştırma işleminin yapılması gerekir.

Fonksiyon çağırma/çalıştırma yolları =

- Bir olay meydana geldiğinde-Kullanıcı HTML nesnesine tıkladığında ;

```
<button onclick="alert('Merhaba JavaScript');">Merhaba</button>
```

- Fonksiyon çağırma işlemi yapıldığında ;

```
function uyari(mesaj) {

    alert(mesaj);

}

uyari("Merhaba JavaScript");
```

- Otomatik çalışan fonksiyon oluşturma ;

```
(function test () {  
    document.write ("Merhaba") ;  
})();
```

Fonksiyonlarda length özelliği :

Length : Fonksiyona tanımlanan parametre sayısı değerini geriye döndürür.

```
function ornek (isim,soyisim,yas,sehir,ülke){  
    var islemyap = ornek.length ;  
    document.write ( islemyap );      // Çıktı olarak 5 sayısı ekrana yazdırır.  
}  
Ornek () ;
```

Fonksiyonlarda arguments özelliği :

arguments : Fonksiyona herhangi bir parametre tanımlanmadan bile gelebilecek tüm parametre değerlerinin alınmasını sağlar.

```
function ornek (){  
    var sonuc = arguments.length ;  
    document.write ( sonuc );      // Çıktı olarak 5 sayısı ekrana yazdırır.  
}  
Ornek ("sercan" , "Lülleci" ,28,"Zonguldak" , "Türkiye") ;
```

Fonksiyonlarda caller özelliği :

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

```
function esas ( ){
    for (var baslangic = 0 ; baslangic < esas . caller .arguments . length ; baslangic ++ ){
        document.write ( diger . arguments [baslangic] ) ;
    }
}
Function diger ( isim , soyisim ){
    Esas ( ) ;           //çıktı olarak SercanLülleci ekrana yazdırır.
}
Diger ( “Sercan” ,”Lülleci” ) ;
```

DİZİLER VE KULLANIMLARI :

Dizi : Bir değişken içerisinde toplanmış her türden bilgi ve veriyi depolamak için kullanılan sıralı veri gruplarına verilen isimdir.

İndexler / indisler her zaman 0 ile başlar ve her elemanda 1 artarak devam eder.

Dizi elemanları [] arasında yazılır.

Kullanılışı:

```
<script>
    var isimlrs = [ “Volkan”,”Hakan”, “Onur”, ”Serkan”, “Ümit”]
    document.write (isimler + “<br />“<br />“<br />“<br />”);
    document.write (isimler[0] + “<br />”);
    document.write (isimler[1] + “<br />”);
    document.write (isimler[2] + “<br />”);
    document.write (isimler[3] + “<br />”);
    document.write (isimler[4] );
</script>
```

- Dizi içerisinde elemanlar oluştururken “ (çift tırnak) ile başlıyorsa “ (çift tırnak) ile bitmelidir.
- Dizi içerisinde elemanlar oluşturulurken ‘ (tek tırnak) ile başlıyorsa ‘ (tek tırnak) ile bitmelidir.

- Eleman içerisinde aynı tırnak işareti kullanılacak ise muhakkak \ (ters slash) karakteri ile kullanılmalıdır.
- Rakamsal bir değer (number) verilecek ise tırnak kullanılmamalıdır.

Not: Bir başka dizi oluşturma şeklide new Array ifadesidir.

Örnek;

```
var günler=new  
Array("Pazartesi","Salı","Çarşamba","Perşembe","Cuma","Cumartesi","Pazar");  
  
var günler =[ Pazartesi","Salı","Çarşamba","Perşembe","Cuma","Cumartesi","Pazar"];
```

İki örnekte birer dizi belirtmektedir.

isArray () : Bir değişkenin içeriğinin dizi olup olmadığını kontrol eder. Boolean türünde sonuç döndürür. (true-false).

length : Dizi içeriğinin eleman sayısını veya dizi elemanının karakter sayısını geriye döndürür.

Kullanılışı:

```
<script>  
  
var renkler = [ "Siyah","Sarı", "Beyaz", "Kırmızı", "Mavi"];  
  
document.write(renkler + "<br/>");  
  
var islemyap = renkler.length;           //Çıktı olarak 5 sayısını verir.  
  
document.write("Dizinin eleman sayısı: " + islemyap);  
  
</script>
```

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

```
var renkler = [ "Siyah","Sarı", "Beyaz", "Kırmızı", "Mavi"];  
  
delete renkler[2];  
  
//çıktıda 2 indis numarsında yer alan Beyaz yerine underlined yazar.
```

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

```
var renkler = [ "Siyah","Sarı", "Beyaz", "Kırmızı", "Mavi"];  
  
var islemyap = renkler.pop();  
  
//Çıktı olarak islemyap içinde mavi rengi depolanır ve renkler dizisinden mavi çıkartılır.
```

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

```
var renkler = [ "Siyah", "Sarı", "Beyaz", "Kırmızı", "Mavi" ];  
  
var islemyap = renkler.shift();  
  
//Çıktı olarak islemyap içinde siyah rengi depolanır ve renkler dizisinden siyah çıkartılır.
```

push () : Dizi sonuna eleman veya elemanlar eklenmesini sağlar.

```
var renkler = [ "Siyah", "Sarı", "Beyaz", "Kırmızı", "Mavi" ];  
  
var islemyap = renkler.push("Pembe");  
  
//çıktı olarak renkler dizisinin sonuna pembe rengini ekler.
```

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

```
var renkler = [ "Siyah", "Sarı", "Beyaz", "Kırmızı", "Mavi" ];  
  
var islemyap = renkler.unshift("Pembe");  
  
//çıktı olarak renkler dizisinin başına pembe rengini ekler.
```

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

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

```
var sayılar = [ "bir", "iki", "üç", "dört", "beş" ];  
  
var yenisayılar = sayılar.slice(2);  
  
/yenisayılar dizisi sayılar dizisinin 2. Indexinden sonraki elemanlardan oluşur yani;  
  
Yenisayılar = üç-dört-beş elemanlarını içerir.
```

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

1.parametre : Hedef

2.parametre : Başlangıç

3.parametre : Bitiş

```
var sayılar = [ "bir", "iki", "üç", "dört", "beş", "altı", "yedi", "sekiz" ];  
  
sayılar.copyWithin(2);  
  
document.write(sayılar);
```

```
//çıktı : bir,iki,bir,iki,üç,dört,beş,altı
```

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

```
var meyveler = [ "elma","armut", "kiraz", "kayısı", "karpuz","kavun","muz","ananas"];  
meyveler.fill("nar",4,6);  
document.write(meyveler);  
  
//çıktı: elma,armut, kiraz, kayısı, nar, nar, muz, ananas
```

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.

```
var erkek = [ "volkan","hakan", "sercan"];  
var bayan = [ "aslı","banu", "burçin"];  
var isimler = erkek.concat(bayan);  
document.write(isimler);  
  
//çıktı : volkan,hakan,Sercan,aslı,banu,burçin
```

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.

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

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

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

```
var isimler = [ "volkan","hakan", "sercan"];  
var islemyap = isimler.indexOf("Sercan");  
document.write(islemyap)  
  
//çıktı : 2
```

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

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

- 1.parametre : Her işlemde elde edilen dizi elemanı.
- 2.parametre : Her işlemde elde edilen dizi elemanının indis/index değeri.
- 3.parametre : Her tekrar işleminde diziyi tutan değişkenin referansı.

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.

- 1.parametre : Her işlemde elde edilen dizi elemanı.
- 2.parametre : Her işlemde elde edilen dizi elemanının indis/index değeri.
- 3.parametre : Her tekrarda diziyi tutan değişkenin referansı.

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

- 1.parametre : Dizi içerisinde mevcut değer öncesindeki değerlerin toplamı / birleşimi.
- 2.parametre : Her işlemde elde edilen dizi elemanı.
- 3.parametre : Her işlemde elde edilen dizi elemanının indis/index değeri.
- 4.parametre : Her tekrarda diziyi tutan değişkenin referansı.

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

- 1.parametre : Dizi içerisinde mevcut değer öncesindeki değerlerin toplamı / birleşimi.
- 2.parametre : Her işlemde elde edilen dizi elemanı.
- 3.parametre : Her işlemde elde edilen dizi elemanının indis/index değeri.
- 4.parametre : Her tekrarda diziyi tutan değişkenin referansı.

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

```
var meyveler=["elma","armut","karpuz","kavun","muz","ananas"]  
document.write(meyveler + "<br/>"); //çikti: elma,armut,karpuz,kavun,muz,ananas  
var yenimeyveler = meyveler.reverse();  
document.write(yenimeyveler); //çikti:ananas,muz,kavun,karpuz,armut,elma
```


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

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

- 1.parametre : Her işlemde elde edilen dizi elemanı.
- 2.parametre : Her işlemde elde edilen dizi elemanının indis/index değeri.
- 3.parametre : Tekrarlama işlemi için dizinin atandığı değişkenin referansı.

some () : Kendisine parametre olarak belirtilen fonksiyon true (doğru / geçerli / olumlu / başarılı) değeri döndürene kadar dizi elemanları üzerinde sıra ile gezinilmesini sağlar.

- 1.parametre : Her işlemde elde edilen dizi elemanı.
- 2.parametre : Her işlemde elde edilen dizi elemanının indis/index değeri.
- 3.parametre : Her tekrar işleminde diziyi tutan değişkenin referansı.

every () : Kendisine parametre olarak verilen fonksiyon false (yanlış / geçersiz / olumsuz / başarısız) değeri döndürene kadar dizi elemanları üzerinde sıra ile gezinilmesini sağlar.

- 1.parametre : Her işlemde elde edilen dizi elemanı.
- 2.parametre : Her işlemde elde edilen dizi elemanının indis/index değeri.
- 3.parametre : Her tekrar işleminde diziyi tutan değişkenin referansı.

filter () : Kendisine parametre olarak verilen fonksiyonu dizideki her eleman üzerine çalıştırır.Fonksiyon geriye true değerini döndüren tüm elemanları kopyalayarak yeni bir dizi oluşturur.Oluşturduğu dizinin değerini geriye döndürür.

- 1.parametre : Her işlemde elde edilen dizi elemanı.
- 2.parametre : Her işlemde elde edilen dizi elemanının indis/index değeri.
- 3.parametre : Tekrarlama işlemindeki diziyi tutan değişkenin referansı.

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.

- 1.parametre : Her işlemde elde edilen dizi elemanı.
- 2.parametre : Her işlemde elde edilen dizi elemanının indis/index değeri.
- 3.parametre : Her tekrarda diziyi tutan değişkenin referansı.

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

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

JAVASCRIPTTE MATEMATİKSEL İŞLEMLER :

MAX_VALUE : Atanabilecek en büyük pozitif sayı değerini geriye döndürür.(1.7976931348623157e+308)

MIN_VALUE : Atanabilecek en küçük negatif sayı değerini geriye döndürür.(5e-324)

POSITIVE_INFINITY : Atanabilecek en büyük pozitif sayı değerinden büyük sayılarda sonsuzluk değerini geriye döndürür.

NEGATIVE_INFINITY : Atanabilecek en küçük negatif sayı değerinden küçük sayılarda negatif sonsuzluk değerini geriye döndürür.

NAN : Değerin sayı olmaması durumunda geriye döner.Not a Number(Bir sayı değil).

E : Euler sayı değerini geriye döndürür. (e sayısı = 2.718281828459045)

```
var deger = Math.E;
```

LN2 : 2 sayısının doğal logaritması değerini geriye döndürür. (0.6931471805599453)

```
var deger = Math.LN2;
```

LN10 : 10 sayısının doğal logaritması değerini geriye döndürür. (2.302585092994046)

```
var deger = Math.LN10;
```

LOG2E : Euler sayısının 2 tabanına göre logaritmik değerini geriye döndürür.
(1.4426950408889634)

```
var deger = Math.LOG2E;
```

LOG10E : Euler sayısının 10 tabanına göre logaritmik değerini geriye döndürür.
(0.4342944819032518)

```
var deger = Math.LOG10E;
```

PI : Pi sayısı değerini geriye döndürür. (pi sayısı = 3.141592653589793)

```
var deger = Math.PI;
```

SQRT1_2 : ½ sayısının karekök değerini geriye döndürür.(0.7071067811865476)

```
var deger = Math.SQRT1_2;
```

SQRT2 : 2 sayısının karekök değerini geriye döndürür.(1.4142135623730951)

```
var deger = Math.SQRT2;
```

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

```
var degerbir = "Sercan";  
var degeriki = 8;  
var degeruc = 5.5;  
var sonucbir = Number.isNaN(degerbir);  
var sonucbir = Number.isNaN(degeriki);  
var sonucbir = Number.isNaN(degeruc);  
document.write(sonucbir);    //çıktı : false  
document.write(sonuciki);    //çıktı : false  
document.write(sonucuc);    //çıktı : false
```

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.

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

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

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

```
Var deger = 4.324231213456;  
Document.write(deger + "<br/>");  
var sonuc = deger.toFixed(2);  
document.write(sonuc);    //çıktı: 4.32
```

toPrecision() : Kendisine parametrik olarak verilen değer doğrultusunda değişken içeriğindeki sayının kaç basamak olacağını belirleyerek yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür.(Nokta hariç)

```
Var degerbir = 12.6645;  
Var degeriki = 55;  
Var sonucbir = degerbir.toPrecision(4);
```

```
Var sonuciki = degeriki.toPrecision(4);  
  
Document.write(sonucbir + "<br/>");           //çıktı :12.66  
  
Document.write(sonuciki + "<br/>");           //çıktı : 55.00
```

toExponential() : Kendisine parametrik olarak verilen değer doğrultusunda değişken içeriğindeki sayının ondalık hanesinde kaç basamak olacağını belirleyerek,kalan değeri üstsek hale getirerek yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür.

parseInt() : Kendisine parametrik olarak verilen değerler doğrultusunda değişken içeriğini tam sayı haline dönüştürerek,yeni bir içerik oluşturur ve oluşturduğu değeri geriye döndürür.(rakam,artı değeri veya eksi değeri haricinde bir karakter gördüğünde işlem durur.)

parseFloat() : Kendisine parametrik olarak verilen değer doğrultusunda değişken içeriğini ondalıklı sayı haline dönüştürerek,yeni bir içerik oluşturur ve oluşturduğu içeriği geriye döndürür.(Rakam,eksi değeri,artı değeri veya haricinde bir karakter gördüğünde işlem durur.)

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

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

Number() : Değişken içeriğini mutlak olarak number (sayı) veri türüne çevirerek oluşturduğu değeri geriye döndürür.

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

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

```
Var degerbir = 3.12;  
  
Var degeriki = 54.0000000000001;  
  
Var sonucbir = Math.ceil(degerbir);  
  
Var sonuciki = Math.ceil(degeriki);  
  
Document.write(sonucbir);           //çıktı: 4  
  
Document.write(sonuciki);           //çıktı: 55
```

ceil() : Kendisine parametre olarak verilen ondalıklı sayı değerini aşağı yuvarlayarak değeri geriye döndürür.

```
Var degerbir = 12.99;  
  
Var degeriki = 77.00000;
```

```
Var sonucbir = Math.floor(degerbir);  
  
Var sonuciki = Math.floor(degeriki);  
  
Document.write(sonucbir);           //çıktı: 12  
  
Document.write(sonuciki);           //çıktı: 77
```

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.

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

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

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

min() : Kendisine parametre olarak verilen sayı değerinin en küçüğünü bularak, bulunduğu değeri geriye döndürür.

```
var degerbir = 5;  
  
var degeriki = 8;  
  
var degeruc = 12;  
  
var degerdort = 2;  
  
var sonuc = Math.min(degerbir , degeriki, degeruc, degerdort);  
  
document.write(sonuc);           //çıktı:2
```

max() : Kendisine parametre olarak verilen sayı değerinin en büyüğünü bularak, bulunduğu değeri geriye döndürür.

```
var degerbir = 5;  
  
var degeriki = 8;  
  
var degeruc = 12;  
  
var degerdort = 2;  
  
var sonuc = Math.max(degerbir , degeriki, degeruc, degerdort);  
  
document.write(sonuc);           //çıktı:12
```

pow() : Kendisine parametre olarak verilen iki sayının üssünü bularak, bulunduğu değeri geriye döndürür.

```
var degerbir = 2;  
  
var degeriki = 5;
```

```
var sonucbir = Math.pow(degerbir,degeriki); //2*2*2*2*2=32
var sonuciki = Math.pow(2,5); //2*2*2*2*2=32
document.write(sonucbir + "<br/>"); //çıktı:32
document.write(sonuciki); //çıktı:32
```

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

sqrt() : Kendisine parametre olarak verilen herhangi bir sayı değerinin karekökünü bularak, bulunduğu değeri geriye döndürür.

```
var deger = 9;
var sonuc = Math.sqrt(deger);
document.write(sonuc); //çıktı:3
```

cbrt() : Kendisine parametre olarak verilen herhangi bir sayının küp kökünü bularak,bulduğu değeri geriye döndürür.

```
var deger = 55;
var sonuc = Math.cbrt(deger);
document.write(sonuc); //çıktı:3,8023524607613916
```

sin() : Kendisine parametre olarak verilen herhangi bir sayının sinüsünü bularak,bulduğu değeri geriye döndürür.

sinh() : Kendisine parametre olarak verilen herhangi bir sayının hiperbolik sinüsünü bularak,bulduğu değeri geriye döndürür.

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

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

cos() : Kendisine parametre olarak verilen herhangi bir sayının kosinüsünü bularak,bulduğu değeri geriye döndürür.

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

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

acosh() : Kendisine parametre olarak verilen herhangi bir sayı değerin hiperbolik ark kosinüsünü bularak,bulduğu değeri geriye döndürür.

tan() : Kendisine parametre olarak verilen herhangi bir sayının tanjantını bularak,bulduğu değeri geriye döndürür.

tanh() : Kendisine parametre olarak verilen herhangi bir sayının hiperbolik tanjantını bularak,bulduğu değeri geriye döndürür.

atan() : Kendisine parametre olarak verilen herhangi bir sayı değerin ark tanjantını radyan cinsinden bularak,bulduğu değeri geriye döndürür.

atanh() : Kendisine parametre olarak verilen herhangi bir sayı değerin / -1 ile 1 arasındaki sayı değerin hiperbolik ark tanjantını bularak,bulduğu değeri geriye döndürür.

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

exp() : Kendisine parametre olarak verilen herhangi bir sayı değerin euler sayısı üssünü bularak,bulduğu değeri geriye döndürür.

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

log() : Kendisine parametre olarak verilen herhangi bir sayının doğal logaritmasını bularak,bulduğu değeri geriye döndürür.(Baz: Euler sayısı)

JAVASCRIPTTE DÜZENLİ İFADELER :

Regular expression (düzenli ifade), herhangi bir karakter diziliminin yapısını yani içerdiği karakterlerin neler olduğu ile bunların diziliş sırası ve biçimlerinin önceden tanımlanan belli bir biçime uygunluğunun sınanması sırasında kullanılan standart gösterim biçimleri ya da denetim şablonlarıdır.

Regular expression (düzenli ifade), kısaca RegExp olarak isimlendirilmektedir. Tüm gelişmiş yazılım dillerinde sınaama işlemleri için yaygın olarak kullanılmaktadır. Regular expression'ların (düzenli ifadelerin) kullanılması ile sağlanan en önemli kolaylık, karakter dizilimlerinde bulunan karakterlerin önceden bilinen belli bir desene uygun olup olmadığını sınaama yeteneğinden gelmektedir. Bu sayede karakter dizilimi içerisinde belli bir karakter, kelime veya daha karmaşık olan belli bir dizilime sahip bir bilgiyi aramaya yardımcı olur ve belli bir kural dahilinde uygun verileri filtreleme sağlanmış olur. Desenler, bir karakter dizilimi içindeki karakterler için sınaama kriterlerini tanımlarlar. Desenler, düz metinler şeklinde çok basit olabileceği gibi bazı özel karakterler kullanılarak oldukça karmaşık halde de oluşturulabilirler. Herhangi bir desen, araştırılması istenen metnin / (slash) karakterleri arasına yerleştirilmesiyle tanımlanır.

Regular expression'ler (düzenli ifadeler), genellikle bir metni düzenlemek, bir metin içerisinde belli kurallara uyan metinleri elde etmek, kullanıcılardan bilgi talep ederken istenilen bilgilerin doğru ve tam olarak girilmesini sağlamak, karakterleri belirlemek, yasaklamak veya sınırlamak vs. gibi işlemlerde kullanılır. Regular expression'lerde (düzenli ifadelerde) kullanılan birtakım özellikler ve karakterler bulunmaktadır. Bu özellik ve karakterlerden bahsedecek olur isek;

Ayarlar	
ifade	Açıklama
i	Karakter diziliminin karşılaştırılmasında büyük harf / küçük harf ayrımının yapıp yapılmayacağını tanımlamasında kullanılır.
g	Karakter diziliminin tamamının karşılaştırmada kontrol edilip edilmeyeceğini tanımlamada kullanılır.
m	Karakter diziliminin birden fazla satırı varsa tüm satırlarının karşılaştırmada kontrol edilip edilmeyeceğini tanımlamada kullanılır.

Konum belirleyiciler	
ifade	Açıklama
^	Karakter diziliminin başlangıcı ile düzenli ifade arasında bir eşleşme aramak için kullanılır.
\$	Karakter diziliminin sonu ile düzenli ifade arasında bir eşleşme aramak için kullanılır.
\b	Düzenli ifadeye uygun olabilecek metin parçalarını, karakter diziliminin içerisindeki kelimelerin başında yada sonunda aramak için kullanılır.
\B	Düzenli ifadeye uygun olabilecek metin parçalarını, karakter diziliminin içerisindeki kelimelerin içerisinde aramak için kullanılır.
?=	Karakter dizilimi içerisindeki belirtilen bir referans değer önünde düzenli ifade ile eşleşebilecek değerleri aramak için kullanılır.
?!	Karakter dizilimi içerisindeki belirtilen bir referans değer ile takip edilmeyen değerleri düzenli ifade ile eşleşebilecek değerleri aramak için kullanılır.

Nicelik belirleyiciler	
ifade	Açıklama
{x}	Karakter diziliminde bulunan değer x defa tekrarlanmış olması gerektiğini belirtmek için kullanılır.
{x,}	Karakter diziliminde bulunan değer x defa yada daha fazla tekrarlanmış olması gerektiğini belirtmek için kullanılır.
{x,y}	Karakter diziliminde bulunan değer en az x defa yada en fazla y defa tekrarlanmış olması gerektiğini belirtmek için kullanılır.
+	Karakter diziliminde bulunan değer 1 defa yada daha fazla tekrarlanmış olması gerektiğini belirtmek için kullanılır.
?	Karakter diziliminde bulunan değer 0 defa yada 1 defa tekrarlanmış olması gerektiğini belirtmek için kullanılır.
*	Karakter diziliminde bulunan değer 0 defa yada daha fazla tekrarlanmış olması gerektiğini belirtmek için kullanılır.

Özel belirleyiciler	
ifade	Açıklama
()	Düzenli ifadelerde grup tanımlamak için kullanılır.
\	Düzenli ifadelerde ardından gelecek olan karakterin özel bir karakter olup olmadığını belirtmek için kullanılır.
(?:)	Düzenli ifadeler içerisinde alt grup oluşturmak için kullanılır.
[abc]	Karakter diziliminde bulunan değer, belirtilen karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.
[a-z]	Karakter diziliminde bulunan değer, belirtilen karakterler aralığından herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.
[^abc]	Karakter diziliminde bulunan değer, belirtilen karakterlerin dışındaki karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.
[^a-z]	Karakter diziliminde bulunan değer, belirtilen karakterler aralığı dışındaki karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.
(x y)	Karakter diziliminde bulunan değer, belirtilen x yada z karakterlerinden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.
.	Karakter diziliminde bulunan değer, yeni satır karakterleri hariç herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.
\w	Karakter diziliminde bulunan değer, harf ve rakamlardan veya _ (alt çizgi) karakterlerinden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.
\W	Karakter diziliminde bulunan değer, harf ve rakamlardan veya _ (alt çizgi) karakterler dışındaki karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.
\d	Karakter diziliminde bulunan değer, sadece rakam karakterlerinden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.
\D	Karakter diziliminde bulunan değer, rakam karakterleri dışındaki karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.
\s	Karakter diziliminde bulunan değer, sadece boşluk karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.
\S	Karakter diziliminde bulunan değer, boşluk karakteri dışındaki karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.
\0	Karakter diziliminde bulunan değer, sadece Null karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.
\n	Karakter diziliminde bulunan değer, sadece yeni satır karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.
\f	Karakter diziliminde bulunan değer, sadece form besleme karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.
\r	Karakter diziliminde bulunan değer, sadece paragraf sonu karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.
\t	Karakter diziliminde bulunan değer, sadece yatay sekme karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.
\v	Karakter diziliminde bulunan değer, sadece dikey sekme karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.
\000	Karakter diziliminde bulunan değer, octal numeral system (sekizlik sayı sistemi) tabanında ASCII (American Standard Code for Information Interchange) (bilgi değişimi için amerikan standart kodlama sistemi) karşılığı olan karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.
\x00	Karakter diziliminde bulunan değer, hexadecimal numeral system (onaltılık sayı sistemi) tabanında ASCII (American Standard Code for Information Interchange) (bilgi değişimi için amerikan standart kodlama sistemi) karşılığı olan karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.
\u0000	Karakter diziliminde bulunan değer, 16 bit'lik Unicode (Evrensel Kod) formatındaki karşılığı olan karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.

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.

g : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriğinin tamamının kontrol edilmesini sağlar.

```
var icerik = "Merhaba Sercan nasılsın ? Senin adın Sercan mıydı? Sercan";  
var ifade = new RegExp ("Sercan","g");  
var sonuc = icerik.match (ifade);  
document.write(sonuc);
```

//çıktı : Sercan,Sercan,Sercan

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

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

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

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

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

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

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

source : Düzenli ifade içerisinde kullanılan desen değerini geriye döndürür.

search() : Kendisine parametre olarak verilen düzenli ifadeli değer doğrultusunda değişken içeriğini kontrol eder ve ilk eşleşmeyi sağlayan değişken

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

```
var metin = "Dünyada ve Türkiye’de en yaygın olarak kullanılan script dili javascript’tir.”;  
var ifade = new RegExp ("script","g");  
ifade.text(metin);
```

```
var sonuc = ifade.lastIndexOf ;  
  
document.write(sonuc); //çıktı: 56
```

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

```
var metin = "Dünyada ve Türkiye’de en yaygın olarak kullanılan script dili javascript’tir.”;  
var ifade = new RegExp ("script","g");  
ifade.test(metin);  
  
var sonuc = ifade.lastMatch ;  
  
document.write(sonuc); //çıktı: script
```

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

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

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

```
var metin = "Merhaba Sercan,merhaba Ömer,merhaba Onur";  
  
document.write(metin + "<br/>"); //çıktı:Merhaba Sercan,merhaba Ömer,merhaba Onur  
  
var ifade = new RegExp("Ömer");  
  
var sonuc = metin.replace(ifade , "Mehmet");  
  
document.write(sonuc); //çıktı: Sercan,merhaba Mehmet,merhaba Onur
```

split() : Kendisine verilecek olan düzenli ifadeli değer doğrultusunda değişken içeriğini kontrol eder ve aranan değer eşleşmesi durumunda eşleşmeyi sağlayan değişken içeriğini bölümlererek yeni bir dizi oluşturur ve oluşturduğu diziyi geriye döndürür.

exec() : Kendisine verilecek olarak verilen değişken içeriğinde düzenli ifade ile aranan değer eşleşmesi durumunda eşleşmeyi sağlayan değişken değerinden yeni bir dizi oluşturarak,oluşturduğu dizi değerini geriye döndürür.

g ayarı kullanılacak olur ise exec() metodu her çağrıldığında lastIndex değeri otomatik eşleşen bir sonraki indis / index değerini alacaktır.

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

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

^ : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriğini karakter diziliminin başlangıcı ile düzenli ifade arasında eşleşme aramak için kullanılır.

\$: Düzenli ifade dahilinde kontrol edilecek olan değişken içeriğini karakter diziliminin sonu ile düzenli ifade arasında eşleşme aramak için kullanılır.

\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.

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

? : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter dizilimi içerisindeki belirtilen bir referans değerin önünde düzenli ifade ile eşleşebilecek değerleri aramak için kullanılır.

?! : 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.

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

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

{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 ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin 1 defa yada daha fazla tekrarlanmış olması gerektiğini belirtmek için kullanılır.

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

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

() : Düzenli ifadelerde grup tanımlamak için kullanılır.

\ : Düzenli ifadelerde ardından gelecek olan karakterin özel bir karakter olup olmadığını belirtmek için kullanılır.

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

[abc] : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin, belirtilen karakterlerden herhangi birisi eşleşmesi gerektiğini belirtmek için kullanılır.

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

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

[^a-z] : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin, belirtilen karakterler dışındaki karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.

(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.

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

```
var metin = "Javascript rapor";  
var ifade = new RegExp("r.p","g");  
var sonuc = metin.match(ifade);  
document.write(sonuc); //çikti:rip,rap
```

\w : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin, harf,rakam veya _ (alt çizgi) karakterinden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.

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

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

\D : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin, rakam karakterleri dışındaki herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.

\s : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin,sadece boşluk karakterleri ile eşleşmesi gerektiğini belirtmek için kullanılır.

\S : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin,sadece boşluk karakterleri dışındaki karakterlerden herhangi birisi ile eşleşmesi gerektiğini belirtmek için kullanılır.

\0 : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin,sadece Null (boş) karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.16 bit Null değeri :\u0000

\f : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin,sadece form besleme karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. \u000C : 16 bit yazım şekli

\n : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin,sadece yeni satır karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. \u000A : 16 bitlik yazım şekli

\r : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin,sadece paragraf sonu karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. \u000D : 16 bit yazım şekli

\t : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin,sadece yatay sekme (tab) karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. \u0009 : 16 bit yazım şekli

\v : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin,sadece dikey sekme karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. \u000B : 16 bit yazım şekli

\XXX : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin,octal numeral system tabanında ASCII karşılığı olan karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.

Değer aralığı : \000-\377

Octal numeral system : sekizlik sayı sistemi

ASCII : American Standart Code for Information Interchange:Bilgi değişimi için Amerikan standart kodlama sistemi

166 157 154 153 141 156

\xxx : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin, hexadecimal numeral system tabanında ASCII karşılığı olan karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır.

Değer aralığı : \x00-\xFF

Hexadecimal numeral system : ondalıklı sayı sistemi

ASCII : American Standart Code for Information Interchange: Bilgi değişimi için Amerikan standart kodlama sistemi

\x76 \x6F \x6C \x61 \x6E

\uXXXX : Düzenli ifade dahilinde kontrol edilecek olan değişken içeriği karakter diziliminde bulunan değerin, 16 Bit'lik Unicode formatındaki karşılığı olan karakteri ile eşleşmesi gerektiğini belirtmek için kullanılır. 16 bit'lik Unicode karakter aralığı: \u0000-\uFFFF

JAVASCRIPTTE TARİH VE SAAT İŞLEMLERİ :

Date() : Kullanıldığı değişken içerisinde tarih ve saat bilgilerini, depoladığı değeri geriye döndürür. (Tarihe etki edilebilir.)

```
var zaman = new Date ();  
  
document.write(zaman);
```

getFullYear() : Kullanıldığı değişken içerisinde yıl bilgilerini 1000 ile 9999 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür.

```
var zaman = new Date ();  
  
var yil = zaman.getFullYear ();  
  
document.write(yil); //çıktı:2021
```

getMonth() : Kullanıldığı değişken içeriğine yılın ay bilgisini 0 ile 11 arasında olmak üzere rakamsal olarak depolayarak, depoladığı değeri geriye döndürür. (İlk ay Ocak ayıdır ve değeri 0'dır.)

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

```
var zaman = new Date ();
```

```
var gün =zaman.getDate ();  
  
document.write(gün);
```

getDay() : Kullanıldığı değişken içeriğine haftanın gün bilgisini 0 ile 6 arasında olmak üzere rakamsal depolayarak,depoladığı değeri geriye döndürür.(Haftanın ilk günü Pazar olarak değerlendirilir ve değeri 0'dır.)

```
var zaman = new Date ();  
  
var gunumuz = zaman.getDay();  
  
document.write(gunumuz);
```

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

```
var zaman = new Date ();  
  
var saat = zaman.getHours();  
  
document.write(saat);
```

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

```
var zaman = new Date ();  
  
var dakika = zaman.getMinutes();  
  
document.write(dakika);
```

getSeconds() : Kullanıldığı değişken içeriğine dakikanın saniyesini 0 ile 59 arasında olmak üzere rakamsal olarak depolayarak,depoladığı değeri geriye döndürür.

```
var zaman = new Date ();  
  
var saniye = zaman.getSeconds();  
  
document.write(saniye);
```

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

```
var zaman = new Date ();  
  
var milisaniye = zaman. getMilliseconds ();
```

```
document.write(milisaniye);
```

getUTCFullYear() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait yıl bilgisini 1000 ile 9999 arasında olmak üzere rakamsal olarak depolayarak,depoladığı değeri geriye döndürür.UTC :Eş güdümlü evrensel zaman...

```
var zaman = new Date ();  
  
var yilimiz = zaman. getUTCFullYear ();  
  
document.write(yilimiz);
```

getUTCMonth() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait yılın ay bilgisini 0 ile 11 arasında olmak üzere rakamsal olarak depolayarak,depoladığı değeri geriye döndürür.(ilk ay Ocak ayıdır ve değeri 0'dır.) UTC :Eş güdümlü evrensel zaman...

```
var zaman = new Date ();  
  
var ayimiz = zaman. getUTCMonth ();  
  
document.write(ayimiz);
```

getUTCDate() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait ayın gün bilgisini 1 ile 31 arasında olmak üzere rakamsal olarak depolayarak,depoladığı değeri geriye döndürür. UTC :Eş güdümlü evrensel zaman...

```
var zaman = new Date ();  
  
var gun = zaman. getUTCDate ();  
  
document.write(gun);
```

getUTCDay() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait haftanın gün bilgisini 0 ile 6 arasında olmak üzere rakamsal olarak depolayarak,depoladığı değeri geriye döndürür.(Haftanın ilk günü Pazar'dır ve değeri 0'dır.).UTC :Eş güdümlü evrensel zaman...

```
var zaman = new Date ();  
  
var hgunu = zaman. getUTCDay ();  
  
document.write(hgunu);
```

getUTCHours() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait günün saat bilgisini 0 ile 23 arasında olmak üzere rakamsal olarak depolayarak,depoladığı değeri geriye döndürür.UTC :Eş güdümlü evrensel zaman...


```
var zaman = new Date ();

var saat = zaman. getUTCHours ();

document.write(saat);
```

getUTCMinutes() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait saatin dakika bilgisini 0 ile 59 arasında olmak üzere rakamsal olarak depolayarak,depoladığı değeri geriye döndürür.UTC :Eş güdümlü evrensel zaman...

```
var zaman = new Date ();

var dakika = zaman. getUTCMinutes ();

document.write(dakika);
```

getUTCSeconds() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait saniye bilgisini 0 ile 59 arasında olmak üzere rakamsal olarak depolayarak,depoladığı değeri geriye döndürür.UTC :Eş güdümlü evrensel zaman...

```
var zaman = new Date ();

var saniye = zaman. getUTCSeconds ();

document.write(saniye);
```

getUTCMilliseconds() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait saniyenin milisaniye bilgisini 0 ile 999 arasında olmak üzere rakamsal olarak depolayarak,depoladığı değeri geriye döndürür.UTC :Eş güdümlü evrensel zaman...

```
var zaman = new Date ();

var milisaniye = zaman. getUTCMilliseconds ();

document.write(milisaniye);
```

setFullYear() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine yıl bilgisini 1000 ile 9999 arasında olmak üzere rakamsal olarak ayarlayarak depolayarak ve depoladığı değeri geriye döndürür.Çoklu ayarlama işlemi yapılabilir.

```
var zaman = new Date ();

zaman. setFullYear (2050);

document.write(zaman);
```

setMonth() : Kendisine parametre olarak belirtilmiş değer doğrultusunda değişken içeriğine yılın ay bilgisini 0 ile 11 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür.(İlk ay Ocak'tır ve değeri 0'dır.)Çoklu ayarlama işlemi yapılabilir.(Sıralama: Ay,Gün)

```
var zaman = new Date ();  
  
zaman. setMonth (5);  
  
document.write(zaman);
```

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

```
var zaman = new Date ();  
  
zaman. setDate (12);  
  
document.write(zaman);
```

setHours() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine günün saat bilgisini 0 ile 23 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür.Çoklu ayarlama işlemi yapılabilir.(Sıralama:saat,dakika,saniye,milisaniye)...

```
var zaman = new Date ();  
  
zaman. setHours (18);  
  
document.write(zaman);
```

setMinutes() : Kendisine parametre olarak belirtilmiş değer doğrultusunda değişken içeriğine saatin dakika bilgisini 0 ile 59 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür.Çoklu ayarlama işlemi yapılabilir.(Sıralama:Dakika,Saniye,Milisaniye)...

```
var zaman = new Date ();  
  
zaman. setMinutes (40);  
  
document.write(zaman);
```

setSeconds() : Kendisine parametre olarak belirtilmiş değer doğrultusunda değişken içeriğine dakikanın saniye bilgisini 0 ile 59 arasında olmak üzere rakamsal olarak ayarlayarak depolar

ve depoladığı değeri geriye döndürür. Çoklu ayarlama işlemi yapılabilir.(Sıralama:Saniye,Milisaniye)...

```
var zaman = new Date ();  
  
zaman. setSeconds (30);  
  
document.write(zaman);
```

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

```
var zaman = new Date ();  
  
zaman. setMilliseconds (600);  
  
document.write(zaman);
```

setUTCFullYear() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait yıl bilgisini 1000 ile 9999 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür.Çoklu ayarlama işlemleri yapılabilir.(Sıralama:Yıl,Ay,Gün).UTC:Eş güdümlü evrensel zaman...

```
var zaman = new Date ();  
  
zaman. setUTCFullYear (3050);  
  
document.write(zaman);
```

setUTCMonth() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait yılın ay bilgisini 0 ile 11 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür.(İlk ay Ocak'tır ve değeri 0'dır.)Çoklu ayarlama işlemleri yapılabilir.(Sıralama:Ay,Gün).UTC:Eş güdümlü evrensel zaman...

```
var zaman = new Date ();  
  
zaman. setUTCMonth (9);  
  
document.write(zaman);
```

setUTCDate() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait ayın gün bilgisini 1 ile 31 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür.UTC:Eş güdümlü evrensel zaman...

```
var zaman = new Date ();  
  
zaman. setUTCDate (8);  
  
document.write(zaman);
```

setUTCHours() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait günün saat bilgisini 0 ile 23 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür.Çoklu ayarlama işlemi yapılabilir.(Sıralama:Saat,Dakika,Saniye,Milisaniye).UTC:Eş güdümlü evrensel zaman...

```
var zaman = new Date ();  
  
zaman. setUTCHours (15);  
  
document.write(zaman);
```

setUTCMinutes() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait saatin dakikasını 0 ile 59 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür.Çoklu ayarlama işlemi yapılabilir.(Sıralama:Dakika,Saniye,Milisaniye).UTC:Eş güdümlü evrensel zaman...

```
var zaman = new Date ();  
  
zaman. setUTCMinutes (30);  
  
document.write(zaman);
```

setUTCSeconds() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait dakikanın saniye bilgisini 0 ile 59 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür.Çoklu ayarlama işlemi yapılabilir.(Sıralama:Saniye,Milisaniye).UTC:Eş güdümlü evrensel zaman...

```
var zaman = new Date ();  
  
zaman. setUTCSeconds(55);  
  
document.write(zaman);
```

setUTCMilliseconds() : Kendisine parametre olarak belirtilmiş değer doğrultusunda değişken içeriğine evrensel zaman dilimine ait saniyenin milisaniye bilgisini 0 ile 999 arasında olmak üzere rakamsal olarak ayarlayarak depolar ve depoladığı değeri geriye döndürür. UTC:Eş güdümlü evrensel zaman...

```
var zaman = new Date ();
```

```
zaman. setUTCMilliseconds(500);  
  
document.write(zaman);
```

getTime() : Kullanıldığı değişken içeriğine tarihin zaman damgası bilgisini depolayarak,depoladığı değeri geriye döndürür.Zaman damgası başlangıç:1 Ocak 1970

```
var zaman = new Date ();  
  
document.write(zaman + "<br/>");  
  
var damga = zaman.getTime();  
  
document.write(damga);
```

setTime() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken zaman damgasına ait tarihin bilgisini depolayarak,depoladığı değeri geriye döndürür.(Zaman damgası başlangıç tarihi:1 Ocak 1970)

```
var zaman = new Date ();  
  
zaman.setTime(1480926330147);  
  
document.write(zaman);
```

now() : Kullanıldığı değişken içeriğine evrensel zaman dilimine ait tarihin zaman damgası bilgisini depolayarak,depoladığı değeri geriye döndürür.Zaman damgası başlangıç tarihi:1 Ocak 1970.UTC:Eş güdümlü evrensen zaman...

```
var zaman =Date.now ();  
  
document.write(zaman);
```

UTC() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine evrensel zaman dilimine ait tarihin zaman damgası bilgisini depolayarak,depoladığı değeri geriye döndürür.Zaman damgası başlangıç tarihi 1 Ocak 1970'dir. UTC:Eş güdümlü evrensen zaman...Parametre sıralaması :Yıl,Ay,Gün.Opsiyonel:Saat,Dakika,Saniye,Milisaniye.

```
var zamandamgası =Date.UTC(2010,10,24);  
  
document.write(zamandamgası);
```

parse() : Kendisine parametre olarak belirtilmiş değer doğrultusunda kullanıldığı değişken içeriğine tarihin zaman damgası başlangıcı ile arasındaki zaman farkı bilgisini depolayarak,depoladığı değeri geriye döndürür.Zaman damgası başlangıç tarihi 1 Ocak 1970'dir.Parametre sıralaması: Ay (İngilizce) Gün,Yıl...

```
var fark =Date.parse("july 5,1975");  
  
document.write(fark);
```

getTimezoneOffset() : Kullanıldığı değişken içeriğine yerel zaman ile 0 zaman dilimi arasındaki zaman farkı bilgisini dakika olarak depolayarak, depoladığı değeri geriye döndürür. GMT : Greenwich ortalama zamanı...

```
var zaman = new Date ();  
  
var fark = zaman.getTimezoneOffset();  
  
document.write(fark);
```

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

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

toUTCString() : Kullanıldığı değişken içeriğine atanmış olan tarih ve saat bilgilerini evrensel zaman diliminde alarak string (metin) veri türünde düz bir metne dönüştürerek, oluşturduğu değeri geriye döndürür. UTC: Eş güdümlü evrensel zaman...

toISOString() : Kullanıldığı değişken içeriğine atanmış olan tarih ve saat bilgilerini ISO standartını kullanarak string (metin) veri türünde düz bir metne dönüştürerek, oluşturduğu değeri geriye döndürür. ISO: Uluslar arası standartlar organizasyonu...

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

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

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.

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.

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

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

NESNE TABANLI PROGRAMLAMA :

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

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

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

isPrototypeOf : Parametrik olarak girilen nesne prototype zincirinde, bir constructor (yapıcı metot) bulunup bulunmadığı kontrol ederek boolean veri türünde sonucu geriye döndürür.

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

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

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

- 1.parametre : Özelliğin adı
- 2.parametre : Özelliğin eski değeri
- 3.parametre : Özelliğin yeni değeri

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

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

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

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

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

DOM EVENTS :

onclick : Belge içerisindeki herhangi bir elemana Mouse (fare) ile sol tıkladığında belirtilecek olan fonksiyon çalışır.

```
<body>
    <buton onclick="ornek()" >TIKLA </button>
<script>
function ornek ( ){
    document.getElementById("sonucalani").innerHTML = "TIKLAMA İŞLEMİ GERÇEKLEŞTİ" ;
}
</script>
</body>
```

ondblclick : Belge içerisindeki herhangi bir elemana Mouse (fare) ile çift sol tıkladığında belirtilecek olan fonksiyon çalışır.

onmousedown : Belge içerisindeki herhangi bir elemana Mouse (fare) ile sol veya sağ tıkladığında belirtilecek olan fonksiyon çalışır.

oncontextmenu : Belge içerisindeki herhangi bir elemana Mouse (fare) ile sağ tıkladığında belirtilecek olan fonksiyon çalışır.

onmouseover : Belge içerisindeki herhangi bir elemanın Mouse (fare) ile üzerine gelindiğinde belirtilecek olan fonksiyon çalışır.

onmouseout : Belge içerisindeki herhangi bir elemanın Mouse (fare) ile üzerinden gidildiğinde belirtilecek olan fonksiyon çalışır.

onmouseenter : Belge içerisindeki herhangi bir elemanın Mouse (fare) ile kapsama alanına giriş yapıldığında belirtilecek olan fonksiyon çalışır.

onmouseleave : Belge içerisindeki herhangi bir elemanın Mouse (fare) ile kapsama alanından çıkış yapıldığında belirtilecek olan fonksiyon çalışır.

onmousemove : Belge içerisindeki herhangi bir elemanın Mouse (fare) ile kapsama alanında gezildiğinde belirtilecek olan fonksiyon çalışır.

onmouseup : Belge içerisindeki herhangi bir elemana Mouse (fare) ile sol basılı duran tıklama bırakıldığında belirtilecek olan fonksiyon çalışır.

onmousewheel & onwheel : Belge içerisindeki herhangi bir eleman üzerindeyken Mouse'un (farenin) tekerleği yukarı veya aşağı yuvarlandığında belirtilecek olan fonksiyon çalışır.

button : Belge içerisinde Mouse (fare) ile tıklama event'ı (olayı) gerçekleştiğinde mouse'un (farenin) hangi tuşuna basıldığı değerini geriye döndürür.

0 : Sol Tuş

1:Orta tuş & tekerlek

2:Sağ Tuş

buttons : Belge içerisinde Mouse (fare) ile tıklama event'ı (olayı) gerçekleştiğinde mouse'un (farenin) hangi tuşuna veya tuşlarına basıldığı değerini geriye döndürür.

1 : Sol Tuş

2:Sağ Tuş

3:Sol tuş ve aynı anda Sağ tuş

4:Orta tuş & tekerlek

5:Sol tuş ve aynı anda orta tuş & tekerlek

6:Sağ tuş ve aynı anda orta tuş & tekerlek

8: 4. tuş

16: 5.tuş

```
<body>
<div onmousedown="ornek()">Bu alanda mousun bir tuşuna basınız</div>
<div id=sonucalani></div>
<script>
function ornek ( ){
    var tusdegeri = event.button;
    if(tusdegeri==0){
        document.getElementById("sonucalani").innerHTML = "Mouse'un sol tuşuna basıldı"; }
    else if(tusdegeri==1){
        document.getElementById("sonucalani").innerHTML = "Mouse'un orta tuşuna ve tekerlek
tuşuna basıldı"; }
    else if(tusdegeri==2){
        document.getElementById("sonucalani").innerHTML = "Mouse'un sağ tuşuna basıldı"; } }
</script>
</body>
```

detail : Belge içerisinde Mouse (fare) ile tıklama event'ı (olayı) gerçekleştirildiğinde mouse'a (fareye) kaç defa tıklandığı sayısı değerini geriye döndürür.

```
<body>

    <buton onclick="ornek()" >TIKLA </button>

<script>
function ornek ( ){

    var say = event.detail;

    document.getElementById("sonucalani").innerHTML = "Butona tıklanma sayısı : " + say ;

}

</script>

</body>
```

screenX : Belge içerisindeki herhangi bir Mouse (fare) event'ı (olayı) gerçekleştiğinde bilgisayar ekranına göre Mouse'un (farenin) yatay konum koordinatının değerini geriye döndürür.

screenY : Belge içerisindeki herhangi bir Mouse (fare) event'ı (olayı) gerçekleştiğinde bilgisayar ekranına göre Mouse'un (farenin) dikey konum koordinatının değerini geriye döndürür.

clientX : Belge içerisindeki herhangi bir Mouse (fare) event'ı (olayı) gerçekleştiğinde browser (tarayıcı) penceresine göre Mouse'un (farenin) yatay konum koordinatının değerini geriye döndürür.

clientY : Belge içerisindeki herhangi bir Mouse (fare) event'ı (olayı) gerçekleştiğinde browser (tarayıcı) penceresine göre Mouse'un (farenin) dikey konum koordinatının değerini geriye döndürür.

pageX : Belge içerisindeki herhangi bir Mouse (fare) event'ı (olayı) gerçekleştiğinde browser (tarayıcı) penceresine / sayfaya göre Mouse'un (farenin) yatay konum koordinatının değerini geriye döndürür.

pageY : Belge içerisindeki herhangi bir Mouse (fare) event'ı (olayı) gerçekleştiğinde browser (tarayıcı) penceresine / sayfaya göre Mouse'un (farenin) dikey konum koordinatının değerini geriye döndürür.

onkeydown : Belge içerisindeki herhangi bir eleman üzerinde klavyeden bir tuşa basıldığında belirtilecek olan fonksiyon çalışır.

onkeypress : Belge içerisindeki herhangi bir eleman üzerinde klavyeden herhangi bir tuşa basıldığında belirtilecek olan fonksiyon çalışır.

onkeyup : Belge içerisindeki herhangi bir eleman üzerinde klavyeden basılı duran bir tuşa bırakıldığında belirtilecek olan fonksiyon çalışır.

key : Belge içerisindeki onkeypress,onkeydown veya onkeyup event'ı (olayı) gerçekleştiğinde klavyeden basılan tuşun değerini döndürür.

charCode : Belge içerisindeki onkeypress event'ı (olayı) gerçekleştiğinde klavyeden basılan tuşun Unicode (evrensel kod) kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür.

keyCode : Belge içerisindeki onkeypress,onkeydown veya onkeyup event'ı (olayı) gerçekleştiğinde klavyeden basılan tuşun Unicode (evrensel kod) kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür.

which : Belge içerisindeki onkeypress,onkeydown veya onkeyup event'ı (olayı) gerçekleştiğinde klavyeden basılan tuşun Unicode (evrensel kod) kodlamasına dönüştürerek oluşturduğu değeri geriye döndürür.

location : Belge içerisindeki onkeydown veya onkeyup event'ı (olayı) gerçekleştiğinde ilgili tuşun klavyedeki lokasyon / konum bilgisini değerini geriye döndürür.

0:Standart Tuşlar

1:Sol tuşlar

2:Sağ tuşlar

3:Numpad Tuşları

altkey : Belge içerisindeki herhangi bir event (olay) tetiklenirken alt tuşuna basılıp basılmadığını kontrol ederek boolean (mantıksal) veri türünde sonucu geriye döndürür.

ctrlkey : Belge içerisindeki herhangi bir event (olay) tetiklenirken ctrl tuşuna basılıp basılmadığını kontrol ederek boolean (mantıksal) veri türünde sonucu geriye döndürür.

shiftKey : Belge içerisindeki herhangi bir event (olay) tetiklenirken shift tuşuna basılıp basılmadığını kontrol ederek boolean (mantıksal) veri türünde sonucu geriye döndürür.

metaKey : Belge içerisindeki herhangi bir event (olay) tetiklenirken pencere tuşuna veya emir tuşuna (apple) basılıp basılmadığını kontrol ederek boolean (mantıksal) veri türünde sonucu geriye döndürür.

onload : Sayfanın veya herhangi bir HTML elemanının yüklenmesi tamamlandığında belirtilecek olan fonksiyon çalışır.

onunload : Sayfanın veya herhangi bir HTML elemanının kapatıldığında veya kaldırıldığında belirtilecek olan fonksiyon çalışır.

Onbeforeunload : Sayfanın kullanıcı tarafından yenileme ve kapatma isteklerinde işlem gerçekleştirilmeden belirtilecek olan fonksiyon çalışır.

onhaschange : Sayfa URL'inin bağlantı bölümünde değişiklik olur ise belirtilecek fonksiyon çalışır.

oldURL: Sayfa URL'inin bağlantı bölümünde değişiklik olduğundaki eski URL adresine geriye döndürür.

newURL: Sayfa URL'inin bağlantı bölümünde değişiklik olduğundaki yeni URL adresine geriye döndürür.

onpageshow: Sayfa kullanıcı tarafından görüntülenebiliyorsa belirtilecek olan fonksiyon çalışır.

onpagehide: Sayfa kullanıcı tarafından görüntülenemiyorsa belirtilecek olan fonksiyon çalışır.

ononline: Browser (tarayıcı) sayfayı çevrimdışı moda çalıştırırken,çevrimiçi moda çalıştırmaya başladığında belirtilecek olan fonksiyon çalışır.

onoffline: Browser (tarayıcı) sayfayı çevrimiçi moda çalıştırırken,çevrimdışı moda çalıştırmaya başladığında belirtilecek olan fonksiyon çalışır.

onresize: Kullanıcı tarafından browser'ın sayfa boyutu değiştirilmek istendiinde belirtilecek olan fonksiyon çalışır.

onscroll: Sayfanın veya herhangi bir HTML elemanın kaydırma çubuğu yukarı veya aşağı kaydırıldığında belirtilecek olan fonksiyon çalışır.

ontoggle: Belge içerisindeki HTML details elemanının içeriği açıldığında veya kapandığında belirtilecek olan fonksiyon çalışır.

oncanplay: Medya elemanı yürütülmeye hazır olduğunda belirtilecek olan fonksiyon çalışır.

onplay: Medya elemanı yürütülmeye başladığında belirtilecek olan fonksiyon çalışır.

onpause: Medya elemanı yürütülmesi durdurulduğunda belirtilecek olan fonksiyon çalışır.

onvolumechange: Medya elemanı ses seviyesi değiştirildiğinde belirtilecek olan fonksiyon çalışır.

onratechange: Medya elemanı yürütülme hızı değiştirildiğinde belirtilecek olan fonksiyon çalışır.

onseeked: Medya elemanının yürütme konumu kullanıcı tarafından yeni konuma getirildiğinde belirtilecek olan fonksiyon çalışır.

onseeking: Medya elemanının yürütülme konumu kullanıcı tarafından hareket ettirildiğinde belirtilecek olan fonksiyon çalışır.

ontimeupdate: Medya elemanının yürütülme konumu browser (tarayıcı) tarafından / kullanıcı tarafından değiştirildiğinde belirtilecek olan fonksiyon çalışır.

onended: Medya elemanı oynatma süresi bittiğinde belirtilecek olan fonksiyon çalışır.

onloadstart: Medya elemanı verisi browser (tarayıcı) tarafından aranmaya başlandığında belirtilecek olan fonksiyon çalışır.

onprogress: Browser (tarayıcı) medya verisini indirirken belirtilecek olan fonksiyon çalışır.

onloadedmetadata: Medya elemanı meta bilgilerini (boyut / süre vs.) verisi tamamen yüklendiğinde belirtilecek olan fonksiyon çalışır.

onloadeddata: Medya elemanı verisi yüklendiğinde belirtilecek olan fonksiyon çalışır.

oncanplaythrough: Medya elemanı verisi tamamıyla arabelleğe yüklendiğinde ve yürütülmeye hazır olduğunda belirtilecek olan fonksiyon çalışır.

onwaiting: Medya elemanının yürütülmesi arabelleğe daha fazla veri yüklenmesi için duraklatıldığında ve devam etmesi beklendiğinde belirtilecek olan fonksiyon çalışır.

onplaying: Medya elemanı yürütülmeye başlandığında ve yürütme işlemi arabelleğe yükleme vs. gibi işlemleri için durup başlayınca belirtilecek olan fonksiyon çalışır.

onsuspend: Browser (tarayıcı) medya elemanı verisibi bilerek indirmediğinde belirtilecek olan fonksiyon çalışır.

onerror: Sayfanın,HTML elemanının veya elemanı verisinin yüklenmesi browser (tarayıcı) tarafından durdurulduğunda veya hata oluştuğunda belirtilecek olan fonksiyon çalışır.

onemptied: Medya elemanı kullanılamaz duruma geldiğinde (aniden erişilmediğinde / bağlantı kesildiğinde) belirtilecek olan fonksiyon çalışır.

onstalled: Browser (tarayıcı) medya elemanı verisini indirmeye çalışırken medya verisini alamıyorsa belirtilecek olan fonksiyon çalışır.

onabort: Sayfanın,HTML elemanının veya medya elemanı verisinin yüklenmesi durdurulduğunda veya iptal edildiğinde belirtilecek olan fonksiyon çalışır.

ondurationchange: Medya elemanının nesnesinin süresi değiştirildiğinde belirtilecek olan fonksiyon çalışır.

onfocus: Form içerisindeki bir elemana odaklanıldığında belirtilecek olan fonksiyon çalışır.

```
<body>
    Adınız Soyadınız : <input id="isimsoyisim" type="text" onfocus="ornek()">
</script>
function ornek () {
    document.getElementById("isimsoyisim").style.backgroundColor="#F1F1F1";
    document.getElementById("isimsoyisim").style.color="red"; }
```

```
</script>
```

```
</body>
```

onblur: Form içerisindeki bir elemanın odağından çıkıldığında belirtilecek olan fonksiyon çalışır.

onfocusin: Form içerisindeki bir elemana odaklanılmak üzereyken belirtilecek olan fonksiyon çalışır.

onfocusout: Form içerisindeki bir elemana odağından çıkılmak üzereyken belirtilecek olan fonksiyon çalışır.

onchange: Form içerisindeki bir elemanın değeri veya seçimi değiştirildiğinde belirtilecek olan fonksiyon çalışır.

oninput: Form içerisindeki bir elemanın değeri veya seçimi değiştirilirken belirtilecek olan fonksiyon çalışır.

oninvalid: Form içerisindeki bir elemanın değeri geçersiz ise belirtilecek olan fonksiyon çalışır.

onsearch: Form içerisindeki arama alanı işleme alındığında belirtilecek olan fonksiyon çalışır.

onsubmit: Belge içerisindeki form gönderilmek istenirse belirtilecek olan fonksiyon çalışır.

onreset: Belge içerisindeki form sıfırlanmak istenirse belirtilecek olan fonksiyon çalışır.

onselect: Form içerisindeki bir eleman metni seçilirse belirtilecek olan fonksiyon çalışır.

oncopy: Form veya belge içerisindeki bir elemanın metin içeriği kullanıcı tarafından kopyalandığında belirtilecek olan fonksiyon çalışır.

oncut: Form veya belge içerisindeki bir elemanın metin içeriği kullanıcı tarafından kesildiğinde belirtilecek olan fonksiyon çalışır.

onpaste: Form veya belge içerisindeki bir elemanın metin içeriği kullanıcı tarafından herhangi bir içerik yapıştırıldığında belirtilecek olan fonksiyon çalışır.

ondrag: Belge içerisindeki herhangi bir eleman sürüklendiğinde belirtilecek olan fonksiyon çalışır.

ondragstart: Belge içerisindeki herhangi bir eleman sürüklenmesi başladığında belirtilecek olan fonksiyon çalışır.

ondragend: Belge içerisindeki herhangi bir eleman sürüklenmesi bittiğinde belirtilecek olan fonksiyon çalışır.

ondragenter: Belge içerisindeki herhangi bir eleman hedef kapsama alanı üzerine giriş yaptığında belirtilecek olan fonksiyon çalışır.

ondragleave: Belge içerisindeki herhangi bir elemanın hedef kapsama alanından çıkış yaptığında belirtilecek olan fonksiyon çalışır.

ondragover: Belge içerisindeki herhangi bir elemanın hedef kapsama alanı üzerindeyken belirtilecek olan fonksiyon çalışır.

ondrop: Belge içerisindeki herhangi bir elemanın hedef kapsama alanı üzerine bırakıldığında belirtilecek olan fonksiyon çalışır.

onbeforeprint: Belge içeriği kullanıcı tarafından yazdırılmak istendiğinde yazdırma işlemi başlamadan belirtilecek olan fonksiyon çalışır.

onafterprint: Belge içeriği kullanıcı tarafından yazdırılmak istendiğinde yazdırma işlemi başladıktan sonra belirtilecek olan fonksiyon çalışır.

type: Belge içerisinde tanımlı herhangi bir event (olay) türü adının referansını elde etmek için kullanılır.

currentTarget: Belge içerisinde tanımlı event'ı (olayı) tetikleyen herhangi bir HTML elemanının event (olay) işaretleyicileri / dinleyicileri tarafından tag (etiket) adı referansını elde etmek için kullanılır.

target: Belge içerisinde tanımlı event'ı (olayı) tetikleyen herhangi bir HTML elemanının tag (etiket) adı referansını elde etmek için kullanılır.

timeStamp: Belge içerisinde tanımlı event'ın (olayın) oluşturulduğu zaman ile çalıştırıldığı zaman arasındaki fark süresinin mikrosaniye olarak referansını elde etmek için kullanılır.

cancelable: Belge içerisinde tanımlı event'ın (olayın) iptal edilebilir olmadığını kontrol ederek boolean (Mantıksal) veri türünde sonucu geriye döndürür.

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

defaultPrevented : Belge içerisinde tanımlı event (olay) için preventDefault() metodunun çağrılıp çağrılmadığını kontrol ederek boolean (mantıksal) veri türünde sonucu geriye döndürür.

BOM EVENTS :

BOM-WINDOWS

open: Yeni browser (tarayıcı) penceresi açmak için kullanılır.

1.parametre : URL (pencerenin adresi)

2.parametre : Pencerenin adı

3.parametre : Pencerenin özellikleri

4.parametre : Pencerenin geçmiş bilgisi

opener: Yeni pencereyi oluşturan pencereye belirtilecek olan referans değerini döndürür.

close(): Geçerli pencereyi kapatmak için kullanılır.

closed: Bir pencerenin kapatılıp kapatılmadığını kontrol ederek,boolean (mantıksal) veri türünde sonucu geriye döndürür.

name: Pencere adının referansını elde etmek veya ayarlamak için kullanılır.

length: Pencere içerisindeki tüm çerçeveleri sayısını elde etmek için kullanılır.

frames: Pencere içerisindeki tüm çerçevelerin referansını elde etmek veya ayarlamak için kullanılır.

parent: Geçerli çerçevenin üst penceresini elde etmek veya ayarlamak için kullanılır.

Top & self : İçerisinde bulunan pencerenin en üst seviye pencere olup olmadığını kontrol ederek boolean (mantıksal) veri türünde sonucu geriye döndürür.

stop(): Pencerenin yüklenmesini durdurmak için kullanılır.

history: Pencerenin geçmiş nesnesine erişim için kullanılır.Özellikleri sayesinde geçmiş işlemleri yapılabilir.

Length : Sayfa geçmişinde sayısını bulur

Back() : Sayfa geçmişinde 1 geri alır.

Forward() : Sayfa geçmişinde 1 ileriye alır.

Go() : Sayfa geçmişinde belirtilen değer kadar ileri veya geri gider.

confirm(): Kullanıcı işlemlerinde onay penceresi oluşturmak için kullanılır.

prompt(): Kullanıcıdan bilgi talep etmek için girdi penceresi oluşturmak için kullanılır.

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

atob(): Pencere içerisindeki herhangi bir metnin base64 kodlamasına dönüştürülmüş karakterini çözerek değeri geriye döndürür.

screenX: Pencerenin bilgisayar ekranına göre yatay konum koordinatının değerini geriye döndürür.

screenY: Pencerenin bilgisayar ekranına göre dikey konum koordinatının değerini geriye döndürür.

screenleft: Pencerenin bilgisayar ekranına göre yatay konum koordinatının değerini geriye döndürür.

screenTop: Pencerenin bilgisayar ekranına göre dikey konum koordinatının deęerini geriye döndürür.

innerWidth: Pencere ierik alanının genişlik deęerini elde etmek iin kullanılır.

innerHeight: Pencere ierik alanının yükseklik deęerini elde etmek iin kullanılır.

outerWidth: Pencerenin tüm alanının genişlik deęerini elde etmek iin kullanılır.

outerHeight: Pencerenin tüm alanının yükseklik deęerini elde etmek iin kullanılır.

pageXOffset: Belge ierięinin yatay kaydırma deęerini elde etmek iin kullanılır.

pageYOffset: Belge ierięinin dikey kaydırma deęerini elde etmek iin kullanılır.

scrollX: Belge ierięinin yatay kaydırma deęerini elde etmek iin kullanılır.

scrollY: Belge ierięinin dikey kaydırma deęerini elde etmek iin kullanılır.

resizeBy: Pencerenin mevcut genişlik ve yükseklik deęerini belirtilen pixel'lere göre yeniden boyutlandırmak iin kullanılır.

resizeTo(): Pencerenin mevcut genişlik ve yükseklik deęerini belirtilen pixel'lere göre yeniden boyutlandırmak iin kullanılır.

moveBy(): Pencerenin sol üst köşe koordinatı baz alınarak bilgisayar ekranına göre belirtilen konuma taşımak iin kullanılır.

moveTo(): Pencerenin sol üst köşe koordinatı baz alınarak bilgisayar ekranına göre belirtilen konuma taşımak iin kullanılır.

scrollBy(): Belge ierięini belirtilen deęere göre kaydırmak iin kullanılır.

scrollTo(): Belge ierięini belirtilen deęere göre kaydırmak iin kullanılır.

BOM-SCREEN

width: Kullanıcının toplam ekran genişliği deęerini elde etmek iin kullanılır.

height: Kullanıcının toplam ekran yüksekliği deęerini elde etmek iin kullanılır.

availWidth: Kullanıcının windows görev ubuęu hari ekranının genişlik deęerini elde etmek iin kullanılır.

availHeight: Kullanıcının windows görev ubuęu hari ekranının yükseklik deęerini elde etmek iin kullanılır.

colorDepth: Kullanıcının ekranı iin tanımlanmış olan renk derinliği / renk deęerini elde etmek iin kullanılır.

pixelDepth: Kullanıcının ekran çözünürlüğü için tanımlanmış olan her pixel başına bit değerini elde etmek için kullanılır.

BOM-NAVİGATOR

appName: Browser'ın kod adı referansını elde etmek için kullanılır.

appName: Browser adının referansını elde etmek için kullanılır.

appVersion: Browser'ın sürüm bilgisi referansını elde etmek için kullanılır.

language: Browser'ın dil bilgisi referansını elde etmek için kullanılır.

product: Browser'ın hangi görüntüleme motorunu kullandığı bilgisini elde etmek için kullanılır.

userAgent: Browser'ın tam model referansını elde etmek için kullanılır.

platform: Browser'ın hangi platform üzerinde çalıştığı bilgisi referansını elde etmek için kullanılır.

online: Browser'ın çevrimiçi moda çalıştırılmadığını kontrol ederek,boolean (mantıksal) veri türünde sonucu geriye döndürür.

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

javaEnabled (): Browser'da java kodlarının etkinlik durumunu kontrol ederek , boolean veri türünde sonucu geriye döndürür.

BOM-LOCATION

pathname: Bir URL'in path (yol) bilgisi referansını elde etmek veya ayarlamak için kullanılır.

search: Bir URL'in sorgu dizesi bilgisi referansını elde etmek veya ayarlamak için kullanılır.

protocol: Bir URL'in protokol adı referansını elde etmek için kullanılır.

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

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

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

ZAMANLAYICILAR:

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

Süre değeri : Milisaniye 1000 Ms.=1 s.

```
<body>
<script>
    function ornek(){
        alert ( "5 saniye süreniz doldu");}
setTimeout(ornek,5000);
</script>
</body>
```

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

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

Süre değeri : Milisaniye

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