



CodeCraft

Bootcamp^{'25}

Javascript

Introduction to DOM, Selecting & Changing
Elements

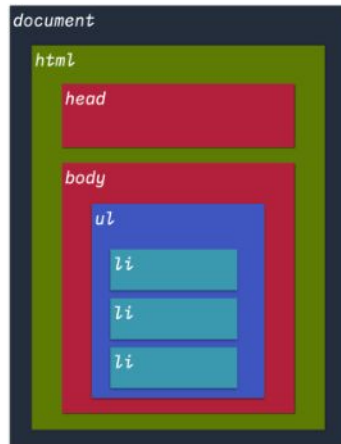
What is Document Object?

- Document Object, JavaScript'in web sayfası ile iletişim kurmasını sağlar.
- Sayfa içeriğini okumamıza, değiştirmemize ve olayları yönetmemize izin verir.
- document nesnesi, web sayfasının her bileşeniyle etkileşime geçmek için kullanılan ana kapıdır.

```
<html lang="en">
<head>...</head>
<body>
  ...
  <ul>
    <li>💃 dance</li>
    <li>🥳 cheer</li>
    <li>🍷 eat</li>
  </ul>
</body>
</html>
```

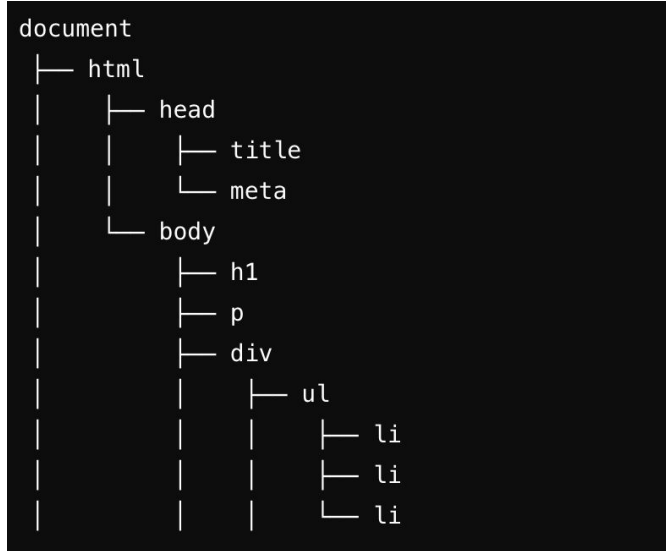
Pick your party style!

- 💃 dance
- 🥳 cheer
- 🍷 eat



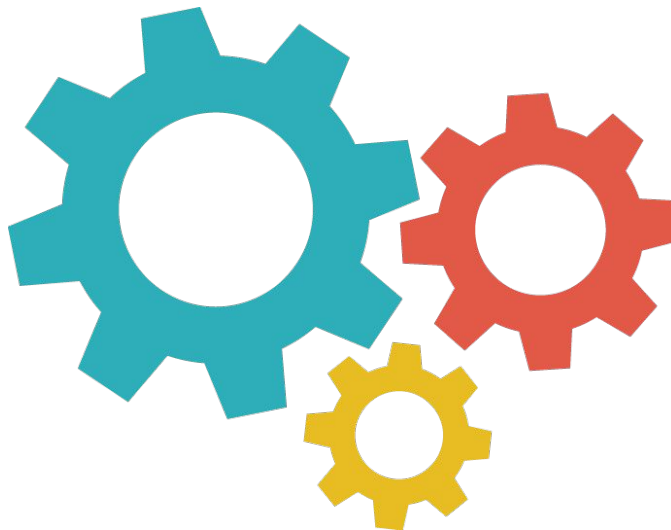
DOM

DOM (**Document Object Model**), bir web sayfasının **tarayıcı tarafından ağaç yapısında temsil edilen versiyonudur**. JavaScript ile bu yapıya erişerek HTML içeriğini **okuyabilir, değiştirebilir ve silebiliriz**.



How do browsers interpret HTML?

- Tarayıcı HTML dosyasını satır satır okur (Parsing).
- HTML öğeleri DOM ağacına dönüştürülür.
- CSS dosyaları yüklenir ve CSSOM oluşturulur.
- DOM + CSSOM birleşerek sayfa ekrana çizilir (Render Tree & Paint işlemi).



DOM Structure

Tarayıcı, HTML belgesini hiyerarşik bir ağaç yapısına dönüştürerek her ögeyi bir düğüm (node) olarak temsil eder.

- **Ebeveyn (Parent):** Bir elementi içeren üst düğümdür
- **Çocuk (Child):** Bir elementin içinde bulunan düğümlerdir
- **Kardeş (Sibling):** Aynı ebeveynin altındaki öğelerdir

```
<ul>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

Node Types

- **ELEMENT_NODE** → ?
- **ATTRIBUTE_NODE (Eski)** → HTML özniteliklerini temsil eder (Artık `getAttribute()` kullanılıyor).
- **TEXT_NODE** → ?
- **CDATA_SECTION_NODE** → Eski XML belgelerinde özel karakterleri saklamak için kullanılır.
- **ENTITY_REFERENCE_NODE (Eski)** → XML'de özel karakter referansları için kullanılır.
- **ENTITY_NODE (Eski)** → XML'de varlıkları temsil eder.
- **PROCESSING_INSTRUCTION_NODE** → XML işleme talimatlarını içerir.
- **COMMENT_NODE** → ?
- **DOCUMENT_NODE** → ?
- **DOCUMENT_TYPE_NODE** → Belge türünü tanımlar (`<!DOCTYPE html>` gibi).
- **DOCUMENT_FRAGMENT_NODE** → Hafif ve performanslı DOM manipülasyonu için kullanılır.
- **NOTATION_NODE (Eski)** → XML notasyonlarını temsil eder.

Selecting Elements

- `document.getElementById("id")`
- `document.getElementsByClassName("class")`
- `document.getElementsByTagName("tag")`
- `document.querySelector("cssSelector")`
- `document.querySelectorAll("cssSelector")`



Changing Elements

İçerik Değiştirme

- `.innerText` → Sadece görünen metni değiştirir.
- `.textContent` → Görünmeyen metinleri de kapsar.
- `.innerHTML` → HTML içeriğini değiştirme.

Stil Değiştirme

- `element.style.color = "red";` → Color değişir
- `classList.add(), classList.remove(), classList.toggle()` → Class List değişir

Adding Html Elements to Dom

Yöntem	Ne Yapar?	Eski Alternatifi
append()	En sona ekler	appendChild()
prepend()	En başa ekler	insertBefore(..., parent.firstChild)
before()	Elementin önüne ekler	insertBefore()
after()	Elementin sonrasına ekler	insertAdjacentElement("afterend", ...)

Geri Sayım Aracı

HTML ile Sayı Girişi ve Butonlar

- Kullanıcı geri sayım süresini belirleyebilsin (örn. 10 saniye gibi). (prompt ile alalım veya yapabiliyorsak input elemanlarını kullanalım.)
- Bir 'Başlat' ve bir 'Sıfırla' butonu olsun.
- Geri sayım değeri bir <p> etiketi içinde gösterilsin.

JavaScript ile Geri Sayım Mantığını Oluşturma

- Başlat butonuna basılınca zaman her saniye bir azalsın.
- Süre 0 olduğunda geri sayımı durduralım ve 'Süre doldu!' yazdıralım.
- 'Sıfırla' butonuna basınca geri sayımı durdur ve sıfırla.