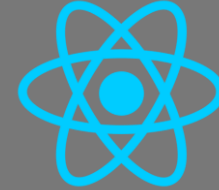


FRONT - END



JAVASCRIPT

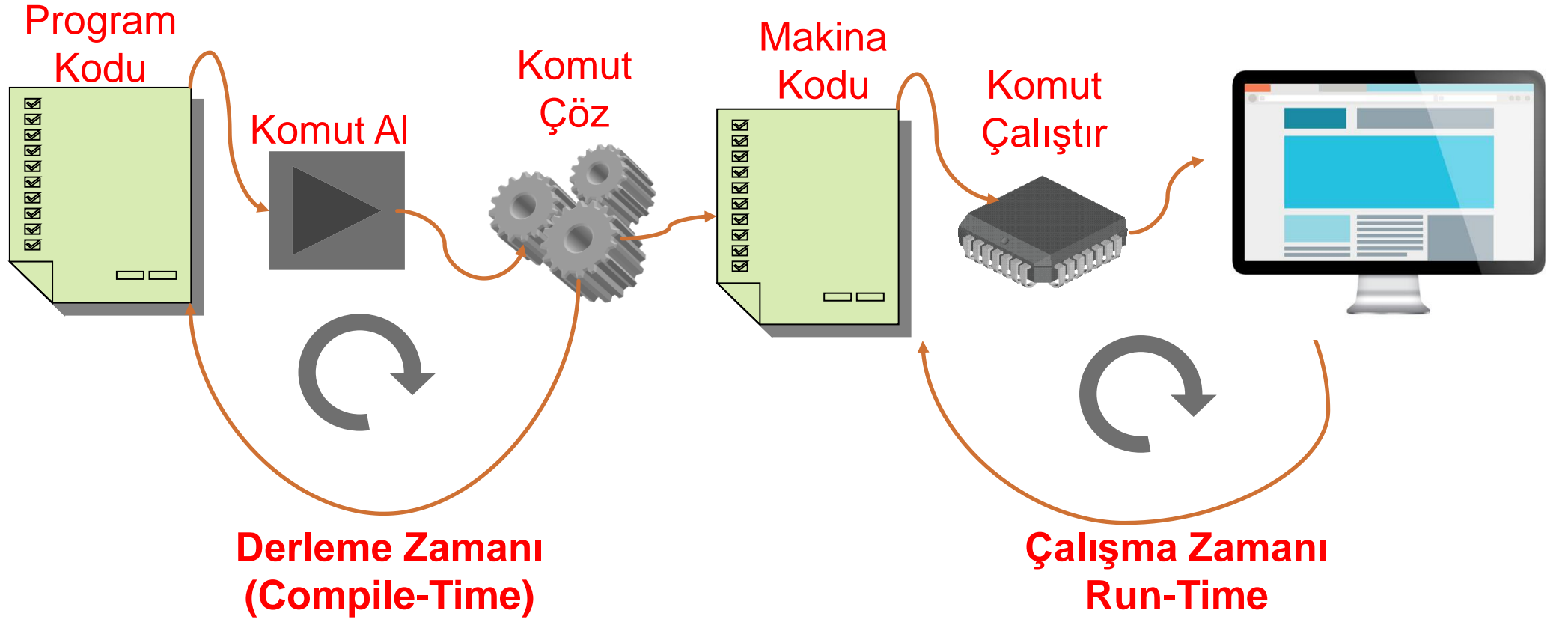


JAVASCRIPT'E GİRİŞ

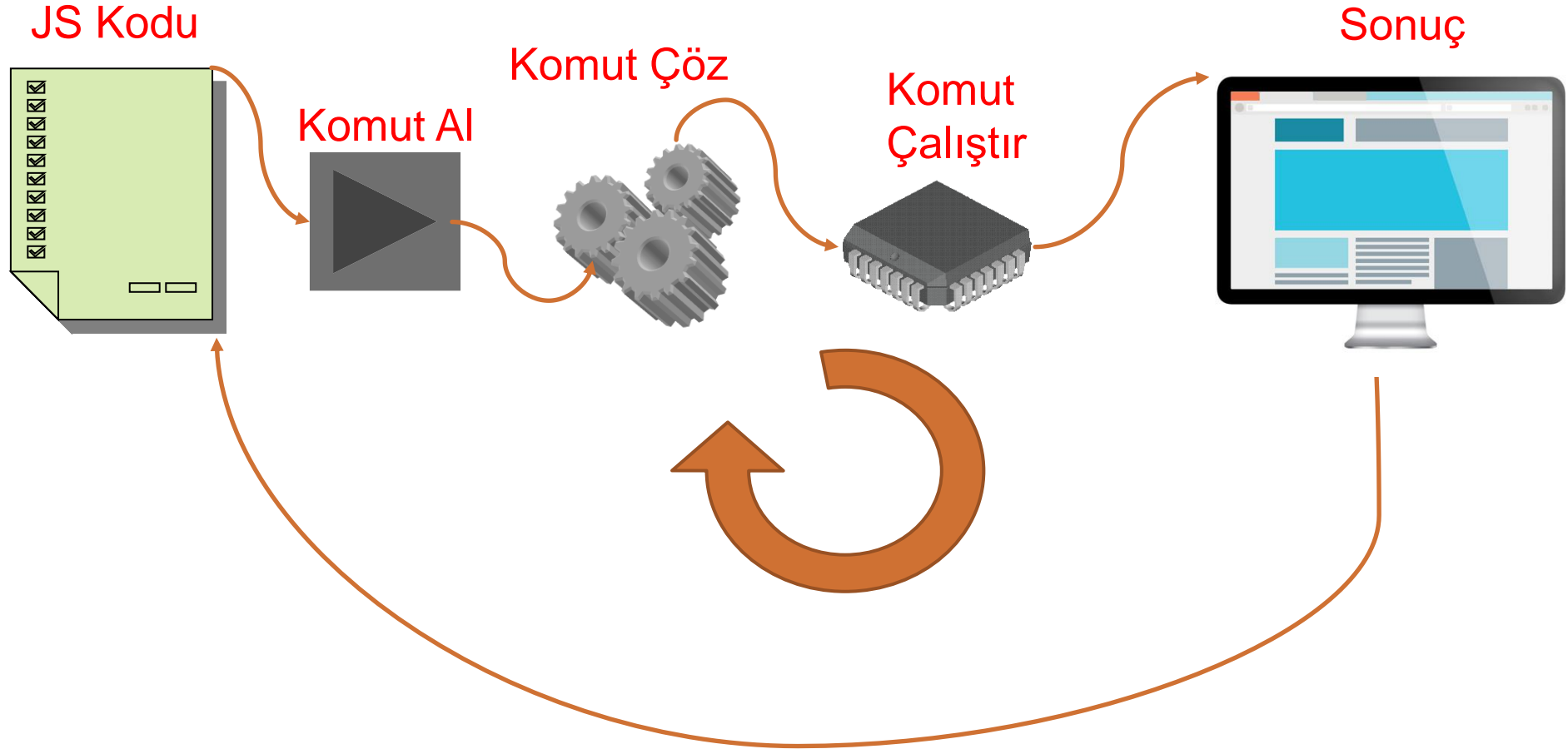
JAVASCRIPT NEDİR?

- **Javascript (JS):** Yüksek-Seviyeli, Nesne-Yönelimli, Yorumlayıcı-tabanlı ve dinamik bir programlama dilidir.
 - **Yüksek Seviye:** Hafıza yönetimi gibi karmaşık görevleri düşünmemize gerek yoktur.
 - **Nesne-Yönelimli(OOP):** Nesne özelliklerinin (kalıtım, çok şekillilik v.b) kullanılmasına imkan sağlamaktadır.
 - **Yorumlayıcı:** Derleyicide olduğu gibi tüm komutların bir kere de makine koduna çevirmek yerine **tek-tek** alınıp makine koduna çevrilip çalıştırılmasını sağlar.
- JS, **Tarayıcıların-dili** olarak da adlandırılır.

DERLEYİCİ (COMPILER)

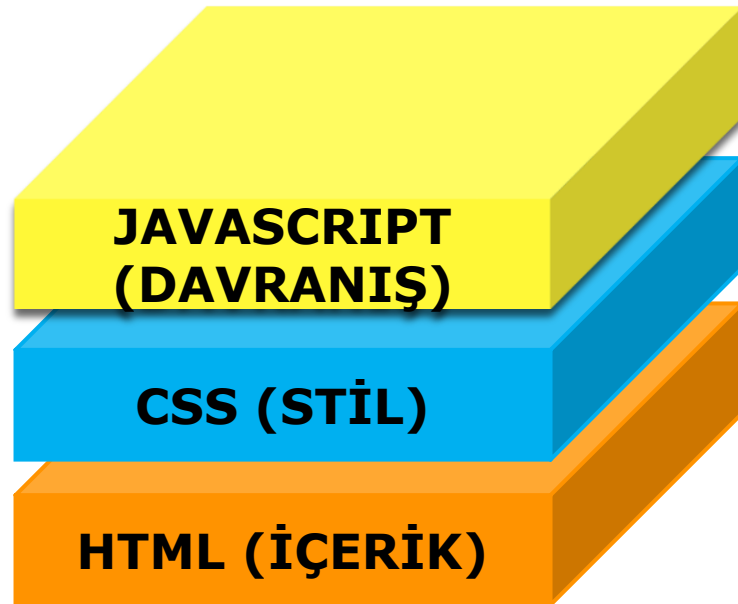


YORUMLAYICI (INTERPRETER)



JAVASCRIPT DİLİ NERELEERDE KULLANILIR?

- JS, **front-end** (kullanıcı-yüzü) geliştirmede en yaygın kullanılan dillerden birisidir.

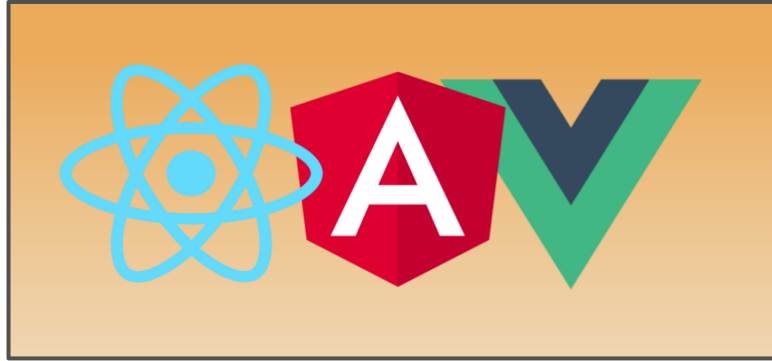


- **JS:** Tasarlanan sitelerin **davranışlarını** belirler (Butonların çalışması, açılır-kapanır menüler, renk değişimleri v.b).
- **CSS:** Girilen içeriklerin **stillerini** (Renk, Yazı-tipi v.b) belirler.
- **HTML:** Web sitelerinin statik **içeriklerinin** girilmesini sağlar.

JAVASCRIPT DİLİ NERELEERDE KULLANILIR?

- Son yıllarda JS, yazılan bir çok kütüphane sayesinde neredeyse her yerde geliştirme dili olarak kullanılır hale gelmiştir.

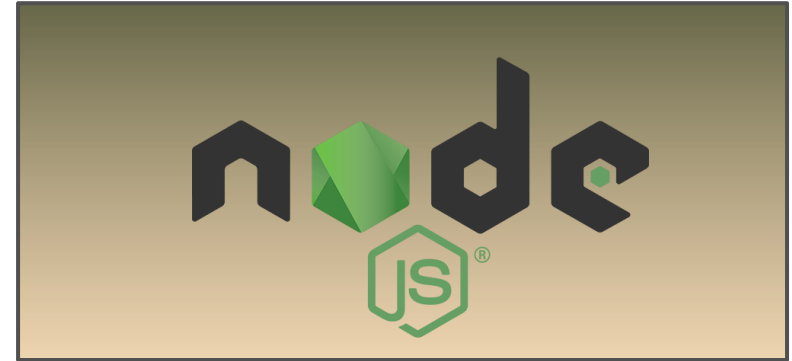
FRONT-END TASARIM



React, Angular ve **Vue** gibi JS kütüphaneleri ile interaktif Kullanıcı-yüzü tasarımı.

Örnekler: Facebook, Netflix, Dropbox, AirBnb vb.

BACK-END TASARIM

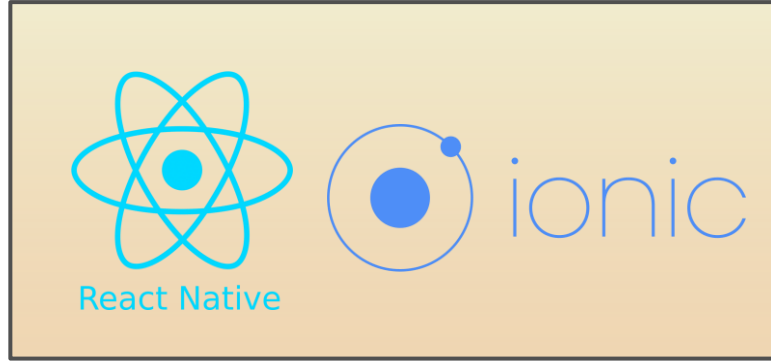


NodeJS kütüphanesi ile Sunucu yüzünde web uygulaması geliştirmede

Örnekler: Netflix, Uber, E-Bay vb.

JAVASCRIPT DİLİ NERELELERDE KULLANILIR?

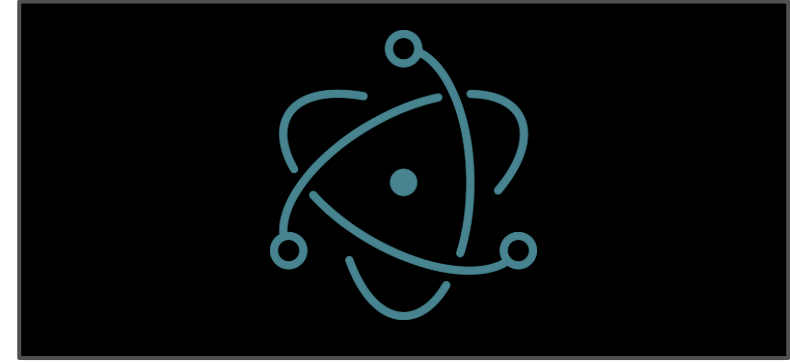
MOBİL TASARIM



React Native ve **Ionic** gibi kütüphaneler ile IOS ve Android için mobil uygulama geliştirmede.

Örnekler: Facebook, Instagram, Skype, Uber, Pinterest

MASAÜSTÜ TASARIM



Electron kütüphanesi ile Masaüstü PC'ler MACOS, Windows, Linux için uygulama geliştirmede.

Örnekler: VSCode, Whatsapp, Slack, Skype, Twitch, Teams

JAVASCRIPT'İN TARİHİ VE SÜRÜMLERİ

- **1995: Brendan Eich** ([Netscape](#) ve [Mozilla](#)'da çalışanı) tarafından **10 günde** üretildi. Adı o zamanlar **Mocha'ydı**.
- **1995:** Adı **LiveScript** olarak değiştirildi, sonra **Sun**'dan marka lisansı alındıktan sonra, ismi **JavaScript** olarak tekrar değiştirildi.
 - Bu, Java'nın o zamanlardaki popülerliğinden faydalanma hareketiydi.
 - **Java** ile **Javascript** tamamen farklı diller.
- **1996-1997**'de standart bir tarifname oluşturmak amacı ile JavaScript, [ECMA](#) tarafından ele alındı.
 - **ECMAScript**, resmi standart ismidir. JavaScript ise bu standart'ın en çok bilinen uygulamasıdır.
 - ECMAScript (**ES1**), JavaScript için ilk resmi standarttır.

JAVASCRIPT'İN TARİHİ VE SÜRÜMLERİ

- **2009:** Bir çok yenilik içeren ECMAScript5 (**ES5**) yayınlandı.
- **2015:** Javascript'teki en büyük güncelleme **ECMAScript2015 (ES6 /ES2015)** yayınlandı.
- Güncelleme başına daha az değişiklik olması için güncelleme çevrimi **yıllık** olarak değiştirildi.
 - ES2016 -> ES2017 -> ES2018 -> ES2019 -> ES2020 -> ... -> ES20??

JAVASCRIPT'İN TARİHİ VE SÜRÜMLERİ



- Yeni çıkan versiyonlar *arttırımsal* güncellemelerdir. Eski özellikler kaldırılMAmıştır.
- Ancak, eski tarayıcılar yeni JS sürümlerini desteklemeyebilir.
 - **ES5** → Tüm tarayıcılar desteklemektedir.
 - **ES6/ES2015 ve sonrası :**
 - Modern tarayıcılar desteklemektedir. Eski tarayıcılar desteklemeyebilir.
 - **Babel** gibi derleyiciler yardımıyla yüksek versiyonlar üretim aşamasında **ES5'e** çevrilerek eski tarayıcı uygunluğu sağlanabilir.

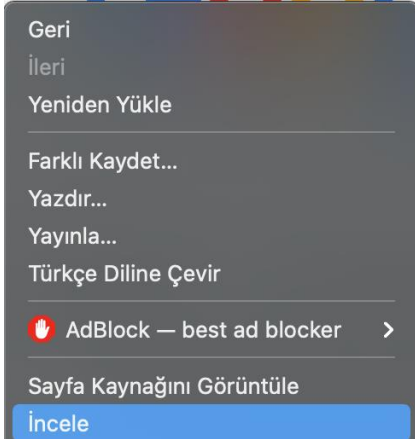
JAVASCRIPT'İN TARİHİ VE SÜRÜMLERİ

- Modern Javascript ile birlikte **Eski JS** versiyonları öğrenmenin faydaları
 - JS'nin nasıl çalıştığını daha iyi anlamaya yardımcı olur.
 - İnternet yada eski kitaplardaki örnekler ES5 standardına göre Javascript'i ele almıştır.
 - Bazı eski projeleri yeni standartlara güncelleyebilmek için temel mantığı bilmek yararlı olacaktır.

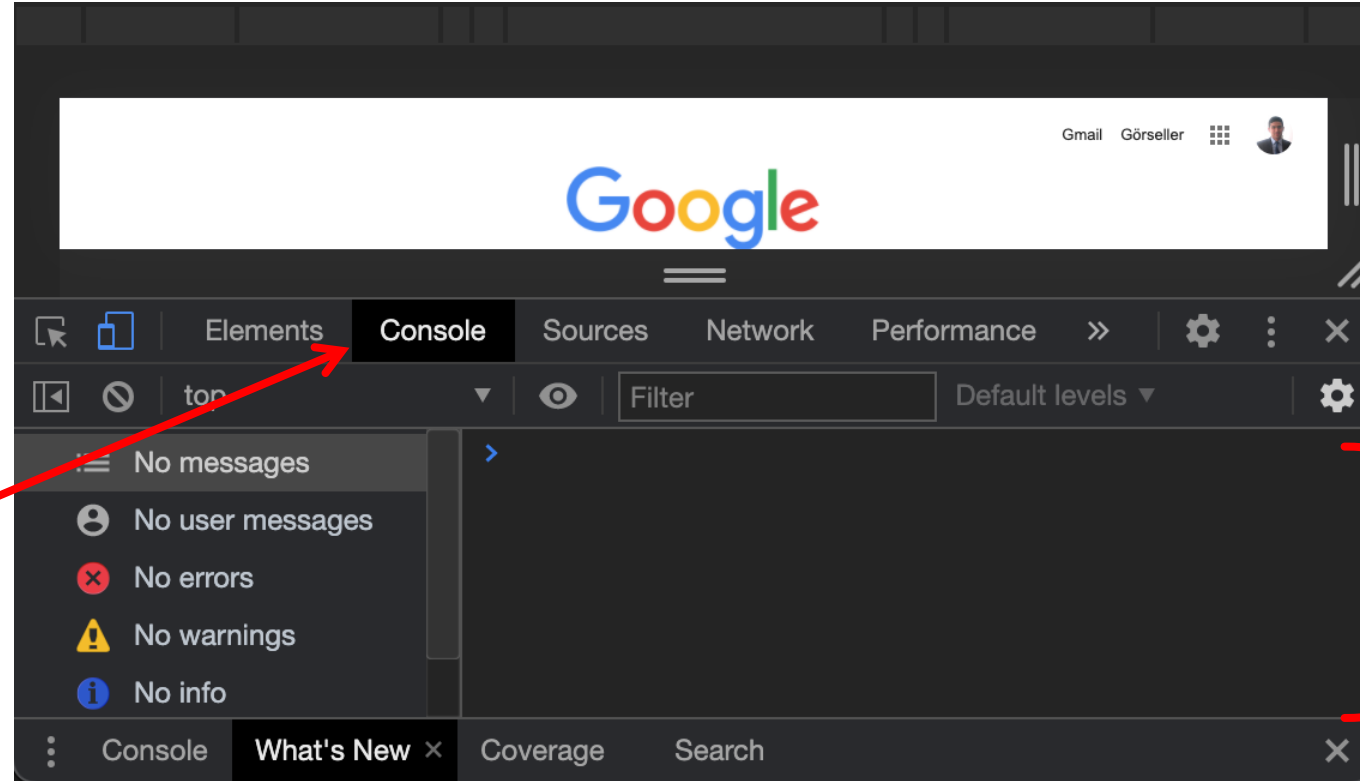
JAVASCRIPT'İN TARAYICIDA ÇALIŞTIRILMASI

- JS ile kod yazmak için **Google Chrome'u** kullanabiliriz.
 - Chrome'u açtıktan sonra sayfanın üzerinde sağ tıklayarak '**İncele**' (**Inspect**) sekmesine tıklarız ve Geliştirici Araçlarını açarız.

1



2
Console sekmesi
ile JS kodunu
çalıştırabileceğimiz
ekrana geçeriz.



3
Komut
satırında
komutlarımızı
yazarız

JAVASCRIPT'İN TARAYICIDA ÇALIŞTIRILMASI

- Konsolda ilk JS kodumuzu yazalım.
 - **console.log()** fonksiyonu ile konsola bir ifade bastırdık.

```
> console.log('Merhaba Javascript')
Merhaba Javascript                               VM1360:1
< undefined
> |
```

NOT: Konsol, bir uygulama geliştirme aşamasında hataların tespiti ve kontrol işlemleri için kullanılmaya çok elverişlidir.

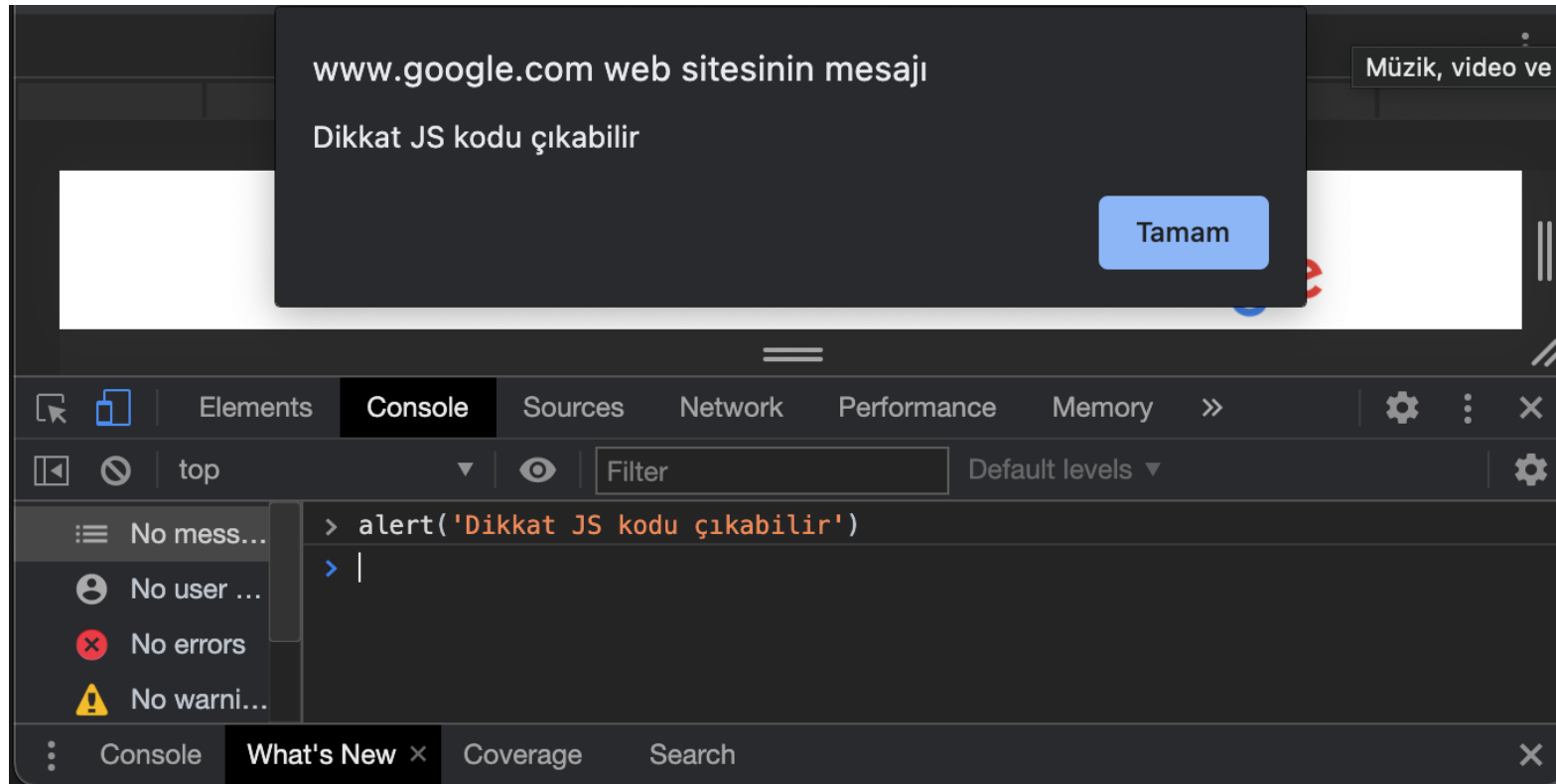
JAVASCRIPT'İN TARAYICIDA ÇALIŞTIRILMASI

- **console.error()** fonksiyonu ile konsola hata olduğunu bastırdık.
- **console.warn()** fonksiyonu ile konsola bir uyarı durumu oluşturduğunu yazdırdık.

```
> console.error("hata")
✖ ▶ hata VM1920:1
< undefined
> console.warn("uyarı")
⚠ ▶ uyarı VM1999:1
< undefined
```

JAVASCRIPT'İN TARAYICIDA ÇALIŞTIRILMASI

- **alert()** fonksiyonu açılan menü (pop-up) ile uyarı vermeye yarar.



JAVASCRIPT'İN TARAYICIDA ÇALIŞTIRILMASI

- Konsolda her türlü JS komutları yazıp çalıştırılabilir.

```
> 3+6+10-4  
< 15  
>
```

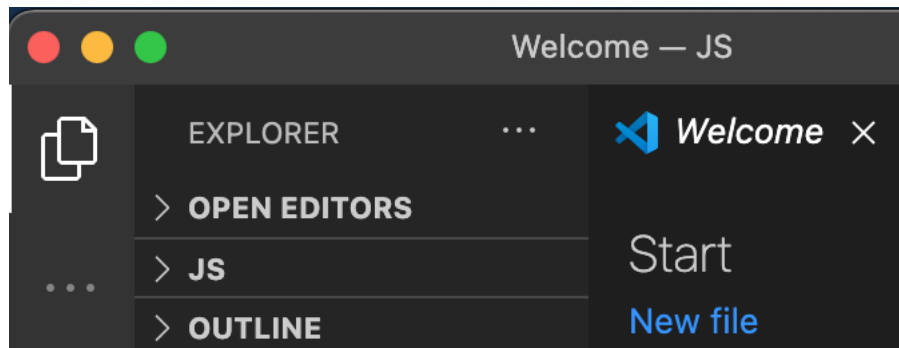
- **Ancak**, Chrome geliştirici araçları bir yazılım geliştirmekten ziyade **test** ve **hata tespiti** gibi konular için daha uygundur.
- Bu yüzden, **JS** kodlarımızı çalıştırmak için dosya oluşturmamız gerekir.
- Gerekli oldukça zaten **Chrome** Geliştirme araçlarını kullanacağız.

VSCODE KULLANIMI

- Javascript ile kod yazmak için PC'nizin Masaüstünde bir klasör oluşturup uygun bir isim (**JS**) verebilirsiniz.



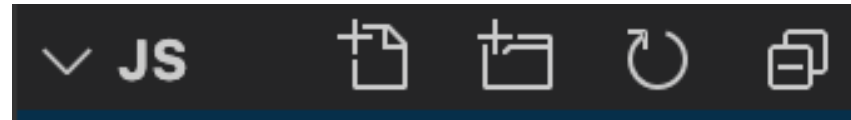
- VSCode'u açıp **File** -> **Open** -> **JS** -> **Aç** sekmeleri ile Masaüstünde bulunan **JS** klasörünü açınız.



İPUCU: Alternatif olarak Masaüstündeki **JS** klasörünü Vscode'a sürükleyerek de aynı şeyi yapmış olursunuz.

VSCODE KULLANIMI

- Soldaki menüde bulunan **JS** klasörüne tıklayarak menünün açılmasını sağlayın.



Yeni Dosya oluştur.

Yeni Klasör oluştur.

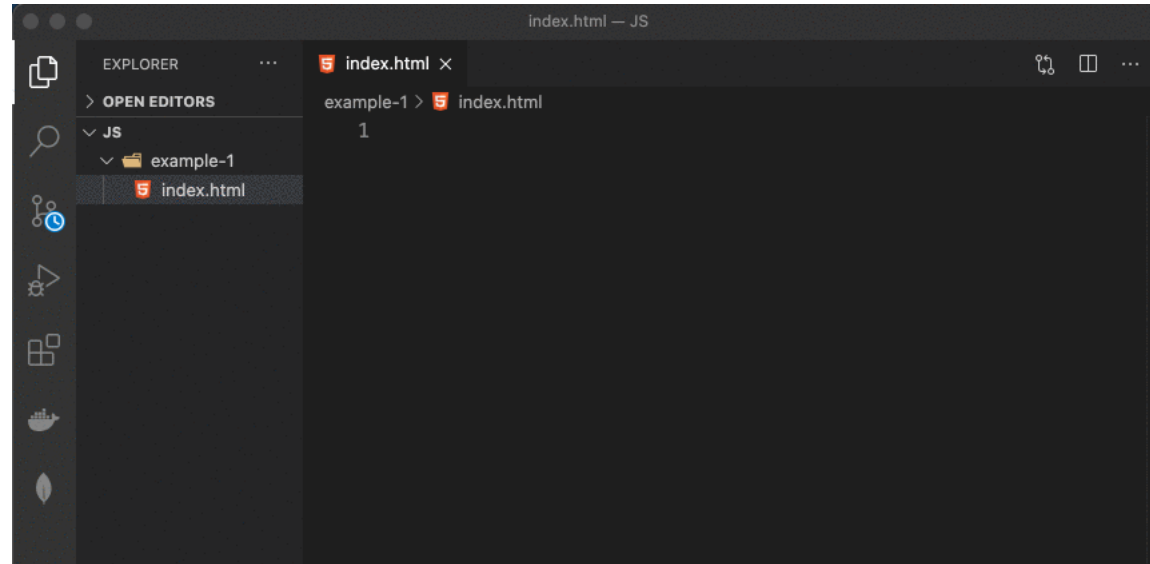
Yenile.

Açılır menüyü kapat.

1. İlgili simgeye tıklayarak **yeni bir klasör** daha oluşturunuz. Dosyaya ad olarak **example-1** verebiliriz.
2. **example-1** klasörünü seçerek içerisinde **index.html** adında **Yeni bir dosya** oluşturalım.

VSCODE KULLANIMI

- Böylelikle, bu pakette yazacağımız kodları saklayacağımız bir ana klasör (**JS**) ve bir alt klasör (**example-1**) oluşturduk.



- HTML kodu eklemek için **index.html** üzerinde **GIF**'de olduğu gibi **doc** yazıp **Enter** basmak yeterli.
- Artık HTML kodumuzun içerisine **JS kodunu** ekleyebiliriz.

VSCODE KULLANIMI

- **index.html** dosyamıza Javascript kodlarını eklemek için

<script> </script> etiketlerini kullanmalıyız.

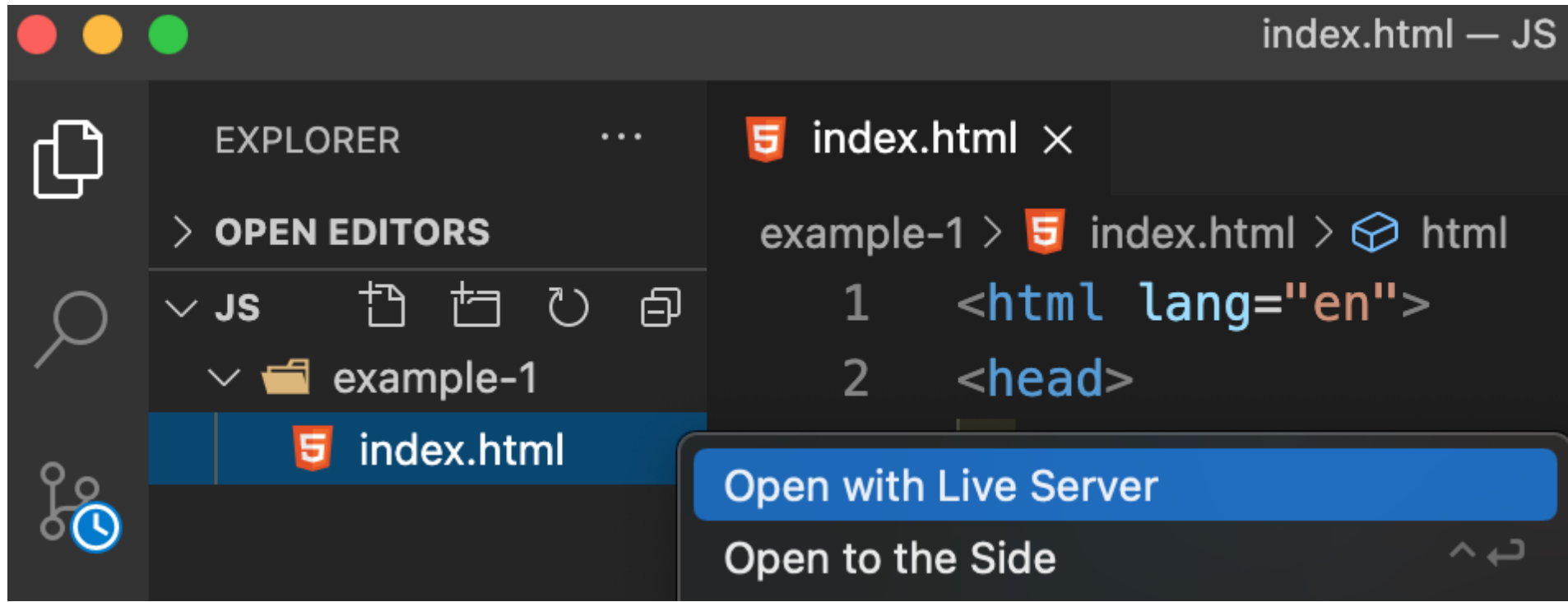
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>
<body>
  <h1>Javascript çok kullanışlı bir dil</h1>
  <script>
    console.log("Merhaba Javascript");
  </script>
</body>
</html>
```

VSCODE KULLANIMI

- Yazdığımız kodu çalıştırmak için :
- Masaüstünde bulunan klasörümüze gidelim.
Masaüstü -> JS -> example-1 -> index.html
- **index.html dosyasına**
 - Çift tıklayarak **Chrome'un** bu dosyayı açmasını sağlayabiliriz.
 - Olmaz ise, sağ tıklayarak Google Chrome ile aç diyebiliriz.
 - Veya **index.html** dosyasını açık bir Chrome sayfasına sürükleyebiliriz.

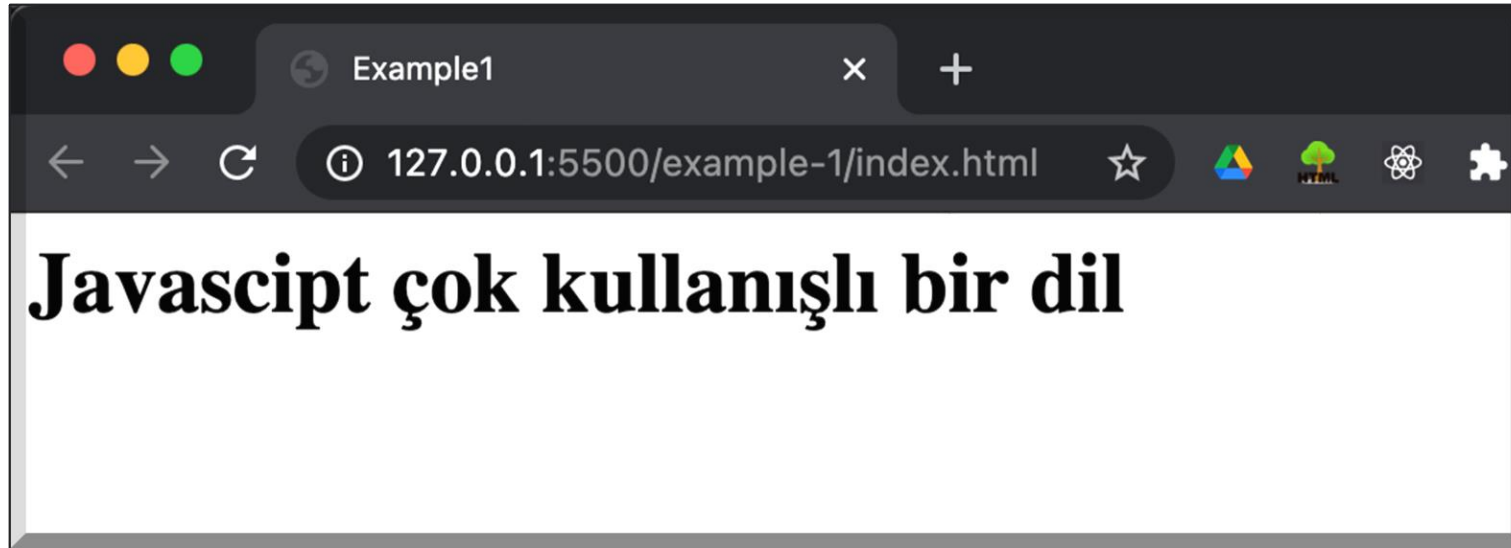
VSCODE KULLANIMI

- **İPUCU:** En kolay yöntem ise daha önceden yüklediğimiz **Live Server** Eklentisini kullanmak.



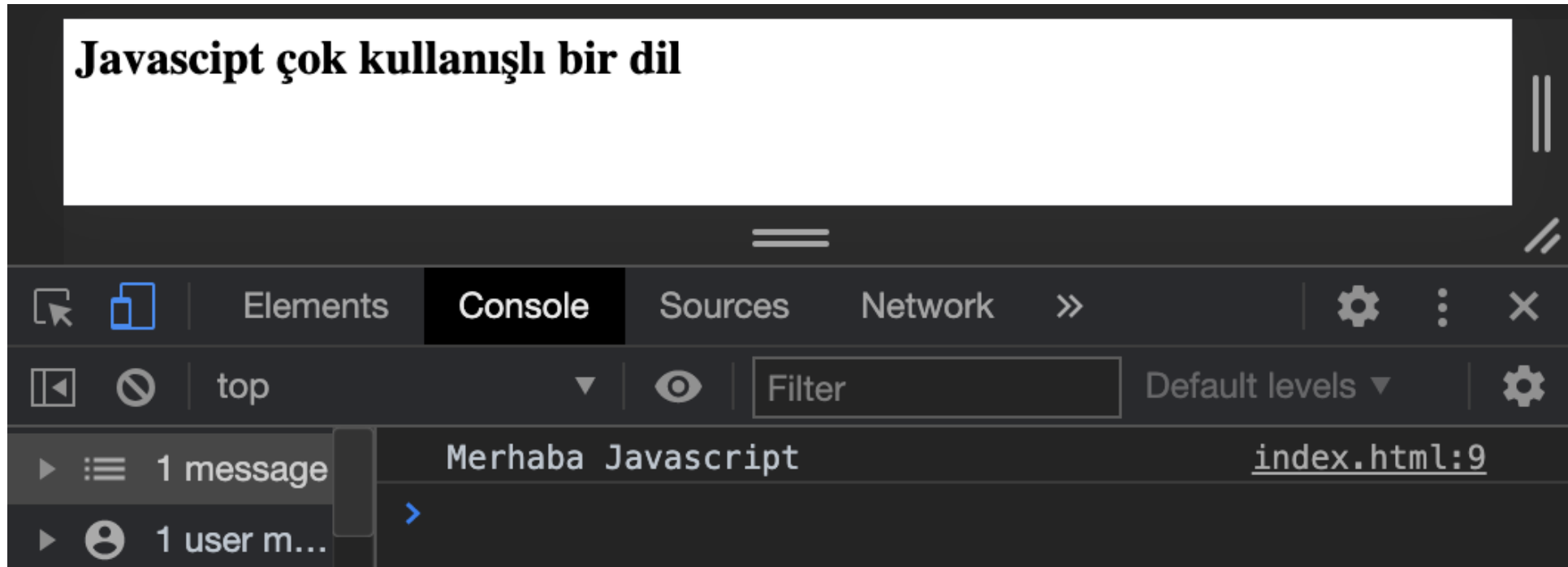
VSCODE KULLANIMI

- Evet, kodumuz çalıştı. Ancak, **Merhaba Javascript** yazısı ekranda görünmüyor ?




VSCODE KULLANIMI

- Bu çok normal çünkü o yazı ekranda değil sadece konsolda görülebilir.



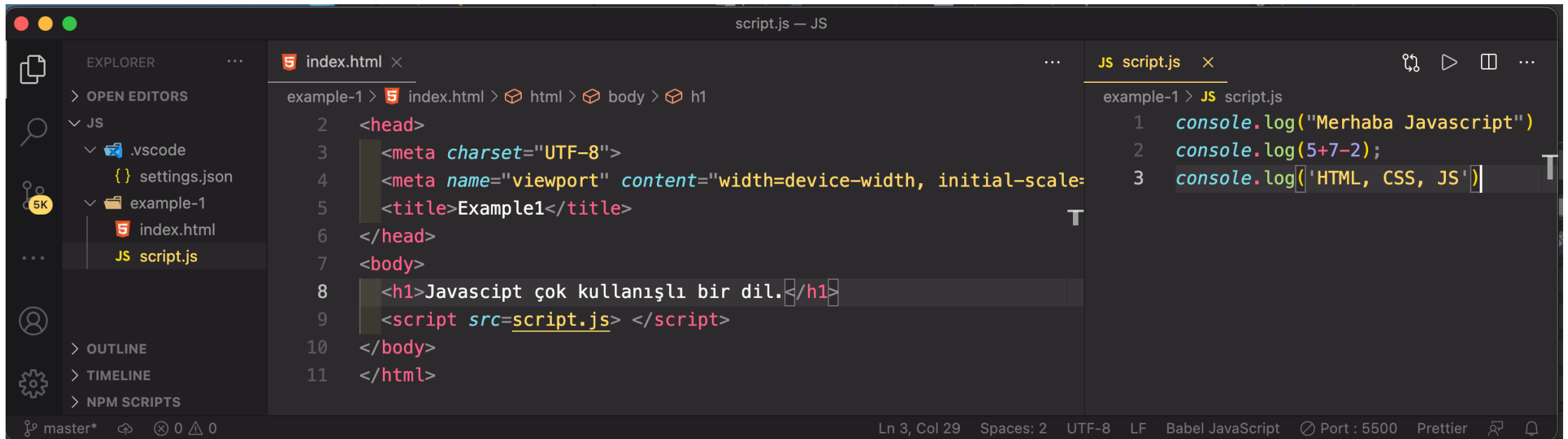
VSCODE KULLANIMI

- JS kodlarının bu şekilde HTML dosyasının içerisine eklenmesine **inline script** denilmektedir.
- **Inline script** JS komutlarını harici başka dosyaya ihtiyaç duymadan yazmak için elverişlidir.
- Ancak, genel kabul HTML, CSS ve JS dosyalarının **ayrı dosyalarda** tutulmasıdır.
- JS komutlarımızı ayrı bir dosyada tutmak için önceki adımlarda olduğu gibi
 - **example-1** klasörünü işaretlememiz,
 - **Yeni dosya ekleme**  butonuna tıklamamız ve
 - Bir js dosyası oluşturmamız gerekmektedir. Örneğin: **script.js**

JAVASCRIPT İÇİN VSCODE KULLANIMI

- **index.html**'de bulunan JS kodlarını kesip **script.js** dosyasına kopyalayabiliriz.
- **index.html** ile **JS** dosyasının bağlantısını sağlamak için;

```
<script src=script.js></script>
```



JAVASCRIPT İÇİN VSCODE KULLANIMI

- **index.html** ile **JS** dosyası farklı klasörlerde ise src linkine önce klasör adını yazmak gerekir.

```
<script src=/Ornek1/script.js></script>
```

- Eğer, **JS** kodu harici bir URL'de ise
 - Örneğin: **www.mysite.com/js** klasörü altında **myScript.js**

```
<script src="https://www.mysite.com/js/myScript.js"></script>
```