**Değişken Türleri**

Değişkenler number , String ve boolean olarak üçe ayrılıyor . Bunların tiplerini typeof ile öğrenebiliriz . String olmayan bir değişkeni toString( ) ile stringe çevirebiliriz .

var sayi = 5

var karakter = 'Selamlar'

var anahtar = true

var kayanNoktali = 5.131

console.log(typeof sayi) //number

console.log(typeof kayanNoktali) //number

console.log(typeof karakter) // string

console.log(typeof anahtar) // boolean

console.log(typeof anahtar.toString()) // string

console.log(typeof sayi.toString()) // string

**İf Else ve Ternarary İf Else**

Düz if else aynı bildiğimiz if else . Ternarary if else’si ise şu şekilde :

var emeklilik = ('Şart') ? 'Olumlu Durum' : 'Olumsuz Durum'

console.log(emeklilik) // Olumlu Durum

**Stringler Detay**

String bir değişkeni başka bir string içerisinde kullanmak istersek ‘ü’ harfinin altındaki tuş + AltGr ’ ye bastığımızda çıkan üst tırkan özel bir üst tırnaktır , şekli ise şudur 🡺 **` `** 🡸

Kullanımı :

var isim = 'Umut`Gedik'

console.log(`Benim Adım ${isim}`)

**String Metodları**

var cumle = 'Vinn Bas Gaza iste Sana tas kafa'

var sonuc =''

sonuc = cumle.toLowerCase()//harfleri küçültür

//vinn bas gaza iste sana tas kafa

sonuc = cumle.toUpperCase()//harfleri büyültür

//VINN BAS GAZA ISTE SANA TAS KAFA

sonuc = cumle.length//uzunluk

//32

sonuc=cumle.slice(0,11)//0.indexten 11.indexe(hariç) kadar al dedik

//Vinn Bas Ga

sonuc = cumle.replace('Vinn','Keke')//ilkini ikincisiyle değiştirir

//Keke Bas Gaza iste Sana tas kafa

sonuc=cumle.indexOf('Gaza',9)//'Gaza' stringinin bailangıç indexini döndürür , ikinci parametre soldan kaçıncı indexten itibaren arıyacağını söyler , bulamazsa -1 döndürür

//9

sonuc = cumle.split(' ')//içerisine gönderilen stringe göre bölme yapar , burada her boşluklar arasındaki stringleri string dizisine döndürür

//(7) ['Vinn', 'Bas', 'Gaza', 'iste', 'Sana', 'tas', 'kafa'] dizisini döndürdü

console.log(sonuc)

**Number Metodları**

🡺 ‘Number’ metodu içerisine girilen string sayıyı number’a çevirir

degisken = Number('5')

console.log(typeof degisken) //number

🡺 ‘parseInt’ metoduda aynı işi yapar ancak farkı bir sonraki kısımda (parseFloat) gönderilen stringin sonu karakterse (Number) patlar bu patlamaz

degisken=parseInt('10.6')

console.log(degisken)//10

console.log(typeof degisken) //number

🡺 ‘parseFloat’ metodu girilen stringi numbera float şeklinde çevirir

degisken=parseFloat('10.6')

console.log(degisken)//10.6

console.log(typeof degisken)//number

Aşağıda gönderilen sayı stringin sonuna ve başına ayrı ayrı karakter gönderilen 2 örnek mevcut , bu örneklerdeki olay (Number’)’da yok

degisken=parseFloat('10.6a')

console.log(degisken)//10.6 verdi , sonrasındaki stringleri sildi

degisken=parseFloat('a10.6')

console.log(degisken)// NaN çıktısı verdi

isNaN(degisken) ? console.log('true') : console.log('false') //true

🡺 ‘toPrecision’ ve ‘toFixed’ metodları float numberlarla alakalı basamak belirleyici metodlarır

degisken = (15.123456789).toPrecision(5) // Sadece ilk 5 rakamı getirir

(yuvarlayarak)

console.log(degisken)//15.123

degisken = (15.123456789).toFixed(5) // Virgülden sonraki ilk 5 rakamı getirir

console.log(degisken)//15.12346   (yuvarlayarak)

**Math Kütüphanesi**

* Math.round( ) : Aşağı ve yukarı yuvarlama yapar (hangi tarafa yakınsa)
* Math.ceil( ) : Yukarı yuvarlama yapar
* Math.floor( ) : Aşağı yuvarlama yapar
* Math.sqrt( ) : Karekök Hesaplar
* Math.pow( ) : Üs almadır
* Math.abs( ) : Mutlak değer
* Math.min( [ ] ) : Girilen Sayılar arasındaki en küçüğü döndürür
* Math.max( [ ] ) : Girilen Sayılar arasındaki en büyüğü döndürür
* Math.random( ) : [ 0 , 1 ) aralığında rastgele sayılar döndürür

degisken = Math.round(2.34) // aşağı ve yukarı yakınlığa göre yuvarlama yapar

console.log(degisken)//2

degisken=Math.ceil(2.34)//Yukarı Yuvarlama yapar

console.log(degisken)//3

degisken=Math.floor(2.34)//Aşağı Yuvarlama Yapar

console.log(degisken)//2

degisken=Math.sqrt(25)//karekök

console.log(degisken)//5

degisken=Math.pow(2,3)//2 üzeri 3

console.log(degisken)//8

degisken=Math.abs(-10)//mutlak değer

console.log(degisken)//10

degisken=Math.min(1,2,3,4,5,10,9,8,7,6)//Girilen sayıların en küçüğünü alır

console.log(degisken)//1

degisken=Math.max(1,2,3,4,5,10,9,8,7,6)//Girilen sayıların en büyüğünü alır

console.log(degisken)//10

degisken = Math.random()//0-1 arası rastgele sayı üretir

console.log(degisken)//random

**Date Objesi**

🡺Tanımlanması :

var tarih

tarih = new Date()

console.log(tarih)//Sun May 29 2022 19:31:31 GMT+0300 (GMT+03:00)

console.log(typeof tarih)//object

🡺 Get Metodları

// Get Metodları

console.log(tarih.getDate())//29 , ayın kaçıncı gününde olduğumuzu verir

console.log(tarih.getDay())//0 verdi , 0=>pazar 6=>Cumartesi olur

console.log(tarih.getFullYear())//20220 , Yılı çeker

console.log(tarih.getMinutes())//35 , Dakikayı çeker ve şuanki saat 19:35

console.log(tarih.getHours())// 19 , saati verir ve şuanki saat 19:35

🡺 Set Metodları

// Set Metodları

tarih.setFullYear(2025)

console.log(tarih)//Thu May 29 2025 19:38:03 GMT+0300 (GMT+03:00)

console.log(tarih.getFullYear)//2025

Bu set metodları ile daha önce tanımlanmış bir ‘Date’ objesinin özelliklerini değiştirebiliriz

🡺 Doğum günümü tanımlayayım

var dogumGunum = new Date(1999,7,27)//Doğum tarihimigirdim(0'dan başlıyor

indexler)

console.log(dogumGunum)

**Diziler**

Dizi Tanımlama ve elemanlara erişim :

var liste = [1,2,3,['Umut','Ali'],true]

console.log(liste[0])//1

console.log(liste[1])//2

console.log(liste[2])//3

console.log(liste[3][0])//Umut

console.log(liste[3][1])//Ali

console.log(liste[4])//true

**Dizi Metodları**

* liste.join( x ) : Bu metod içerisine gönderilen x parametresi ile dizideki elemanları tek tek ayırıp tek bir string olarak döndürür .

console.log(liste.join(','))//Çıktısı : Çınar,-43.12,Ahmet,Mehmet,false

console.log(liste.join(' '))//Çıktısı : Çınar -43.12 Ahmet Mehmet false

* liste.toString ( ) : Bu yöntem ise dizideki tüm elemanları virgülle ayırarak tek bir string olarak geri döndürür . (join) metoduna virgül göndermek gibidir

console.log(liste.toString())//Çıktısı : Çınar,-43.12,Ahmet,Mehmet,false

* Eleman Çıkarmak için liste.pop( ) ve liste.shift( ) metodları mevcuttur .

‘pop’ listenin sonundan , ‘shift’ ise listenin başından eleman siler ve bu metodlar sildikleri elemanı döndürür . Listeden silinerek bu metodlarca geri döndürülmüş değişkenler türlerini (number , boolean vb.) korur . Eğer listede silinecek eleman kalmamasına rağmen bu metodlar kullanılırsa ‘undefined’ döndürür , bu değer ‘isNaN’ metodu ile yakalanabilir .

eleman = liste.pop() // Son elemanı siler ve silinen elemanı kendi türü

ile(number,string,boolean vs.) döndürür

console.log(eleman) // false

eleman = liste.shift() //pop'un ilk elemanı silen versiyonudur

console.log(eleman) // Çınar

* Eleman eklemek için liste.push( ) ve liste.unshift( ) kullanabiliriz . ‘**push**’ listenin sonuna , ‘**unshift**’ ise listenin başına eleman ekler . Bu metodlar ilgili listelere eleman ekledikten sonra o dizinin bu ekleme işleminden sonraki uzunluğunu döndürür . Yani 5 elemanı bir diziye bu metotlarca yeni bir eleman eklendiğinde 6 sayısını verir .

eleman = liste.push('Yıldız Takımı') // Listeye sondan eleman ekler ve

eleman eklendikten sonra dizinin uzunluğunu döndürür

console.log(eleman)

eleman = liste.unshift('Hello there') // buda push'un baştan eklenmeli verisyonu ve geriye dizinin uzunluğunu döndürür

console.log(eleman)

* Liste birleştirmek için .concat( ) kullanılabilir yada benim tercih ettiğim ‘… < array >’ yöntemi kullanılabilir . Uygulaması :

var liste1 = [1,2,3]

var liste2=['a','b','c']

// Liste Birleştirme

var liste3 = [...liste1,...liste2]

console.log(liste3)//(6) [1, 2, 3, 'a', 'b', 'c']

liste3 = liste1.concat(liste2)

console.log(liste3)//(6) [1, 2, 3, 'a', 'b', 'c']

* liste4.splice( ) : Bu metodun ilk parametresi başlangıç indexi ikinci parametresi ise ilk index olan başlangıç indexinden itibaren kaç eleman silineceğidir . Gerekirse bundan sonra tanımlanacak sonsuz parametreler ise bahsi geçen listede bu silme işlemi yapıldıktan sonra aradaki boşluğa yerleştirilmek istenen elemanları alır . Geriye ise o an sildiği elemanları döndürür .

var liste4 = [1, 2, 3, 'a', 'b', 'c']

console.log(liste4.splice(2,3,'eklenecek\_eleman\_1','eklenecek\_eleman\_2'))// (3) [3, 'a', 'b']

console.log(liste4)// (5) [1, 2, 'eklenecek\_eleman\_1', 'eklenecek\_eleman\_2', 'c']

* numbers.map( ) : Bu metod dizi üzerindeki elemanları tek tek gezer , o an üzerinde durduğu elemanı içerisine parametre olarak alır ve dizi döndürür .

const numbers = [1,2,3,4]

const sonuc = numbers.map( sayi => sayi\*sayi)

console.log(sonuc)//(4) [1, 4, 9, 16]

**Obje Tanımlama(Sözlük)**

Tekil Obje tanımlama mevzusu . Python Dict – Dart Map yapısı gibi

var user = {

    'Ad':'Umut',

    'SoyAd':'Gedik',

    'yas':21,

    'Lokasyon':{

        'il':'Ankara',

        'ilce':'Sincan',

        'mahalle':'Mustafa Kemal Mah.'

    }

}

console.log(typeof user)//object

console.log(user.Ad)//Umut

console.log(user.SoyAd)//Gedik

console.log(user.yas)//21

console.log(user.Lokasyon)//{il: 'Ankara', ilce: 'Sincan', mahalle: 'Mustafa Kemal Mah.'}

console.log(user.Lokasyon.il)//Ankara

console.log(user.Lokasyon.ilce)//Sincan

console.log(user.Lokasyon.mahalle)//Mustafa Kemal Mah.

**Döngüler**

Döngülerimiz for,while,do-while,forEach,for-in ve for-of ’lardir .

**For :**

var liste1 = []

for (let index = 0; index < 10; index++) {

    liste1.push(index)

}

console.log(liste1)//(10) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

**While :**

var index = 0

var liste2 = []

while (index<10) {

    liste2.unshift(index)

    index++

}

console.log(liste2)//(10) [9, 8, 7, 6, 5, 4, 3, 2, 1, 0]

**Do-While :**

var i = 10

var liste3 = []

do {

    liste3.push(i--)

} while (i>0);

console.log(liste3)//(10) [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

**Foreach**

liste3.forEach(element => {

    element%2==0 ? liste3.splice(liste3.indexOf(element),1) : ''

})

console.log(liste3)

**For-in ve For-of birlikte :**

var isimler = ['Umut','Zeynep','Doğan','Mine','Bekir','Gamze']

for (const key in isimler) {

    console.log(isimler[key])

}

for (const iterator of isimler) {

    console.log(iterator)

}

For-in’de keyleri çekiyoruz For-of’ta direkt elemanları çekiyoruz .

**Fonksiyonlar**

Fonksiyonlarımız değer ödnüren , değer döndürmeyen , sonsuz parametre alabilen ve öz yinelemeli olarak çeşitlendirilebilir ve bunlar birbirleriyle karıştırılabilir . Sadece fonksiyon ya geriye geri değer döndürür yada geriye değer döndüremez .

🡺 Değer dönüdrmeyen fonksiyon örneği

function yazdir(){

    console.log('selam')

}

🡺 Değer döndüren fonksiyon örneği

function topla(sayi1,sayi2){

    return sayi1+sayi2

}

🡺 Sonsuz parametre alan 2 fonksiyon örneği

function carp(){

    var args = arguments ;

    var sonuc = 1

    for (const iterator of args) {

        sonuc\*=iterator

    }

    return sonuc

}

function carp2(){

    var sonuc = 1

    for (const iterator of arguments) {

        sonuc\*=iterator

    }

    return sonuc

}

🡺 Öz yinelemeli fonksiyon örneği

function faktoriyel(sayi) {

    if (sayi<0) {

        return 1

    }

    else if(sayi==1 || sayi==0){

        return 1

    }

    else{

        return (sayi\*faktoriyel(sayi-1))

    }

}

**Scope Mevzusu**

Değişken tanımlarken üç çeşit değişken tanımlayabiliyor bunlar : ‘var’ , ‘let’ ve ‘const’ .

Bunlar arasında ‘var’ global olup nerede tanımlanırsa tanımlansın yine heryerden erişilebilirdir .

Ancak ‘let’ ve ‘const’ global scope değillerdir . Bu ikisi kendi tanımlandıkları kod bloğunda erişilebilir olurlar . Eğer ana blokta tanımlanırlarsa o zaman yine heryerden eriilebilir olurlar

if (true) {

    var var\_degisken = 'var'//global scope

    let let\_degisken = 'let'

    const const\_degisken = 'const'

}

if(true){

    console.log(var\_degisken)

    console.log(let\_degisken)

    console.log(const\_degisken)

}