



BATCH : B45-46-71
LESSON : **Javascript**
DATE : 24.03.2022
SUBJECT : **Introduction**



 techproeducation.com

 info@techproeducation.com

 +1 (917) 768-7466



JavaScript



Javascript

Introduction

- › Javascript nedir?
- › Compiler & Interpreter
- › Kullanım amaçları
- › Sürümleri
- › Popülarite
- › İlk kodlar



Javascript nedir?



Yüksek seviyeli, nesne yönelimli, yorumlayıcı tabanlı ve dinamik bir programlama dilidir.

Yüksek Seviyeli

Hafıza yönetimi gibi karmaşık görevleri düşünmemize gerek yoktur.

Nesne Yönelimli

Nesne özelliklerinin (kalıtım, çok şekillilik v.b) kullanılmasına imkan sağlamaktadır.

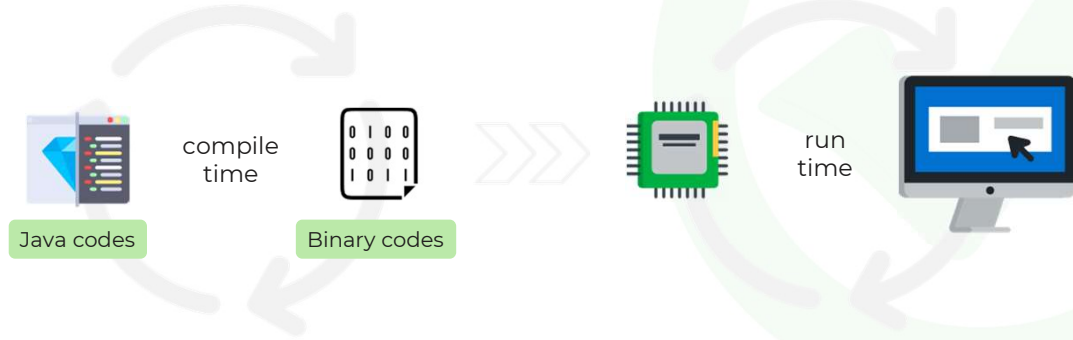
Yorumlayıcı Tabanlı

Derleyicide olduğu gibi tüm komutların bir kere de makine koduna çevirmek yerine tek-tek alınıp makine koduna çevrilip çalıştırılmasını sağlar.



Compiler vs Interpreter

Compiler(Derleyici)

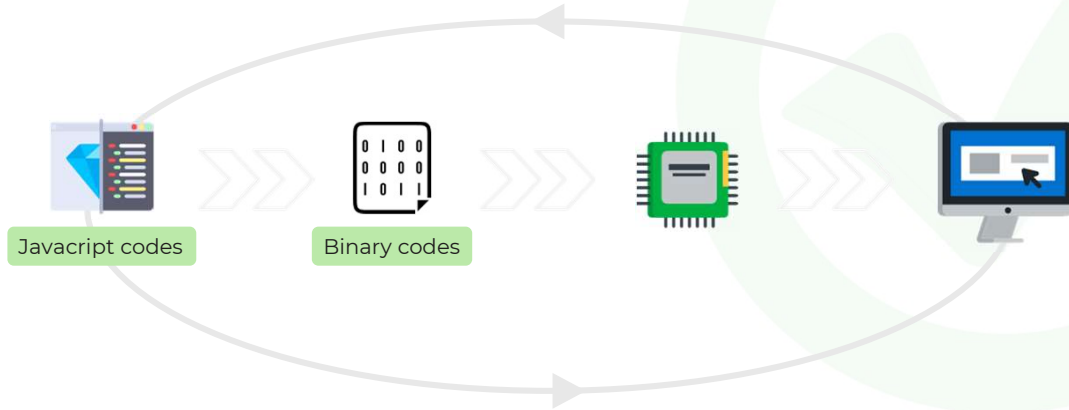


Program çalıştırıldığında, compiler, program kodlarının tamamını compile ederek makine diline (binary codes) çevirir. Bu aşamaya «compile time» denir. Tüm kodun derlenmesi sebebiyle programı ilk çalışması biraz zaman alırken, sonrasında performans üst düzeye çıkar.



Compiler vs Interpreter

Interpreter (Yorumlayıcı)



Javascript yorumlayıcısı, kodları satır satır alıp makine diline çevirir ve işlenmesini sağlar. Bu yapıda compile time veya run time yoktur. Kodların önce derlenmesi söz konusu olmadığı için, uygulamanın ilk çalışma hızı yüksektir.

Javascript yorumlayıcısı tarayıcıların içindeki bir motor yazılımdır. Her tarayıcıda bu farklı olabilir.

Chrome: Google V8 Engine

Firefox: SpiderMonkey

Safari: Javascript Core

Edge: Chakra



Nerede kullanılır?

HTML



JAVASCRIPT



CSS





Neden Javascript

Front-end Dev



React, Svelte, Vue, Angular gibi JS framework ve kütüphaneleri ile web ve mobil uygulamalar geliştirilebilmektedir.

Facebook, Netflix, Dropbox, AirBnb

Back-end Dev



NodeJs ile sunucu tarafında uygulamalar geliştirilebilmektedir.

Netflix, Uber, E-Bay



Neden Javascript

Mobile Dev



React Native, Ionic, Native Script gibi kütüphaneler ile mobile uygulama geliştirilebilir.

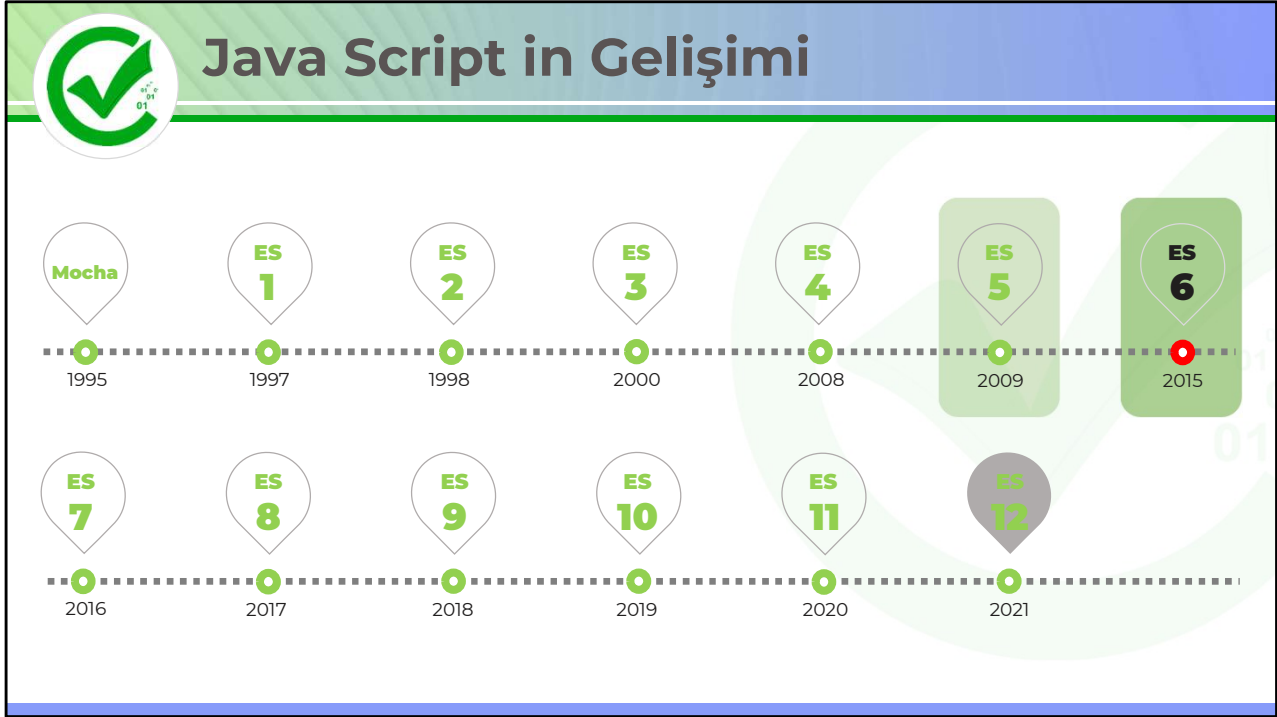
Facebook, Instagram, Skype, Uber, Pinterest

Desktop Dev



Electron kütüphanesi ile masaüstünde çalışacak uygulamalar geliştirilebilir.

VSCode, Whatsapp, Slack, Skype, Twitch, Teams



Tarayıcıların desteğinin en çok olduğu Ecma Script ES5 tir. Kodlama ona göre yapılırsa daha fazla tarayıcıda sorunsuz çalışacaktır.

<https://caniuse.com/?search=es5>

<https://caniuse.com/usage-table>

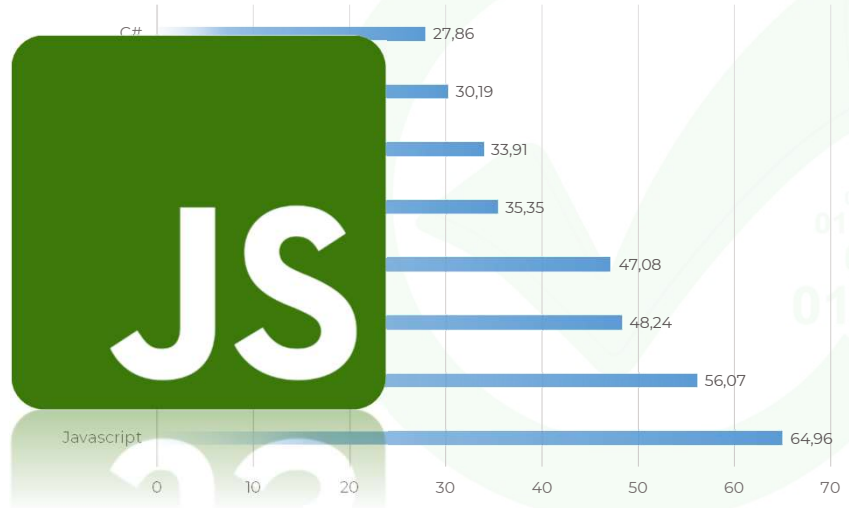
ES6/ES2015 ve sonrası :

- Modern tarayıcılar desteklemektedir. Eski tarayıcılar desteklemeyebilir.
- **Babel** gibi derleyiciler yardımıyla yüksek versiyonlar üretim aşamasında **ES5'e** çevrilerek eski tarayıcı uygunluğu sağlanabilir.

Yeni çıkan versiyonlar arttırımsal güncellemelerdir. Eski özellikler kaldırılmamıştır.



Popülerlik



<https://www.statista.com>

Web in en popüler programlama dillerinden biridir.

Web sitelerinin neredeyse tamamı Javascript kullanmaktadır.

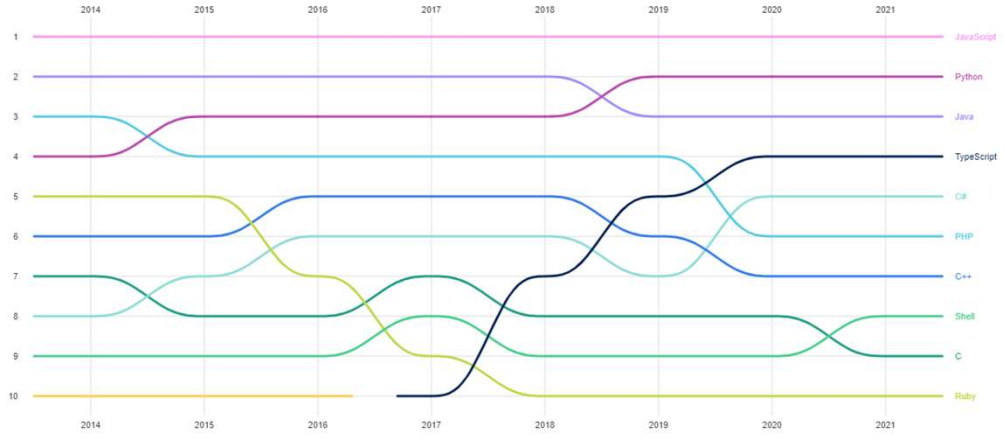
Önceleri client side için tercih edilse de şimdi nodejs kütüphanesi sayesinde backend programlamada da tercih edilmektedir.

<https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>



Popülerlik

2 milyar civarındaki internet sitesinde %95 oranında Javascript kullanılmaktadır.



<https://octoverse.github.com/#top-languages-over-the-years>



Neden popüler?



Kurulum
gerekmez



Object
Oriented



End to end
programming



Library &
Framework



Community

Kurulum gerekmez:

Frontend tarafında hiçbir kurulum gerektirmezken, backend kullanımı için nodejs kurulumu gereklidir. Onun kurulumu da çok basittir.

Object oriented: Nesne tabanlı programlamayı destekler.

End to end programming: Hem frontend hem backend hem mobile hem desktop çözümler üretilebilir.

Library & Framework: Çok geniş kütüphane ve framework desteği vardır.

Community: Çok geniş bir community ile internette en fazla bilgiye sahip programlama dilidir.



Case Types



camelCase



snake_case



kebab-case



Train-Case



PascalCase



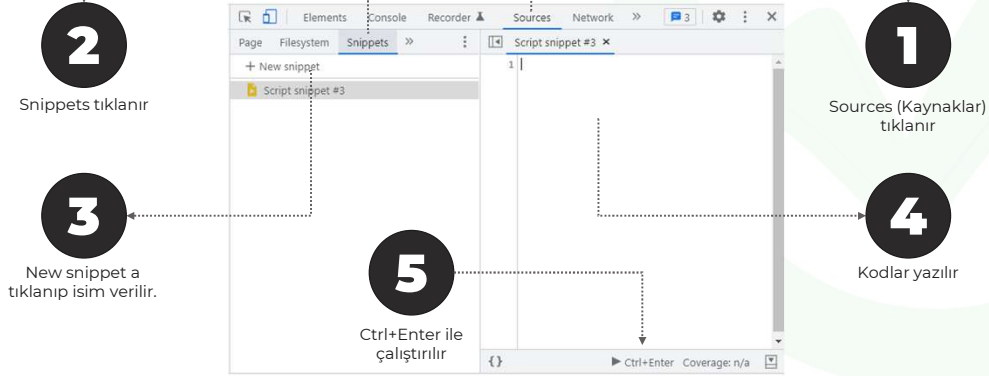
MACRO_CASE



İlk kodlar

Javascript in test amaçlı kullanımı

Boş bir chrome sayfasında sağ tıklayıp inspect (incele) dedikten sonra



- Snippets bölümünde her türlü JS komutları yazıp çalıştırılabilir. **Ancak**, Chrome geliştirici araçları bir yazılım geliştirmekten ziyade **test** ve **hata tespiti** gibi konular için daha uygundur.
- Bu yüzden, **JS** kodlarımızı çalıştırmak için dosya oluşturmalıyız.
- Gerekli oldukça zaten **Chrome** Geliştirme araçlarını kullanacağız.



ilk kodlar

```
<html>
  <head>

  </head>
  <body>

    <script>
      ...
    </script>
  </body>
</html>
```

- ◉ Javascript kodlarını yazmak için script tagi kullanılır.
- ◉ Tüm javascript kodları script tagları arasına yazılır
- ◉ Script taglarının body kapanmadan hemen önce konulması tavsiye edilir.
- ◉ Bu kullanıma internal script denir ve javascript kodları sadece mevcut sayfa için geçerli olur.



ilk kodlar

```
<html>
  <head>

  </head>
  <body>

    <script src="assets/js/script.js">
  </body>
</html>
```

- Javascript kodları harici bir dokümana konularak kullanılırsa buna external script denir
- Bu dosyayı kullanan tüm html sayfalarında kullanılabilir hale gelir.



Kodların alıřtırılması

1



Web sayfalarında bulunan Javascript kodları, tarayıcı tarafından alıřtırıldıđı iin html dosyasına dosyaya ift tıklayarak amak javascript dosyalarının alıřması iin yeterlidir.

2



VSCoDe Live Server eklentisi ile alıřtırılırsa, yapılan deđiřiklikler de anında yansıtılmıř olur. Ve geređe yakın bir ortamda geliřtirme yapılabilir.



JavaScript

Fundamentals

- › Syntax
- › Console
- › Alert, confirm, prompt
- › Comment
- › Variables and constants
- › Operators



Javascript syntax



- › Case sensitive bir dildir.
- › Satır sonlarına genellikle noktalı virgül (;) konulur.
- › Identifier lar(variables, constants, functions) harf, _ veya \$ ile başlayabilir.
- › Identifier lar **camel case** ile yazılır.

Case sensitive: Büyük küçük harf duyarlı.



Console

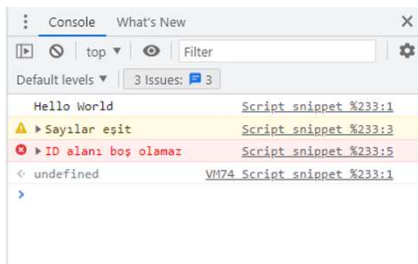
Javascript ile tarayıcıların console bölümüne birşeyler yazdırmak için **console** object kullanılır.

```
console.log("bla bla bla");
```



Console

Console object inin farklı kullanımları vardır.



```
console.log("Hello World");  
console.warn("Sayılar eşit");  
console.error("ID alanı boş olamaz");
```



Alert, Confirm, Prompt

Alert

Bilgilendirme yapmak için kullanılır.
`alert("Silmeye yetkiniz yok");`

Silmeye yetkiniz yok.

Tamam

Confirm

Soru sormak için kullanılır.
`confirm("Silmek istediği...");`

Silmek istediğinizden emin misiniz?

Tamam

İptal

Prompt

Bilgi almak için kullanılır.
`prompt("Adınızı giriniz");`

adınızı giriniz

Tamam

İptal



Comment

// Bu bir comment satırıdır.

/*

Bu şekilde birden çok satırı aynı
anda comment içine alabilirsiniz.

*/



Variables (Değişkenler), Constants (Sabitler)

CANADA	CAD	0.9512	0.8883
CHINA	CNY	7.3169	6.0910
EURO	EUR	0.6644	0.6100
JAPAN	JPY	109.00	102.00
SINGAPORE	SGD	1.3712	1.2630
HONG KONG	HKD	0.0043	0.4072
NEW ZEALAND	NZD	1.1646	1.0675
NEW ZEALAND	NZD	1.2536	1.2818
NEW ZEALAND	NZD	1.2536	1.2818
HONG KONG	HKD	0.0043	0.4072

► Programlama dillerinde ihtiyaç olduğu an ulaşılacak veri tutuculara **değişken** ya da **sabit** adı verilir.



Variables - Constants

Javascript te deęişken veya sabit tanımlamak için **3** farklı ifade kullanılır

var

ES
6
let

ES
6
const

Deęişken

sabit

var ve let, deęişken tanımlamak için, const ise sabit tanımlamak için kullanılır. var ile global, her yerden ulaşılabilen deęişkenler tanımlanabilir. Let ile ise block scoped deęişken tanımlanır. Sadece tanımlandığı yerde geçerlidir. Const ile sabit tanımlanır ve tanımlanırken deęer atanması zorunludur. Const da let gibi block scoped bir türdür.



Variables - Constants

```
var sayi;  
var Sayi;  
var genelOrtalama;  
const kdv = 1.18;
```

- › case sensitive yani büyük küçük harf duyarlıdır.
- › Sabitler tanımlanırken değerleri de verilmelidir.
- › Değişkenler camelCase olarak tanımlanır.
- › İçinde saklayacağı veriyi anımsatacak bir isim verilir.

Javascript te genellikle camel case tercih edilirken, class oluşturulurken pascal case kullanılır.

- camelCaseBuSekildeYazilirDeğişken isimlendirmek için kullanılan farklı standartlar
- PascalCaseBuSekildeYazilir
- snake_case_bu_sekilde_yazılır
- kebab-case-bu-sekilde-yazilir



Variables - Constants

KURALLAR

- Değişken isimlendirilirken hem harfler hem de sayılar kullanılabilir. Ancak sayılar başa gelmez. Örneğin **sayi1** doğru bir isimlendirmeyken **1sayi** doğru bir isimlendirme değildir.
- Değişken isimlendirilirken alt tire (_ ve \$) kullanılabilir. Ancak boşluk ve diğer özel karakterler (?,%,!,., + vb.) kullanılmaz. Örneğin **ev adresi** ya da **kimlik%no** gibi değişken isimleri kurallara aykırı olduğundan hataya neden olacaktır.
- Değişken isimlendirilirken özel kullanım için ayrılmış olan **if, for, true vb.** ifadeler kullanılmamalıdır.
- İngiliz alfabesinde bulunmayan karakterler (**ç,ğ,ı,ö,ş,ü**) kullanılamaz.



Variables - Constants

PRACTISE

✓ders

✗1not

✓gen_ort

✓\$maas

✗ilk sayi

✗son_şekil

✗gun%

✗harf-bir

✓dGunu