JavaScript Nedir?

JavaScript, 1995 yılında Netscape Communications tarafından geliştirilen bir proglamlama dilidir. Öncelik olarak sadece tarayıcılar için kullanılan bir dil iken hem istemci tarafında (tarayıcıda) hem de sunucu tarafında (Node.js gibi) geniş bir uygulama çevresi bulunmaktadır.

JavaScriptin en temel veri tiplerinden biri olan objeler yazılım geliştiricileri için oldukça önemli bir konudur. Bu makale de objenin ne olduğunu ve nasıl çalıştığını öğreneceksiniz.

Object (nesne) Tanımı :

Object (nesne) ; belirli özellikler ve metotlardan oluşan yapılardır.  içerisinde birden fazla değeri depolayan yapılardır.

Object (nesne) Nerelerde ve Ne İçin Kullanılır ?

Nesneler (objects) verileri düzenlemek ve gruplandırmak için kullanılan temel veri yapılarıdır. Objects (nesneler) , anahtar - değer çiftlerini depolamak için kullanılır ve bunun gibi birçok farklı kullanım alanı vardır. Bunlar :

1.Veri Depolamak :

const kullanici= {

ad: “Süleyman”,

soyad : “Öztürk”,

yaş : 23,

email : “suleyman@example.com”

};

Bu kullanım da kullanıcı adında bir obje oluşturduk. Oluşturduğumuz kullanıcı içerisine kullanıcı bilgilerini ekledik.

1. Veri Düzenleme :

kullanici.soyad= “Aras”; // Soyad “Aras” olarak güncellendi.

Objects (nesneler) verileri değiştirmek ve düzenlemek için kullanılır. Burada veriye eriştik ve içerisindeki değeri değiştirdik.

1. Veri Erişimi :

console.log(kullanici.soyad); // Öztürk

console.log(kullanici.yas); // 23

Burada anahtar kullanılarak nesne içerisindeki bilgilere kolay bir şekilde eriebiliriz.

1. Dinamik Veri Yapısı :

kullanici.ulke= “Türkiye”; // Yeni bir özellik ekledik

delete kullanici.email; // Var olan veriyi sildik

Objects(nesneler) veri eklememizi veya çıkarmamızı sağlar. Burada kullanıcı bilgisine yeni bir veri ekledik ve aynı zamanda var olan bir veriyi sildik.

1. JSON (Javascript Object Notation) :

Objects (nesneler), JSON formatında veri iletmek ve saklamak için kullanılır. JSON basit veri yapılarını temsil etmek için tasarlanmış veri değişim formatıdır.

const jsonData = JSON.stringify(kullanici); // Nesneyi JSON formatına dönüştürdük.

const parseData = JSON.parse(jsonData) // JSON verisini nesneye dönüştürdük.

Burada öncelikle nesneyi JSON formatına dönüştürdük. Dönüştürüken stringify fonksiyonunu kullandık. Stringify fonksiyonu herhangi bir javascript nesnesini alır ve onu string formatına çevirir.

Daha sonra JSON verisini nesneye dönüştürdük. Dönüştürürken de parse fonksiyonunu kullandık. Parse fonksiyonu string formatındaki veriyi javascript nesnesi yapar.

Object Keys ve Object Value :

JavaScript de bir objenin anahtarlarını (keys) ve değerlerini (values) elde etmek için iki yöntem kullanılır. ’object.keys()’ ve ‘object.values()’. Bu iki yöntem objenin özelliklerini dizi(array) olarak döndürür.

Object.keys() Kullanımı :

Bu yöntem objenin anahtarını (keys) dizi (array) olarak döndürür.

Örnek Kullanım :

var ogrenciler = {

ogrAd : “Berat”,

ogrSoyad : “Ateş”

ogrNo : 123456

};

var keys = Object.keys(ogrenciler);

console.log( keys);

// Çıktı olarak = [“ogrAd”, “ogrSoyad”, “ogrNo”] değerlerini verecek

Bu örnekte görüldüğü gibi “ogrAd”, “ogrSoyad” ve “ogNo” değerlere karşılık gelen ahtarlardır. Eleman içerisinde yer alan propertyleri (özellikleri) array olarak döndüren methodtur.

Örnek Kullanım :

const araba = {

marka : “Ford”,

model : “Focus”,

renk : “Mavi”

yil : 2015

};

const keys = Object.keys(araba);

console.log(keys);

// Çıktı olarak = [“marka” , “model” , “renk” , “yıl” ]

Örnekte de görüldüğü üzere anahtarlar; “marka” , “model” , “renk” ve “yıl”. Bu örnekte bir nesnenin anahtarlarını aldık ve bu anahtarları bir dizi olarak döndürdük.

Object.values() Kullanımı :

Bu yöntem objenin değerini (values) bir dizi (array) olarak döndürür.

Örnek Kullanım :

var ogrenciler = {

ogrAd : “Berat”,

ogrSoyad : “Ateş”,

ogrNumara : 123456

};

var values = Object.values(ogrenciler);

console.log( values);

// Çıktı olarak = [“Berat”, “Ateş”, 123456] değerlerini verecek

Bu örnekte görüldüğü gibi “Berat” , “Ateş” ve “123456” anahtarlarına karşılık gelen değerlerdir.

Örnek Kullanım :

const araba = {

marka : “Ford”,

model : “Focus”,

renk : “Mavi”,

yil = 2015

};

const values = Object.values(araba);

console.log(values);

// Çıktı olarak = [“Ford” , “Focus” , Mavi, “2015” ] değerlerini verecek.

Örnekte de görüldüğü üzere değerler; “Ford” , “Focus” , “Mavi” ve “2015”. Bu örnekte bir nesnenin değerlerini aldık ve bu değerleri bir dizi olarak döndürdük.

Object.keys() ve Object.values() Birlikte Kullanım Örneği :

const araba = {

marka : “Ford”,

model : “Focus”,

renk : “Mavi,

yil : 2015

};

const keys = Object.keys(araba);

const values = Object.values(araba);

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

console.log(`Anahtar : ${keys[i]} , Değer : ${values[i]}`);

}

// Çıktı olarak :

// Anahtar : marka , Değer : Ford

// Anahtar : model , Değer : Focus

// Anahtar : renk , Değer : Mavi

// Anahtar : yıl , Değer : 2015

Görüldüğü gibi objeler de anahtar - değer çifti ilişkisi bulunmaktadır. Object.keys() anahtar kelimeleri gösterirken , Object.values() anahtar kelimelere karşılık gelen değerleri bize gösterir. Bu iki yöntem kullanılarak obje içeriğini dolaşabilir ve özellikleri üzerinde işlemler yapabilirsiniz.

Obje Oluşturma İşlemi :

JavaScript de obje oluşturmak için birçok farklı yöntem kullanabiliriz. Bunlardan bazıları şunlardır :

1. Nesne Literali ile Objeler Oluşturma:

Bu yöntem sıklıkla kullanılan yöntemlerden biridir. Obje tanımlamak için süslü parantezler ‘{}’ kullanılır. Tabi bu objeleri oluştururken keys ve values çiftlerini belirtmelisiniz.

Örnek Kullanım :

var ogretmen = {

isim : “İbrahim”,

alan: “Yazılım”,

yas : 45

};

console.log(ogretmen) ;

1. Yeni Bir Objeyi Oluşturma(new Object) :

Yeni bir obje oluşturmak için ‘new Object()’ ifadesini kullanabilirsiniz.

Örnek Kullanım :

const personel = new Object();

personel.ad= "John";

personel.soyad= "Doe";

personel.yas= 50;

console.log(personel);

1. Constructor Functions (Obje oluşturan inşa fonksiyonları) :

Constructor functions ile yeni objeler oluşturabilirsiniz.

Örnek Kullanım :

function personel(ad,soyad,yas) {

this.ad = ad;

this.soyad = soyad;

this.yas = yas;

}

var personel = new personel(“Ahmet”, ”Kervan”, 35);

console.log(personel); // personel{ad: Ahmet, soyad: 'Kervan', yaş: 35}

1. Object.create() Yöntemi :

Mevcut bir objeden kalıtım yoluyla bazı bilgiler alarak yeni bir obje oluşturmaya yarar.

Örnek Kullanım :

let ogrenci = {

isim : "Melisa",

yas : 24,

display () {

console.log("İsim:", this.isim);

}

};

let std1 = Object.create(ogrenci);

std1.isim= "Ayşe";

std1.display(); // Çıktı : İsim: Ayşe