

# **MBP 214**

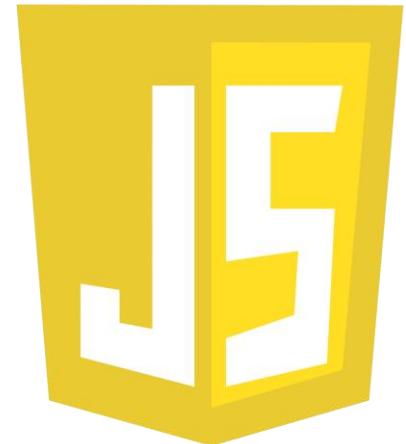
# **WEB PROGRAMLAMA**

---

Bahar 25-26

İEÜ MYO - Bilgisayar Programcılığı

**JavaScript**



# M. Volkan GÜNGÖR



Elektronik ve Haberleşme  
Mühendisi  
Haberleşme, Biyomedikal ve IoT  
(Nesnelerin İnterneti) Teknolojileri

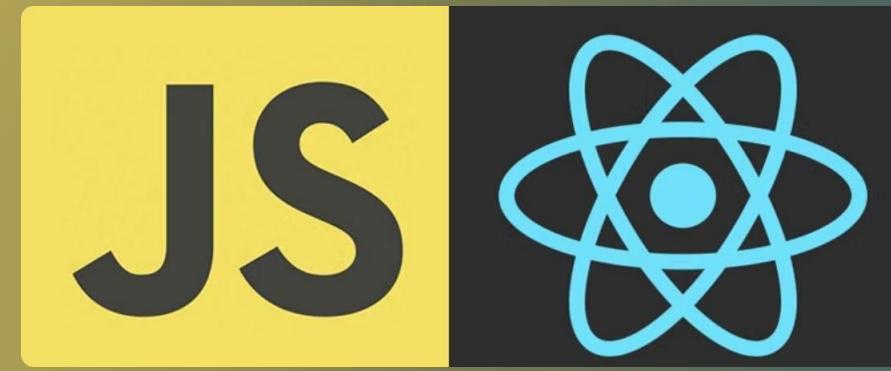


Yazılım Proje Yöneticisi ve  
Backend Developer



Yönetim Bilişim Sistemleri  
Doktora





# ilk ders ilk uygulama - döneme hızlı girelim

Estimate the Age  
of a Name

mustafa volkan 

mustafa volkan is 45 years old

**Kullanıcıdan isim alarak, "tahmin et" butonuna basılınlca", agify API ile sorgu yaparak Json formatında dönen yaş bilgisinin cevabını ekranı Türkçe yazdırın.**

**Sadece HTML, CSS ve Javascript kullanabilirsiniz.**

# Yarıyıl Aktiviteleri

## Ödev

**7 adet**

**%25**

Veriliş tarihinden itibaren 1 hafta sonra teslim

Github Account içinde bulunacak. - Kişisel Portfolyo

## Proje

**1 adet**

**%5**

**İlerleyen haftalarda  
Vize haftasına kadar  
konu belirlenmeli -  
karma uygulama  
sayfası - serbest seçim**

## Vize

**Karma soru tipi**

**%25**

## Final

**%45**

**Dönem içi ilerleyiş  
ile belirlenecek**

**LimitSizsiniz 😊**

# Final Proje Konuları önerileri

## Proje

- Seçeceğiniz bir API ile etkileşerek ondan aldığı veriyi ana sayfada görselleştiren bir uygulama. Json formatında data ile çalışacak.

## Final

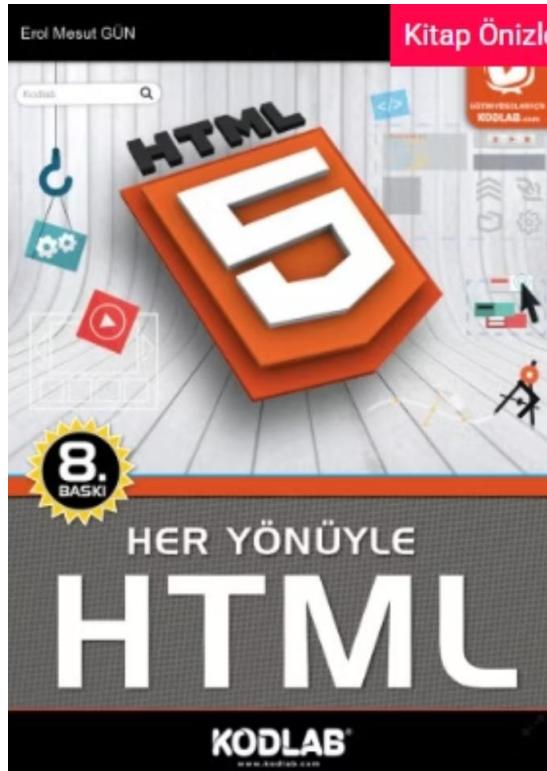
Dönem içi ilerleyiş ile belirlenecek.  
Yazılı veya Proje Teslimi.

Amaç, *github repo olması ve iş başvurusunda bir sorunu anlayıp analiz edip çözümü dijital olarak bulabildiğimizi göstermek.*

Proje teslimi olursa en az 3 sayfadan oluşacak.

- API ile etkileşerek veri görselleştirme,
- JSON formatında veri alacak ve veri görselleştirme,
- Oyunlaştırma,
- Eğlenceli sayfa animasyonu.
- Yönetim Paneli
- Yeni bir LLM etkileşim sayfası
- LimitSizsiniz ☺

# Ders Kaynakları



Erol Mesut Gün

HER YÖNÜYLE HTML EĞİTİM  
KİTABI

Kodlab Yayınları



İbrahim Çelikbilek

JAVASCRIPT PROGRAMLAMA  
EĞİTİM KİTABI

Kodlab Yayınları



Mehmet CANKER

REACT ve REACT NATIVE  
EĞİTİM KİTABI

Kodlab Yayınları



<https://tr.react.dev/>

[www.google.com](http://www.google.com)

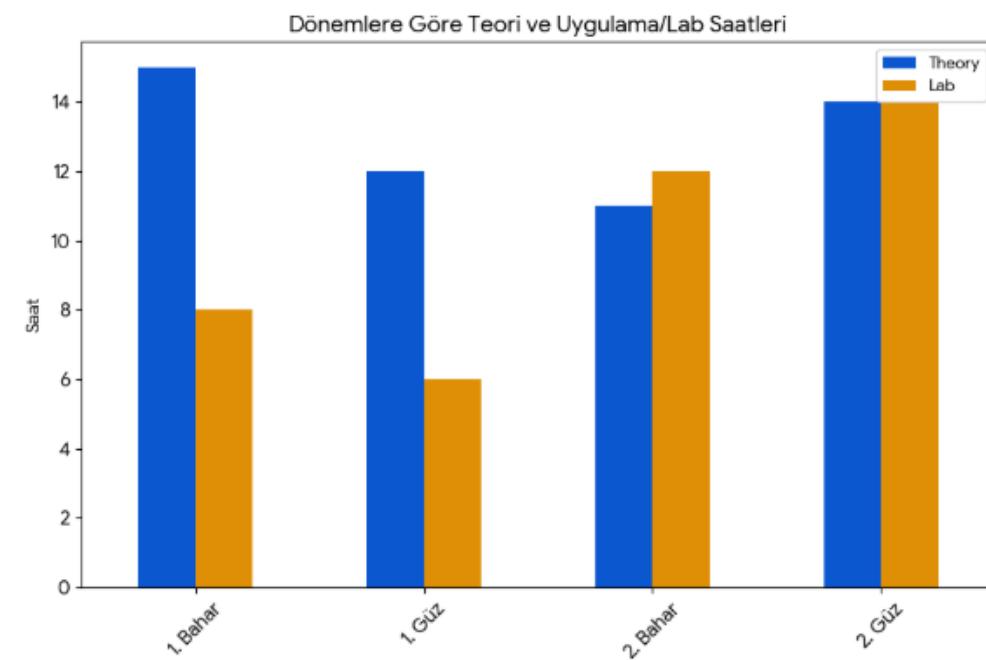
[www.w3school.com](http://www.w3school.com)

[www.youtube.com](http://www.youtube.com)

<https://web.dev/?hl=tr>

# Ders Akışı

- Teorik anlatım
- Teori uygulaması
- Sınıf Örneği / kendi örneğiniz
- Blok 1 (0-45 dk) – 45dk
  - mola (10 dk)
- Blok 2 (46-90 dk) – 45dk
  - mola (10 dk)
- Blok 3 (91-135 dk) – 45dk



# Geliştirme Ortamı – Sadece Notepad ve tarayıcı yeterli

- VS Code
- VS Code eklentileri:
  - Prettier,
  - ES7 React Snippets
- **Node.js** — React için gerekli
- **Tarayıcı:** Chrome, Edge, Firefox
- <https://codepen.io/>
- <https://jsfiddle.net/>

Hafta	Konular	Ön Hazırlık
1	HTML ve CSS temelleri	Gün, M. E., "Her Yı Kodlab (2017). Bölüm 1
2	Responsive tasarım ve Sablon kullanımı	Gün, M. E., "Her Yı Kodlab (2017). Bölüm 2
3	Web sayfası uygulaması	Gün, M. E., "Her Yı Kodlab (2017). Bölüm 3
4	Javascript temel giriş: derlevicilere dair açıklamalar, program yazmak için gerekenler, değişkenler, değişken türleri	Celikbilek, İ., "Java Programlama", Kodl Bolum 3, Bolum 4
5	Aritmatik operatörler, atama operatörleri, if-else koşul ifadeleri, mantıksal operatörler	Celikbilek, İ., "Java Programlama", Kodl Bolum 5, Bolum 6
6	Döngüler	Celikbilek, İ., "Java Programlama", Kodl Bolum 6
7	Diziler	Celikbilek, İ., "Java Programlama", Kodl Bolum 7
8	Ara Sınav	
9	Fonksiyonlar	Celikbilek, İ., "Java Programlama", Kodl Bolum 9
10	Javascript dom yapısı	Celikbilek, İ., "Java Programlama", Kodl Bolum 14
11	Javascript dom yapısı 2	Celikbilek, İ., "Java Programlama", Kodl Bolum 14
12	Javascript dom yapısı 3	Celikbilek, İ., "Java Programlama", Kodl Bolum 14
13	React Giriş ve JSX	Cankır M., "React Web ve Mobil Uygulamalar", Kodlab Yayınları (2018). Bölüm 4
14	React Komponentleri ve Yasam Döngüsü	Kodlab Yayınları (2018). Bölüm 5
15	Dönemin gözden geçirilmesi	
16	Final Sınavı	

# DEVAMSIZLIK: toplam **12** saat

Teorik dersi %70 devam zorunlu  
 Uygulama dersi %80 devam zorunlu

Toplam vize/final hariç 42 saat (14 + 28)

Devamsızlık kayıtları  
**haftalık** olarak sisteme işleniyor.

# Ders Akışı

## HTML ve Modern CSS Temelleri

• **Teori :** Semantik (Anlamsal) HTML5, CSS Kutu Modeli (Box Model), Modern Düzenleme (Layout) teknikleri.

• **Uygulama:**

- Kişisel bir portfolyo sayfası oluşturma.
- Sayfa düzeni için Flexbox kullanımı.
- CSS Değişkenlerine (Variables) giriş.
- **Proje:** Duyarlı (Responsive) profil kartı.

## Kontrol Akışı - Koşullar ve Mantık

• **Teori :** if/else, switch, mantıksal operatörler, true/false kavramları.

• **Uygulama:**

- Taş-Kağıt-Makas oyunu.
- Form doğrulayıcı (Validation).
- **Proje:** Etkileşimli hikaye oyunu (Kendi maceranı seç).

## İleri Seviye CSS ve Duyarlı (Responsive) Tasarım

• **Teori:** CSS Grid, Medya Soruları (Media Queries), "Önce Mobil" (Mobile-first) yaklaşımı.

• **Uygulama:**

- Duyarlı bir fotoğraf galerisi oluşturma.
- Cihaza göre uyum sağlayan navigasyon menüsü yapımı.
- **Proje:** Tüm cihazlarda çalışan bir "Landing Page" (Açılış Sayfası).

## Döngüler ve Tekrarlar

• **Teori:** for, while, do-while, for...of, for...in, break/continue.

• **Uygulama:**

- Desen oluşturucular (yıldızlar, piramitler).
- Sayı tahmin oyunu.
- Çarpım tablosu oluşturucu.
- **Proje:** Yapılacaklar listesi (Sadece dizi tabanlı, henüz DOM yok).

## JavaScript Temelleri - Değişkenler, Veri Tipleri ve Operatörler

• **Teori:** Değişkenler (let, const, var), Veri tipleri, Tip dönüşümü (Type conversion).

• **Uygulama:**

- Etkileşimli hesap makinesi.
- Sıcaklık dönüştürücü (Celsius/Fahrenheit).
- Tarih işlemlerini içeren yaşı hesaplayıcı.
- **Proje:** Basit bir bütçe hesaplayıcı.

## Diziler ve Dizi Metotları

• **Teori:** Dizi metotları (map, filter, reduce, forEach, find, some, every).

• **Uygulama:**

- Alışveriş sepeti mantığı.
- Öğrenci not analizörü.
- Veri filtreleme ve sıralama.
- **Proje:** Arama ve filtreleme özellikli film kütüphanesi.

# Planlanan Haftalık Ders Akışı

## Fonksiyonlar ve Kapsam (Scope)

• **Teori:** Fonksiyon bildirimleri, ifadeler, arrow (ok) fonksiyonları, parametreler, return, kapsam (scope).

• **Uygulama:**

- Yardımcı fonksiyonlar kütüphanesi oluşturma.
- Özyinelemeli (Recursive) fonksiyonlar (faktöriyel, fibonacci).
- Yüksek mertebeden fonksiyonlar (Higher-order functions).
- **Proje:** Yeniden kullanılabilir fonksiyonlarla basit bir oyun.

## DOM Olayları ve Etkileşim

• **Teori:** Olay dinleyiciler (Event listeners), event objesi, olay delegasyonu (event delegation), preventDefault.

• **Uygulama:**

- Tıklama sayacı.
- Klavye olayları ile çalışan oyun.
- Form gönderimi yönetimi.
- **Proje:** "Köstebek Vurmaca" (Whack-a-mole) veya hafıza kartı oyunu.

## 8. HAFTA: ARA SINAV

Sınav: Teori + kodlama

## DOM Manipülasyonuna Giriş

• **Teori:** DOM nedir?, querySelector, getElementById, createElement.

• **Uygulama:**

- Dinamik içerik güncelleştirici.
- Element stili değiştirici.
- **Proje:** Etkileşimli tebrik kartı oluşturucu.

## Gelişmiş DOM ve Yerel Depolama (Local Storage)

• **Teori:** Local Storage, JSON, classList, veri (data) öznitelikleri.

• **Uygulama:**

- Tema değiştirici (koyu/açık mod).
- Kalıcı (sayfa yenilense de gitmeyen) yapılacaklar listesi.
- **Proje:** Kategorilere sahip not defteri uygulaması.

## React'e Giriş - Bileşenler ve JSX

• **Teori:** React temelleri, Vite/create-react-app, JSX, Bileşenler (Components), Props, React Durum(State) ve Hook

• **Uygulama:**

- İlk React uygulaması kurulumu.
- Yeniden kullanılabilir bileşenler oluşturma.
- Props aktarımı egzersizleri.
- **Proje:** Bileşen kütüphanesi (Buton, Kart, Başlık bileşenleri).

# Ders Sözleşmemiz



## Anlamamak serbesttir

Herkesin anlamama ve anlamak için soru sorma hakkı vardır.

Burada "anlamadım" demek tamamen normaldir.



## Hata = öğrenme

Hata yapmak öğrenmenin doğal parçasıdır.

Hatalarla dalga geçilmez, birlikte öğrenme fırsatına çevrilir.



## Soru sormak öğrenmektir

Ders sırasında konu ile ilgili her türlü soru serbesttir.

Özel durumlar dışında - erteleme talebi değerlendirilmez.



## Söz kesmek yok

Fikirler özgür, iletişim saygıdır.



## Anlatabiliyorsan senindir

Yazdığın kodu ve yaptığı çalışmalarını anlatabiliyor, sorulara cevap verebiliyorsan o çalışma senindir.

Değerlendirme de buna göre yapılır.



## Yarış değil, ortak yol

Burası yarış alanı değil, ortak öğrenme alanıdır.

Bilgi paylaşılır; üstünlük kurmak değil birlikte ilerlemek hedeflenir.



## Herkes öğrenir

Eğitmen de öğrenme sürecinin içindedir. Bilinmeyen bir konu varsa açıkça söyleyenir, birlikte araştırılır.



## Kiyas yok, ilerleme var

Başkasının ödevi veya çalışması senin için ölçüt değildir.

Herkes kendi sürecinden sorumludur.



## Zamana saygı

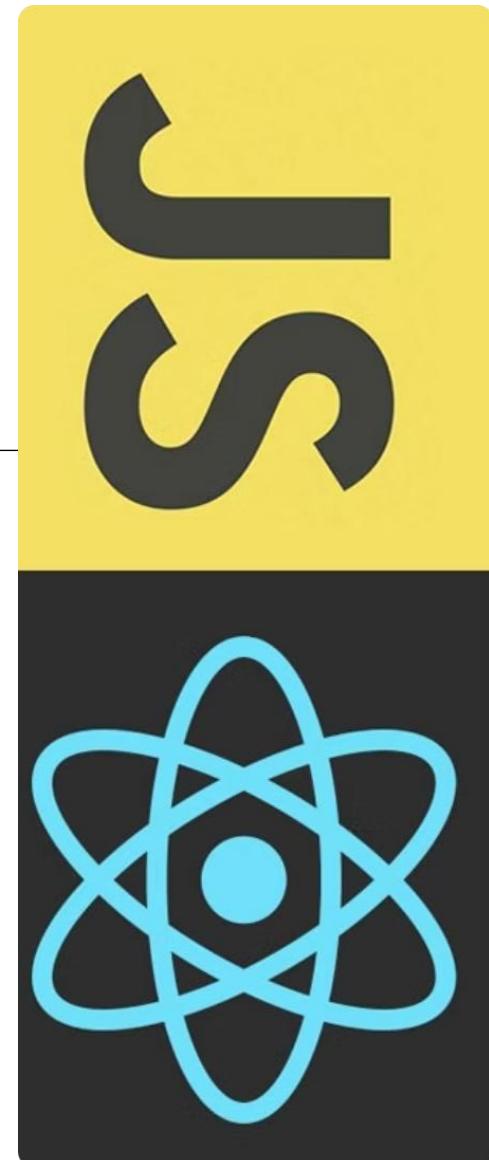
Zaman herkes için değerlidir.

Ders saatlerine, teslim tarihlerine ve ortak kurallara özen göstermek hepimizin sorumluluğudur

# HABERLEŞME:

[mvolkang@gmail.com](mailto:mvolkang@gmail.com)

OBS mesajları



mvolkang@gmail.com

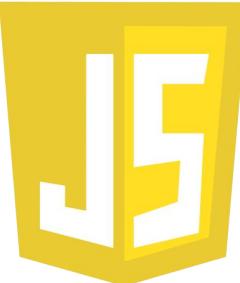
[MBP-214 / Şube] konu....

**Konu kısmında kesinlikle [MBP-214 / şube] bilgisi  
bulunmalı**

İyi günler

İsim/Soyad

Okul numarası



# MBP214'e Kadar Neler Öğrendik?

- **Programlama Temelleri & Nesne Tabanlı Programlama:** Algoritma kurma ve OOP prensipleri (MBP 193, MBP 106, MBP 205).
- **Veri Tabanı Yönetimi:** SQL ve veri modelleme teknikleri (MBP 101, MBP 108).
- **Arayüz Tasarımı:** HTML ve CSS ile statik web sayfaları oluşturma (MBP 191).
- **Görsel & Internet Programcılığı:** Form yapıları ve temel sunucu taraflı mantık (MBP 104, MBP 192).
- **Sistem Analizi ve Tasarımı:** Yapacağımız projeler neden yapılır? Neyi çözer, bunu nasıl yaparız? (MBP 206)

# MBP214 Web Programlama: Bu Derste Neler Öğreneceğiz?

- **Dinamik Web Uygulamaları:** Sadece görünen değil, arka planda çalışan ve veri işleyen siteler kurabileceğiz.
- **Yönetim Panelleri:** İçeriklerin kullanıcı tarafından güncellenebildiği sistemler geliştirebileceğiz.
- **Uçtan Uca Proje Geliştirme:** Front-end ve Back-end süreçlerini birleştirerek tam kapsamlı projeler üretebileceğiz.
- **Veri Görselleştirme** ve Farklı uygulama kütüphaneleri

# Adım Adım JS Odaklı Projeler

Okul hayatınızda veya özel hayatınızda öğrendiğiniz her şeyi bu derste birleştirebilir veya bu derse aktarabilirsiniz.

Temelde JS/React kullanarak çalışacağız.

# Sadece HTML, CSS ve JS ile Neler Yapılabilir?

- **Dinamik Web Siteleri ve Landing Page'ler:** Şirketler için modern, hızlı ve mobil uyumlu (Responsive) tanıtım sayfaları.
- **Web Tabanlı Oyunlar:** JavaScript'in Canvas yapısını kullanarak 2D platform oyunları veya bulmaca oyunları.
- **Tarayıcı Eklentileri:** Chrome veya Firefox için hayatı kolaylaştıran küçük araçlar (Örn: AdBlock benzeri araçlar veya not alma eklentileri).
- **Web Uygulamaları (SPA):** To-do listeleri, hava durumu uygulamaları, döviz çeviriciler veya hesap makineleri.
- **Etkileşimli Portfolyolar:** Animasyonlu, kullanıcının hareketlerine tepki veren profesyonel CV siteleri.

# Bu Çalışma Hangi Yetenekleri Kazandırır?

- **Görsel Mantık ve Tasarım Gözü:** CSS ile uğraşmak, bir fikri görsel bir şablon'a dökme ve kullanıcı deneyimini (UX) önceliklendirme yetisini geliştirir.
- **Problem Çözme ve Algoritma:** JavaScript, öğrencinin bir mantık hatasını (bug) bulma ve adım adım çözüm üretme becerisini artırır.
- **Modern Framework'lere Hazırlık:** Bugün piyasada çok popüler olan React, Vue veya Angular gibi teknolojileri öğrenmenin tek yolu, saf JavaScript'i çok iyi bilmektir. Temeli sağlam olan öğrenci, yeni teknolojileri çok daha hızlı öğrenir.

# Web Geliştirmede Problem Çözme

Web geliştirme sadece kod yazmak değildir. **Bir ihtiyacı anlamak ve ona çalışan bir çözüm üretmektir.** Yapılan her proje, karşılaşılan bir problemi çözmeyi amaçlar.

- Etkileşimli, Kullanışlı
- Kullanıcıyı memnun eden
- Fikri çalışan bir ürüne dönüştüren web siteleri ve uygulamalar geliştirmektir.

**Web geliştirme = Problem + Kod + Çözüm**

Bir web geliştirici olarak,

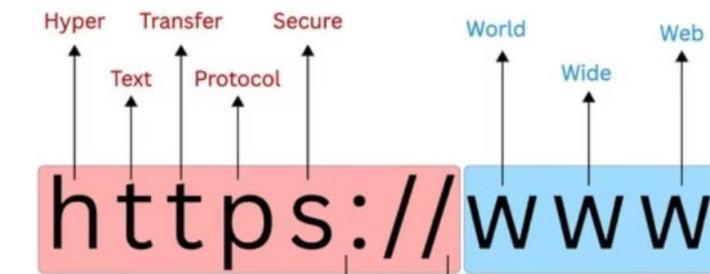
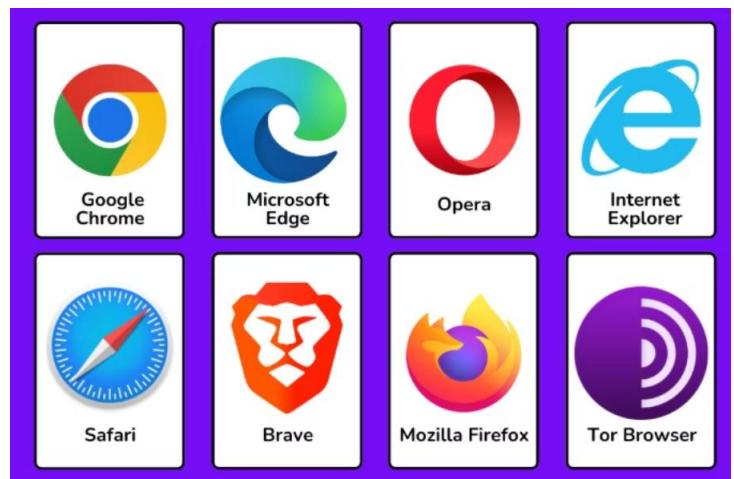
- her projede yeni şeyler öğrenir ve farklı problemlerle karşılaşırınsız
- sadece kod yazan değil, **çözüm üreten bir programcı** olursunuz.

**Bugün YZ hepimize yeni fırsatları ile geliyor. YZ ile meslekler arasındaki farklar hem kapanıyor hem açılıyor. Bu değişimin yönünü belirlemek sizin elinizde.**

**Bu dersin ana amacı sizin yeni bir dilde daha sorun çözebilmenizi sağlamak.**



# Bunlar ne ola ki?



# Web!

# Tarayıcılar! internet!

```
Domain Name: ISC.COM
Updated Date: 2016-06-05T07:03:37Z
Creation Date: 1986-08-05T04:00:00Z
Admin Name: Koosis, Donald
Admin Organization: ISC Consultants Inc.
Admin Street: PO Box 1379
Admin City: Woodstock
Admin State/Province: NY
Admin Postal Code: 12498
Admin Country: US
Admin Phone: +1.2124778800
Admin Email: donald.koosis@ISC.COM
```

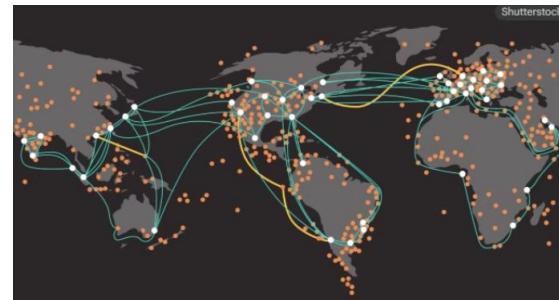
# Temel Tanımlar



## Tarayıcılar!

Tarayıcı, WWW üzerindeki içeriklere erişmemizi, bu içerikleri görüntülememizi ve onlarla etkileşime girmemizi sağlayan bir yazılımdır.

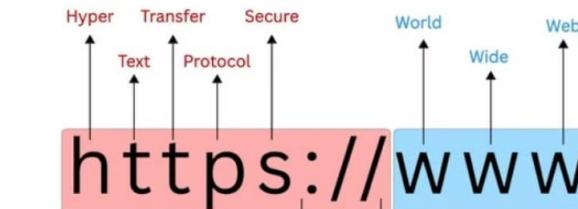
- **Çevirmen Rolü:** Sunucudan gelen karmaşık kodları (HTML, CSS, JavaScript) bizim anlayabileceğimiz görsel bir arayüze (resimler, butonlar, metinler) dönüştürür.
- **Popüler Örnekler:** Google Chrome, Safari, Microsoft Edge, Mozilla Firefox.
- **Temel İşlevi:** URL'yi IP adresine dönüştürmek için istek gönderir ve gelen yanıtı ekranımıza çizer (rendering).



## internet!

Internet, dünya üzerindeki bilgisayar ağlarını birbirine bağlayan **devasa bir fiziksel ağ altyapısıdır** ve üstünde çalışan protokoller bütündür.

- **Kapsamı:** Fiber optik kablolar, uydular, routerlar ve veri merkezlerinden oluşur.
- **Görevi:** Bilginin (verinin) bir noktadan diğerine taşınmasını sağlar.
- **Örnek:** Sadece web sitelerini değil; e-postaları, WhatsApp mesajlarını, online oyun verilerini ve dosya transferlerini (FTP) taşıyan ana "yoldur".



## Web!

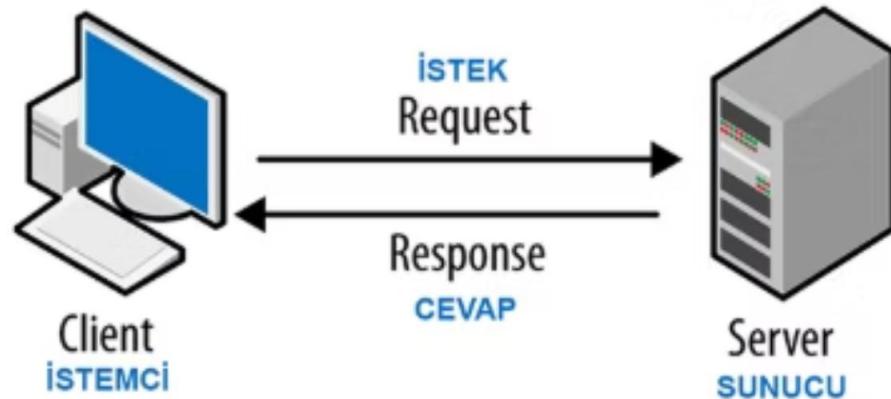
WWW, Internetin altyapısı üzerinde çalışan bir servistir.

HTTP/HTTPS protokolü üzerinden erişilen dokümanlar sistemi.

**Mantığı:** Birbirine "hyperlink" (bağlantı) dediğimiz köprülerle bağlı olan dokümanlar (web sayfaları) bütündür.

**Tarihçesi:** Tim Berners-Lee tarafından 1989'da CERN'de geliştirilmiştir.

# İstemci-Sunucu (Client-Server) Modeli

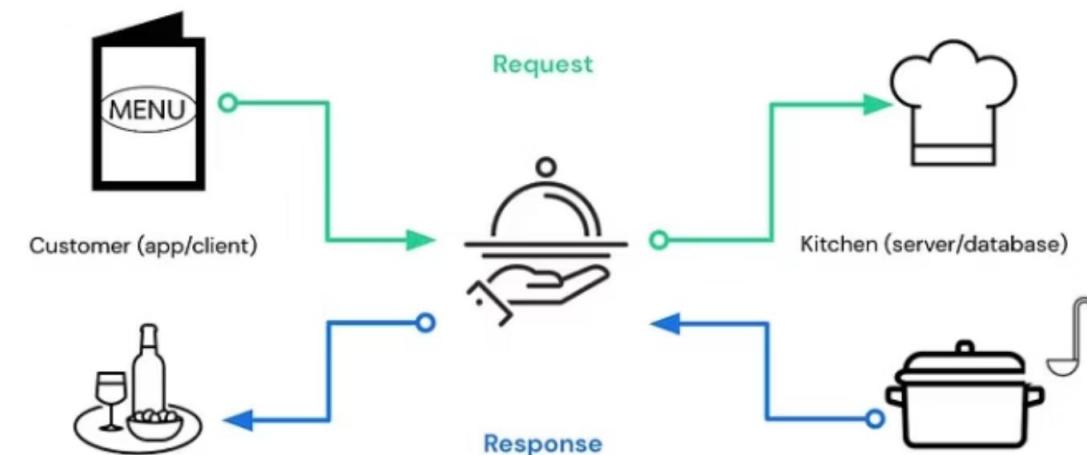


## Sunucu (Server)

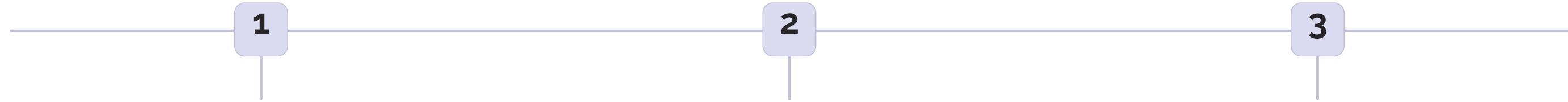
Hizmet sunan, kaynakları (veri, dosya, veritabanı) barındıran taraftır. İstemciden gelen talepleri dinler, işler ve sonucu geri gönderir.

## İstemci (Client)

Hizmet talep eden taraftır. Web bağlamında bu genellikle kullanıcının tarayıcısıdır (Chrome, Firefox vb.). Kullanıcı arayüzü sunar ve kullanıcı etkileşimlerini yakalar.



# HTTP İstek-Yanıt (Request-Response) Döngüsü



## İstek (The Request)

Tarayıcı, sunucuya bir paket gönderir.  
Bu paketin içinde şunlar yer alır:

- Metot (Method):** Ne yapmak istiyoruz?
  - GET: Veri getir (Sayfayı görüntüle).
  - POST: Veri gönder (Form doldur, kayıt ol).
- URL (Path):** Hangi kaynağa erişmek istiyoruz? (/anasayfa veya /api/kullanici)
- Header (Üstbilgi):** İstemci hakkında ek bilgiler (Hangi tarayıcı kullanılıyor, dil tercihi nedir?).

## İşleme (Processing)

Sunucu isteği alır, gerekiyorsa veritabanına gider, dosyaları okur ve bir sonuç hazırlar.

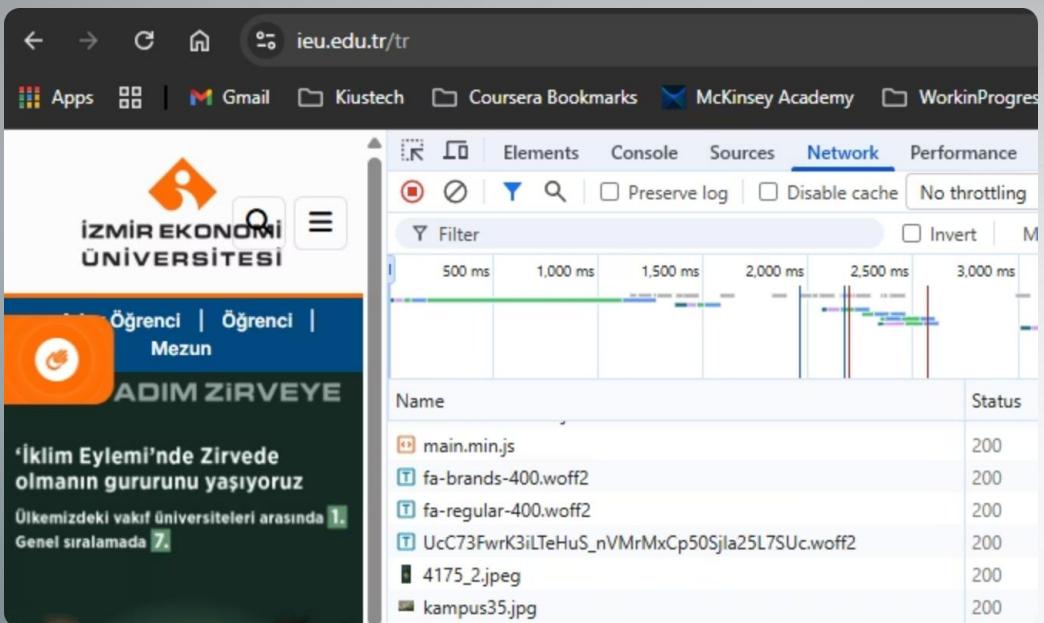
## Yanıt (The Response)

Web Sunucusu hazırladığı paketi istemciye geri gönderir. Bu paketin en önemli parçaları:

- Durum Kodu (Status Code):** İşlem başarılı mı?
  - 200 OK: Her şey yolunda.
  - 404 Not Found: Aradığın sayfa yok.
  - 500 Internal Server Error: Sunucuda işler karıştı.
- Body (Gövde):** İstenilen asıl içerik. Genellikle HTML, JSON veya bir görsel dosyasıdır.

# Adres Çubuğu Arkasında Neler Oluyor?

Bir URL (örneğin: <https://www.ieu.edu.tr/>) yazıldığında gerçekleşen mikro saniyelik olaylar



- **DNS Çözümleme:** Tarayıcı, "[www.ieu.edu.tr](https://www.ieu.edu.tr)" isminin hangi IP adresine (örneğin: 192.0.2.1) karşılık geldiğini öğrenmek için telefon rehberine (DNS) bakar.
- **TCP/IP Bağlantısı:** Sunucu ile güvenli bir hat kurulur
- **HTTP İstekleri:** Tarayıcı "Bana ana sayfayı gönder" der.
- **HTTP Cevapları:** Sunucu "Ana sayfayı ..... ile gönderildi" ile dosyaları gönderir.
- **Tarayıcı Render Süreci:** Sunucudan gelen HTML/CSS/JS dosyaları tarayıcı tarafından yorumlanarak görsel bir arayüze dönüştürülür.

# Web Teknolojilerinin Rol Dağılımı



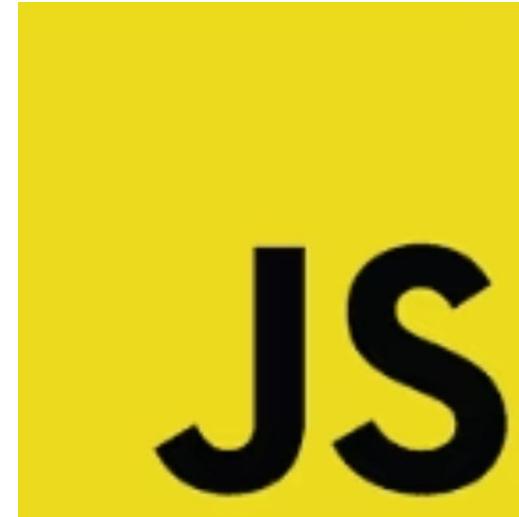
## HTML (İskelet)

Yapı ve içerik (Burada bir başlık var, burada bir buton var).



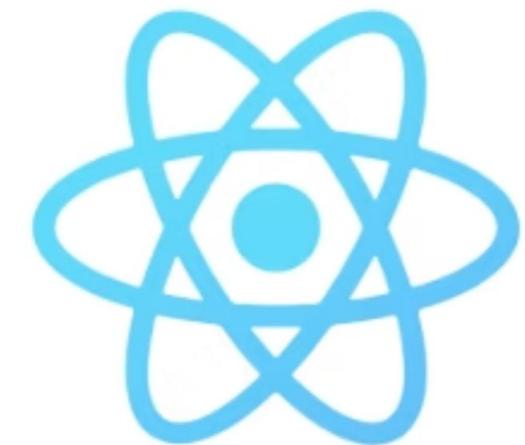
## CSS (Dış Cephe/Makyaj)

Stil ve düzen (Buton kırmızı olsun, başlık sağa yaslaşın).



## JavaScript (Sinir Sistemi)

Etkileşim ve fonksiyon (Butona basınca ne olacak?).



## React

## React (Modern İnşaat Teknikleri)

Bu yapının parçalar (bileşenler) halinde, hızlı ve tekrar kullanılabilir şekilde nasıl inşa edileceği.

# Tarayıcı Anatomisi ve Rendering Engine

Tarayıcı sadece bir pencere değil, karmaşık bir mühendislik harikasıdır.

Web Tarayıcıları (Browser); temel olarak HTML dili ile oluşturulan içerikleri açmak için kullanılan yazılımlardır.

## Rendering Engine (Görüntüleme Motoru):

Sunucudan gelen HTML ve CSS kodlarını alıp ekrandaki piksellere dönüştüren motor sistemidir (Örn: Chrome için Blink, Safari için WebKit).

## JavaScript Engine (JS Motoru):

JavaScript kodlarını makine diline çevirip çalıştırır (Örn: Chrome'da V8, Firefox'ta SpiderMonkey).

JS yorumlayıcıları, kullandıkları bellek miktarı ya da diğer özellikleri ile tarayıcı hızını etkileyen en önemli faktördür.

## Sayfa Nasıl Çizilir?

- 1 DOM Ağacı: HTML etiketleri hiyerarşik bir yapıya dönüştürülür.
- 2 CSSOM: CSS kuralları bu ağaca giydirilir.
- 3 Render Tree: Hangi elemanın nerede duracağı hesaplanır (Layout).
- 4 Painting: Her şey ekrana boyanır.