



# JavaScript Introduction

---

JavaScript Web'in programlama dilidir.

# JavaScript Variables

JavaScript'te bir değişkeni bildirmenin üç yolu vardır: **var**, **let** veya **const** kullanarak.

## VAR

Değişkenleri bildirmenin en eski yoludur. Fonksiyon kapsamına sahiptir ve yeniden atanabilir.

## LET

ES6'da tanıtılan let, blok kapsamlıdır ve yeniden atanabilir. Blok kapsamı davranışı nedeniyle genellikle var'a tercih edilir.

## CONST

ES6'da da tanıtılan const, blok kapsamlıdır ve yeniden atanamaz. Ancak, atanan değer bir nesne veya dizi ise, bu nesne veya dizinin özellikleri değiştirilebilir.

İsimler harf, rakam, alt çizgi ve dolar işareti içerebilir.

İsimler bir harfle başlamalıdır.

İsimler büyük/küçük harfe duyarlıdır (y ve Y farklı değişkenlerdir).

Ayrılmış sözcükler ad olarak kullanılamaz.



# JavaScript Operators

Javascript operatörleri farklı türde matematiksel ve mantıksal hesaplamalar yapmak için kullanılır.

## Arithmetic Operators

| Operator | Description       |
|----------|-------------------|
| +        | Toplama           |
| -        | Çıkarma           |
| *        | Çarpma            |
| **       | Üs                |
| /        | Bölme             |
| %        | Mod (Kalan Bulma) |
| ++       | Artırma           |
| --       | Azaltma           |

## Assignment Operators

| Object | Example | Same As          |
|--------|---------|------------------|
| =      | x = y   | x = y            |
| +=     | x += y  | x = x + y        |
| -=     | x -= y  | x = x - y        |
| *=     | x *= y  | x = x * y        |
| /=     | x /= y  | x = x / y        |
| %=     | x %= y  | x = x % y        |
| **=    | x **= y | x = x ** y       |
| ??=    | x ??= y | x = x ?? (x = y) |

# JavaScript Data Types

JavaScript'teki veri tipleri, temel veri tipleri ve nesne veri tipleri olarak iki ana kategoriye ayrılır.

| Operator  | Description   | Primitive Value |
|-----------|---|-----------------|
| String    | Metin verilerini temsil eder.   | Yes             |
| Number    | Hem tamsayıları hem de kayan noktalı sayıları temsil eder.  | Yes             |
| Bigint    | Çok büyük tamsayıları temsil etmek için kullanılır. n harfi ile sona eren tamsayılar olarak tanımlanır. | Yes             |
| Boolean   | Mantıksal değerleri temsil eder: true veya false.   | Yes             |
| Undefined | Bir değişken tanımlandığında ancak değer atanmadığında, değeri undefined olur.                          | Yes             |
| Null      | Bilinçli olarak hiçbir değeri belirtmeyen özel bir değer.   | Yes             |
| Symbol    | ES6 ile tanıtılan ve benzersiz ve değiştirilemez olan değerleri temsil eden bir veri tipi.              | Yes             |
| Object    |   | No              |

# JavaScript Functions

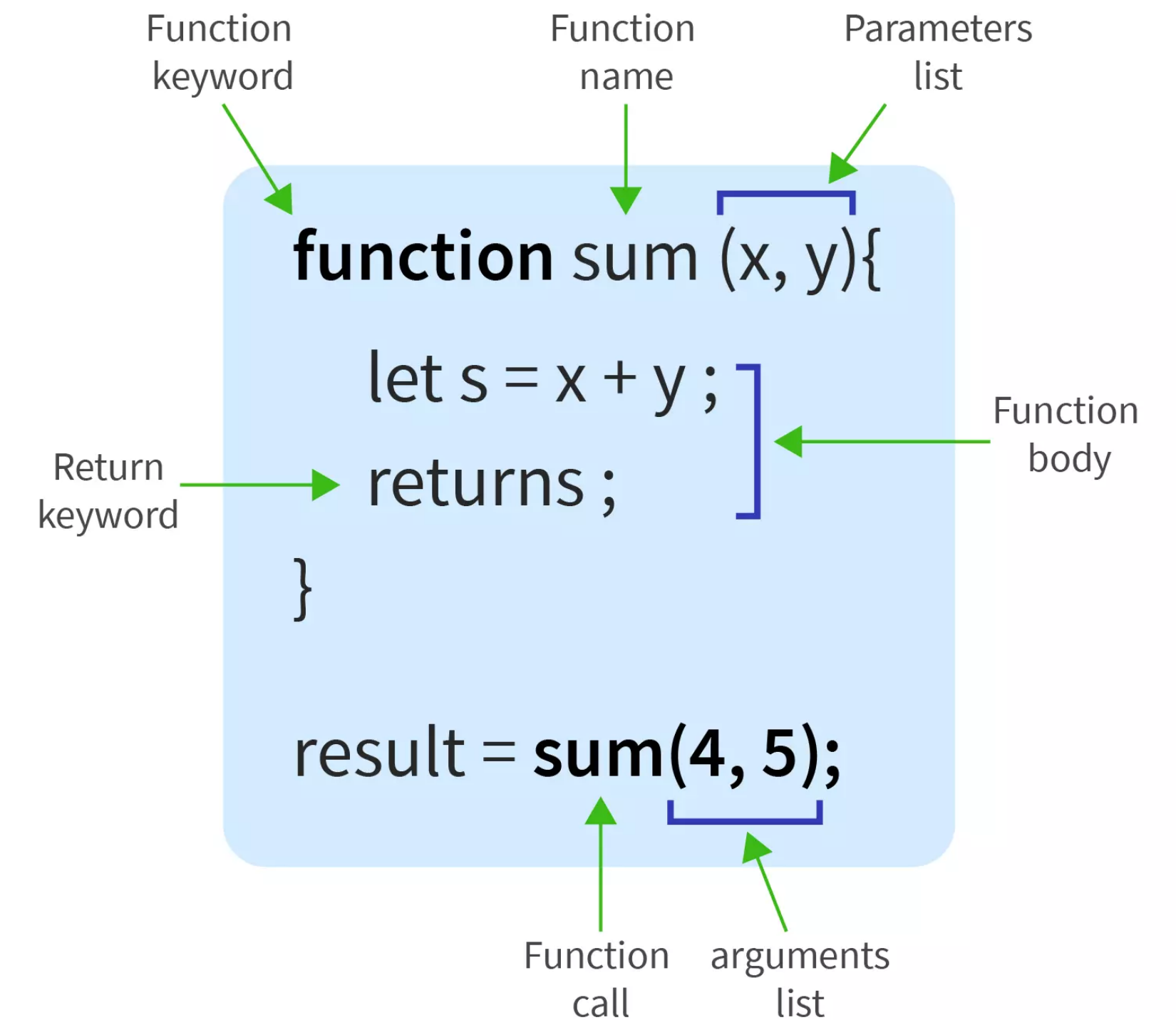
JavaScript function, belirli bir görevi yerine getirmek için tasarlanmış bir kod bloğudur.

## Neden Fonksiyonlar?

Fonksiyonlar ile kodu yeniden kullanabilirsiniz

Birçok kez kullanılacak kod yazabilirsiniz.

Farklı sonuçlar üretmek için aynı kodu farklı argümanlarla kullanabilirsiniz.



# JavaScript Classes

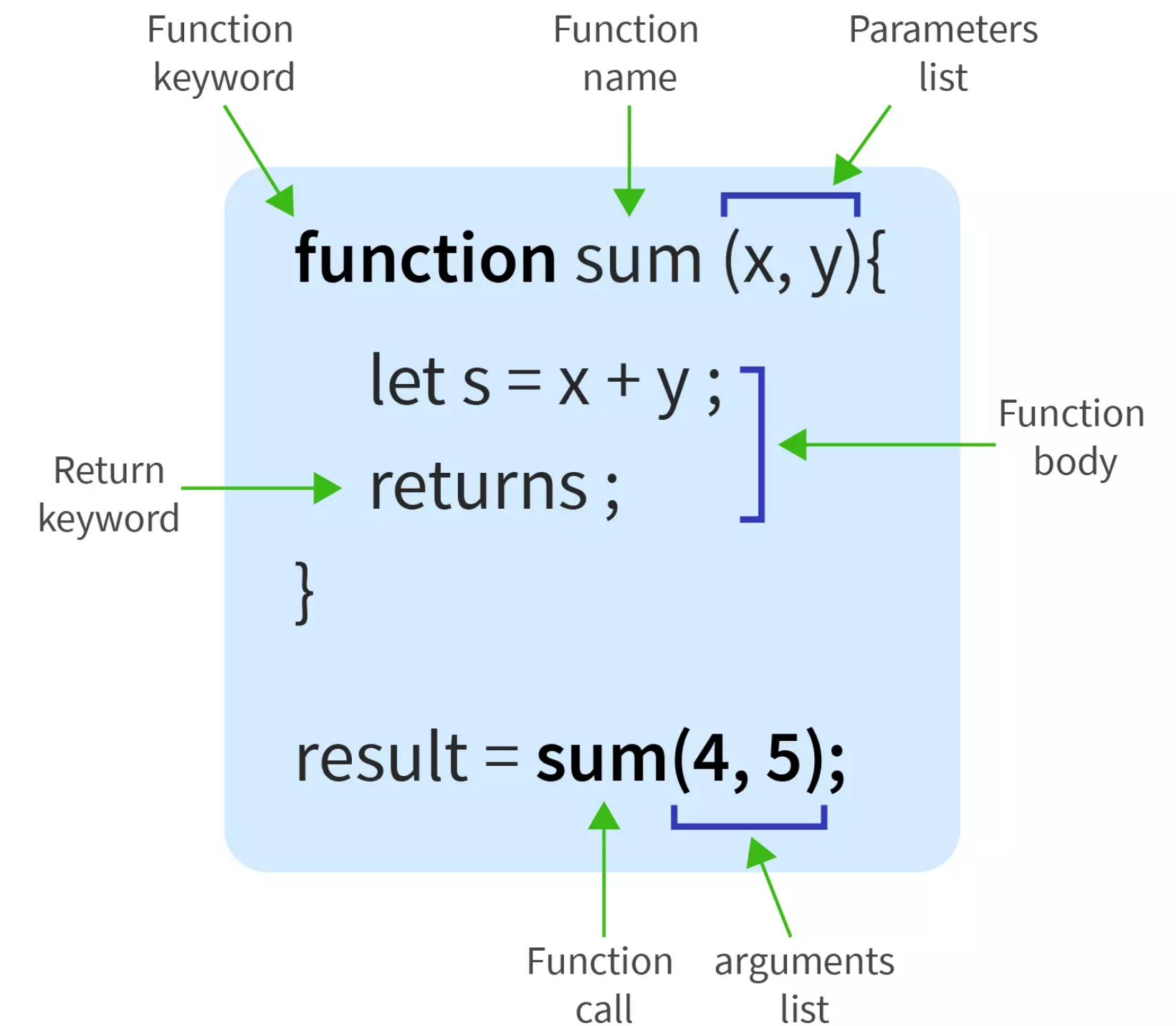
JavaScript function, belirli bir görevi yerine getirmek için tasarlanmış bir kod bloğudur.

## Neden Fonksiyonlar?

Fonksiyonlar ile kodu yeniden kullanabilirsiniz

Birçok kez kullanılacak kod yazabilirsiniz.

Farklı sonuçlar üretmek için aynı kodu farklı argümanlarla kullanabilirsiniz.



# JavaScript Objects

---

JavaScript'in veri türlerinden birini temsil eder. Çeşitli anahtarlı koleksiyonları ve daha karmaşık varlıkları saklamak için kullanılır.

## Object Properties

Özellikler değiştirilebilir, eklenebilir, silinebilir ve bazıları yalnızca okunabilir.

## Object Methods

Yöntem, bir özellik değeri olarak saklanan bir işlev tanımıdır.



# JavaScript Arrays

---

Birden çok öğeden oluşan bir koleksiyonun tek bir değişken adı altında saklanmasını sağlar ve yaygın dizi işlemlerini gerçekleştirmek için üyelere sahiptir.

- JavaScript Array Methods
- JavaScript Array Search
- JavaScript Array Sort
- JavaScript Array Iteration



# JavaScript Comparison and Logical Operators

Karşılaştırma ve Mantıksal operatörler doğru veya yanlış test etmek için kullanılır.

## Comparison Operators

Karşılaştırma operatörleri, değerler arasındaki eşitliği veya farkı belirlemek için ifadelerde kullanılır.

| Operator | Description                  | Comparing              |
|----------|------------------------------|------------------------|
| ==       | Eşit değer                   | <code>x == y</code>    |
| ===      | Eşit değer ve eşit tip       | <code>x === y</code>   |
| !=       | Eşit değer değil             | <code>x != y</code>    |
| !==      | Eşit değer ve eşit tip değil | <code>x !== y</code>   |
| >        | Büyüktür                     | <code>x &gt; y</code>  |
| <        | Küçüktür                     | <code>x &lt; y</code>  |
| >=       | Büyük veya eşittir           | <code>x &gt;= y</code> |
| <=       | Küçük veya eşittir           | <code>x &lt;= y</code> |

## Logical Operators

Mantıksal operatörler, değişkenler veya değerler arasındaki mantığı belirlemek için kullanılır.

| Operator | Description | Example                     |
|----------|-------------|-----------------------------|
| &&       | Ve          | <code>x &amp;&amp; y</code> |
|          | Veya        | <code>x    y</code>         |
| !=       | Değil       | <code>x != y</code>         |

## Conditional (Ternary) Operator

JavaScript ayrıca, bazı koşullara bağlı olarak bir değişkene değer atayan bir koşullu operatör içerir.

# JavaScript Conditional Statements

---

Koşullu ifadeler, farklı koşullara bağlı olarak farklı eylemler gerçekleştirmek için kullanılır.

## The if Statement

Bir koşul doğruysa yürütülecek bir JavaScript kodu bloğu belirtmek için if deyimini kullanın.

Daha karmaşık koşullar ve mantıksal ifadeler için uygundur.

## The Switch Statement

Yürütülecek birçok kod bloğundan birini seçmek için switch deyimini kullanın.

Sabit değerler üzerinde yapılan kontroller için daha uygundur ve daha temiz bir sözdizimi sunar.

# JavaScript Loops

---

Döngüler bir kod bloğunu birkaç kez çalıştırabilir.

## The For Loop

Belirli bir sayıda çalıştırılması gereken kod blokları için kullanılır. Üç ifade içerir: başlangıç ifadesi, koşul ifadesi ve artış/azalış ifadesi.

## The For In Loop

Bir nesnenin özellikleri arasında döngüler.

## The While Loop

Belirli bir koşul doğruyken bir kod bloğu boyunca döner.

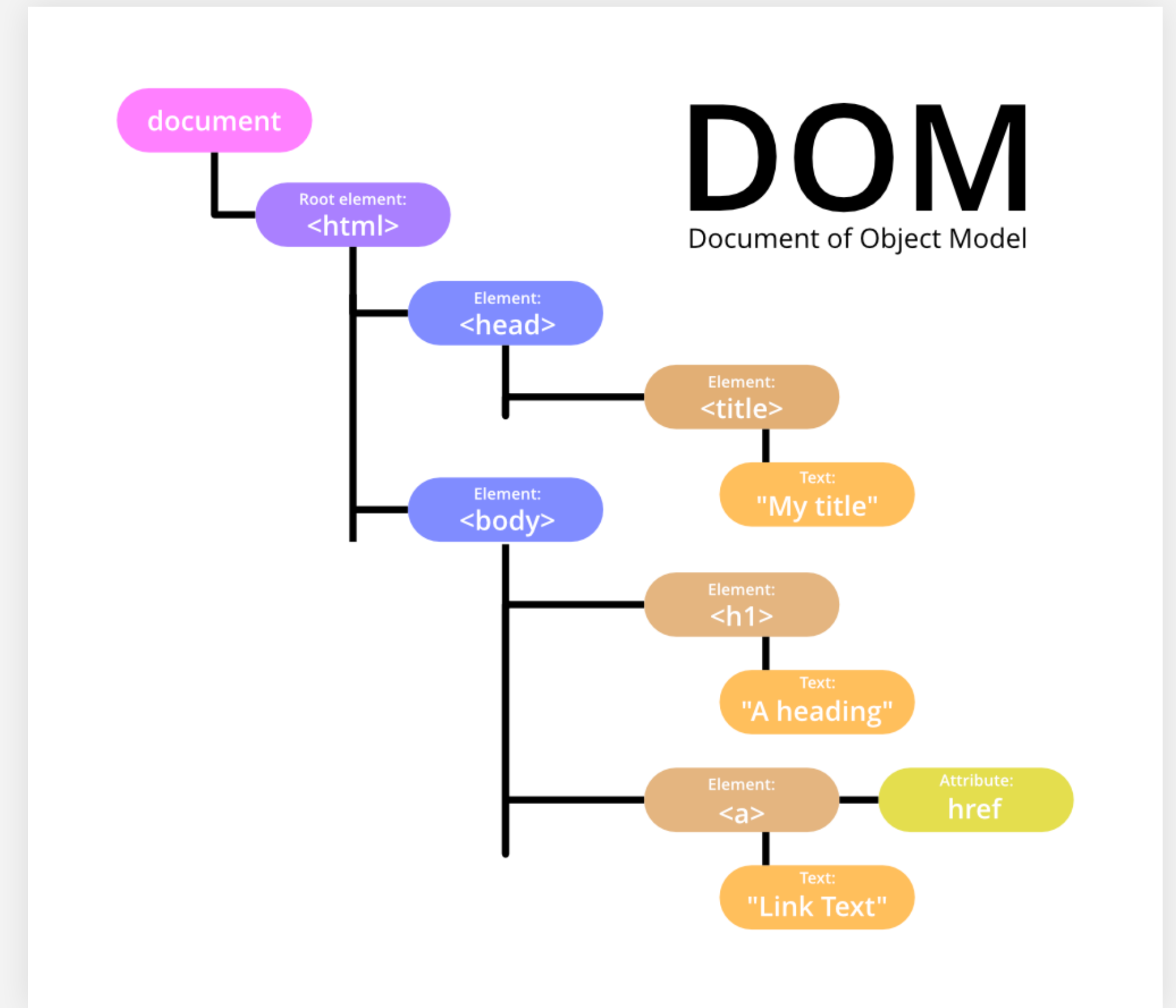
## The For Of Loop

Yinelenebilir bir nesnenin değerleri arasında döngüler .

# Document Object Model

HTML DOM'da Element nesnesi P, DIV, A, TABLE veya başka herhangi bir HTML ögesi gibi bir HTML ögesini temsil eder.

HTML DOM, web geliştiricilerinin HTML belgeleriyle etkileşime geçmesine, belge yapısını ve içeriğini dinamik olarak değiştirmesine olanak tanıyan güçlü bir modeldir. Bu yapı sayesinde web sayfaları interaktif ve kullanıcı dostu hale getirilebilir.



# Browser Object Model

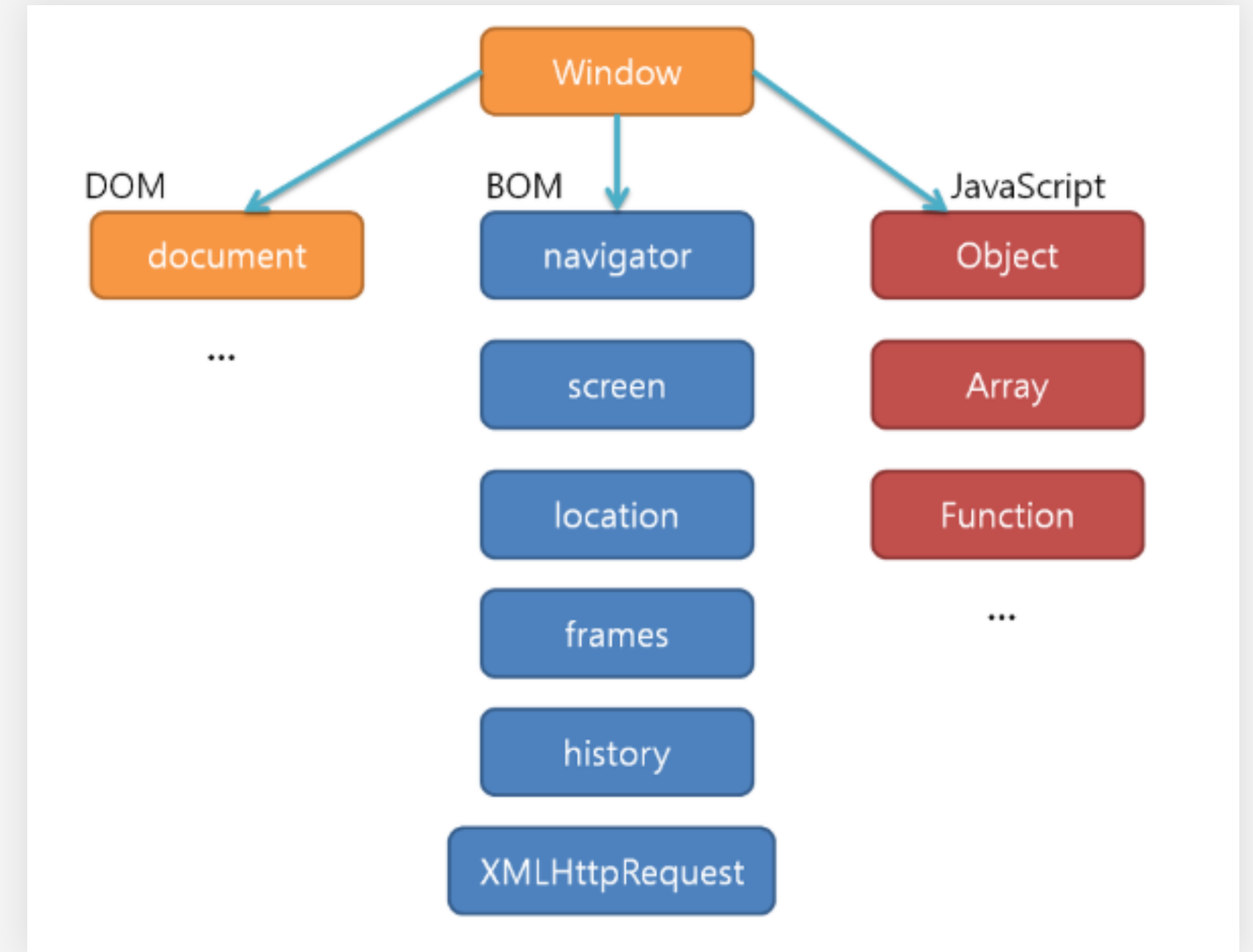
Tarayıcı Nesne Modeli (BOM) JavaScript'in tarayıcıyla “konuşmasını” sağlar.

Pencere nesnesi tüm tarayıcılar tarafından desteklenir.

Tüm global JavaScript nesneleri, fonksiyonları ve değişkenleri otomatik olarak pencere nesnesinin üyeleri haline gelir.

**DOM:** Bir belge içeriğini temsil eder ve HTML/XML belgelerini manipüle etmek için kullanılır. Belge düğümleriyle çalışmayı sağlar.

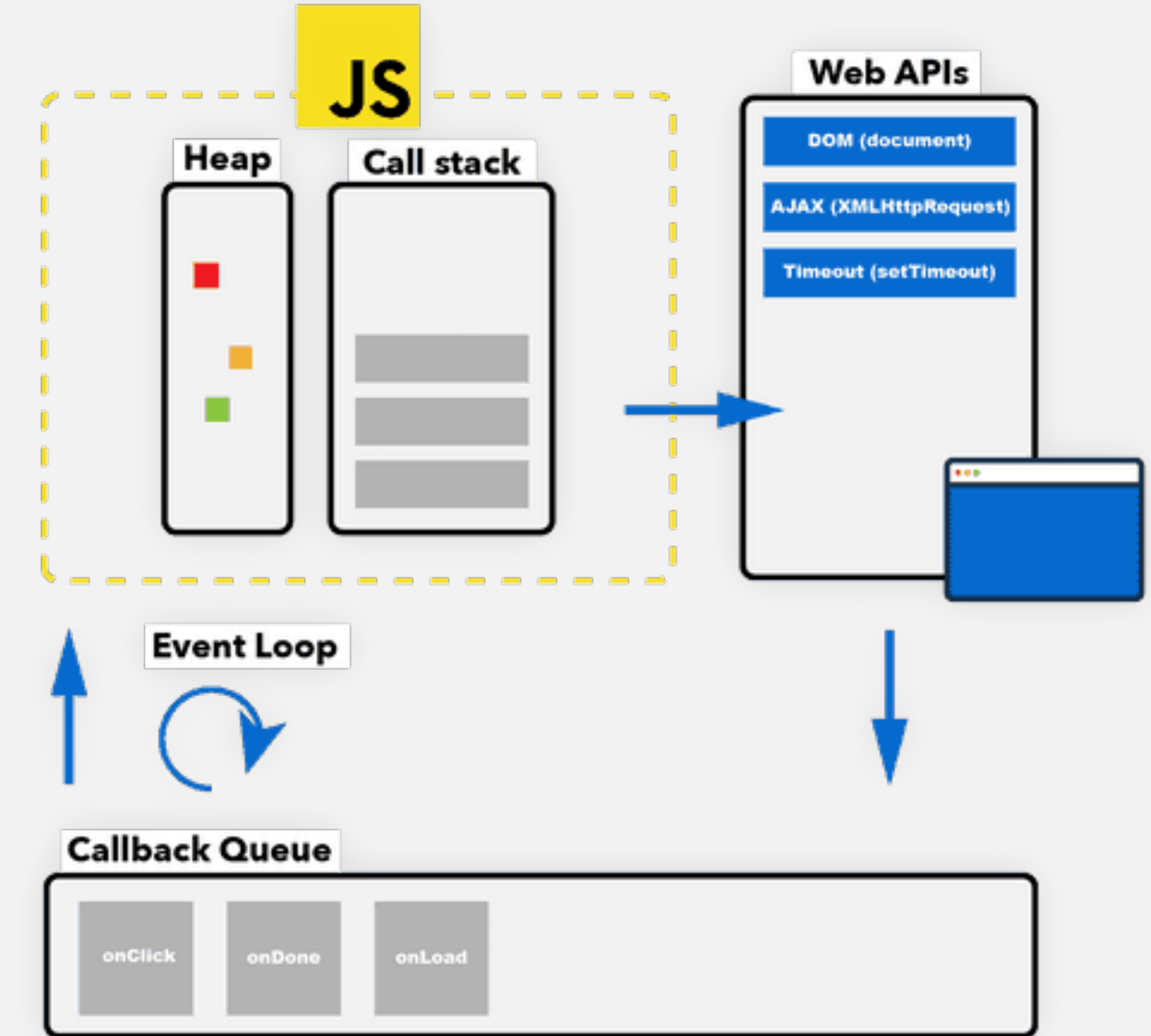
**BOM:** Tarayıcı ile ilgili nesneleri temsil eder ve tarayıcı penceresi, konumu, geçmişi ve diğer özelliklerini manipüle etmek için kullanılır.



# JavaScript Events

HTML'de bir olay meydana geldiğinde, olay belirli bir olay nesnesine aittir, örneğin bir fare tıklaması olayı MouseEvent nesnesine aittir.

| Object         | Description                  |
|----------------|------------------------------|
| AnimationEvent | CSS Animasyonları            |
| ClipboardEvent | Panonun değiştirilmesi       |
| DragEvent      | Sürükle ve bırak etkileşimi  |
| FocusEvent     | Odakla ilgili etkinlikler    |
| InputEvent     | Kullanıcı girişi             |
| KeyboardEvent  | Klavye etkileşimi            |
| MouseEvent     | Fare etkileşimi              |
| TouchEvent     | Dokunmatik etkileşim         |
| WheelEvent     | Fare-tekerlek etkileşimi     |
| UiEvent        | Kullanıcı arayüzü etkileşimi |





# JavaScript Event Propagation

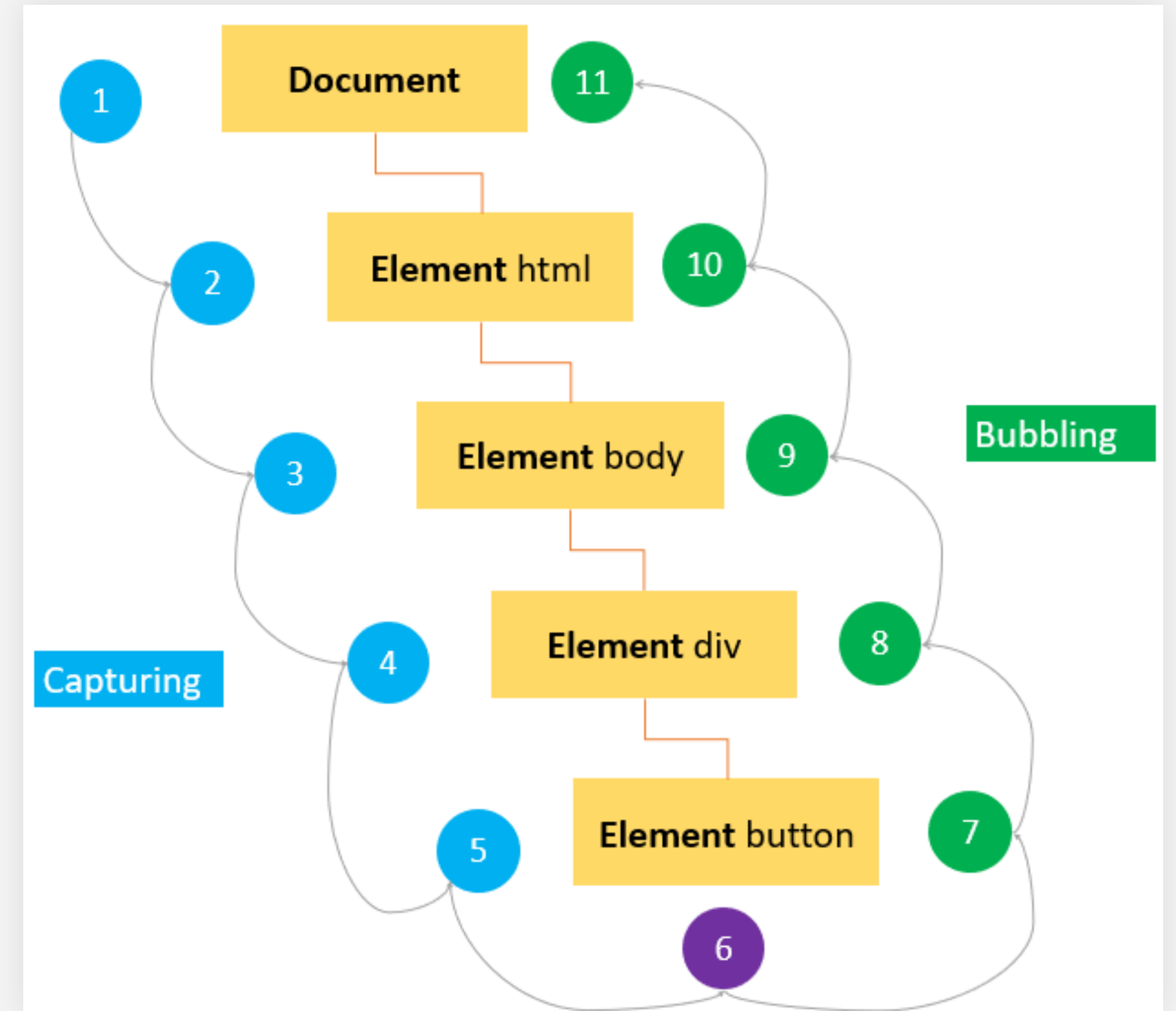
HTML DOM'da olay yayılma mekanizması iki fazdan oluşur: capturing ve bubbling.

## Bubbling

Bubbling fazı, olayın hedef elemandan başlayarak en dıştaki elemana kadar yayıldığı aşamadır. Bu aşamada, olay en içteki (target) elemandan dışarıya doğru ilerler.

## Capturing

Capturing fazı, olayın en dıştaki elemandan başlayarak hedef elemana kadar ulaşana kadar ilerlediği aşamadır. Bu aşamada, olay en dıştaki (root) elemandan içeriye doğru ilerler.





# Asynchronous JavaScript

İşlemlerin arka planda gerçekleşmesini ve bu işlemleri tamamlanmadan akışın devam etmesini sağlayan bir programlama yaklaşımıdır.

## JavaScript Callbacks

JavaScript'te **callback**, asenkron bir işlemin tamamlanmasının ardından çağrılan bir fonksiyon anlamına gelir.

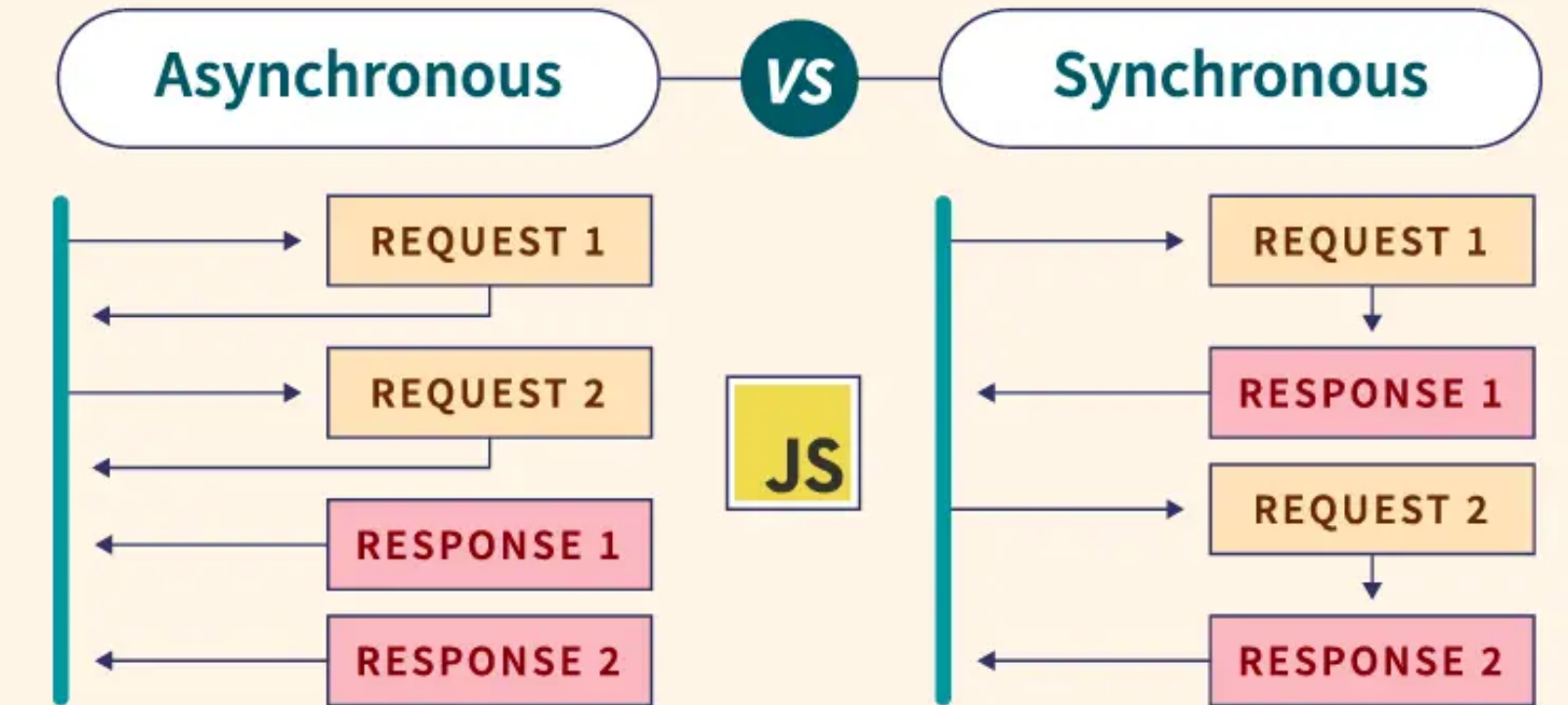
## JavaScript Promises

JavaScript'te asenkron işlemleri yönetmenin güçlü bir yoludur. Geri çağırımların yerine daha okunabilir ve yönetilebilir bir yapı sağlar.

## JavaScript Async/Await

Bir fonksiyondan önce gelen async anahtar sözcüğü, fonksiyonun bir promise döndürmesini sağlar.

**await** anahtar sözcüğü yalnızca bir async işlevi içinde kullanılabilir.



# JavaScript Web APIs

Bir Web API'si bir geliştiricinin hayalidir.

Web API, Web için bir uygulama programlama arayüzüdür.

Bir Browser API'si bir web tarayıcısının işlevselliğini genişletebilir.

Bir Server API'si, bir web sunucusunun işlevselliğini genişletebilir.

