



CodeCraft

Bootcamp^{'25}

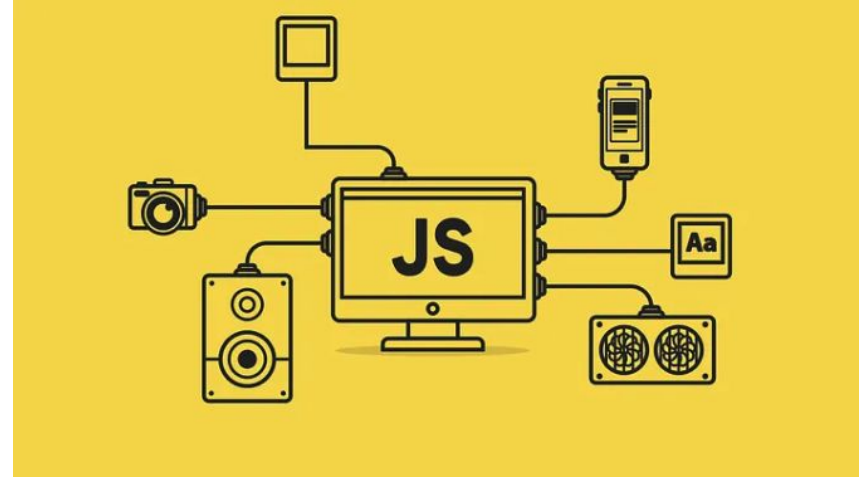
Javascript

JavaScript Syntax, Operators, Data Types,
Variables, Arrays, Objects, JSON

What is Javascript?

JavaScript, web sitelerinin etkileşimini ve dinamik davranışlarını geliştirmek için kullanılan çok yönlü bir programlama dilidir. Modern web geliştirme sürecinin temel bir bileşeni olup, kullanıcı deneyimlerini daha etkileyici hale getirmede önemli bir rol oynar.

JavaScript, yüksek seviyeli ve yorumlanan bir programlama dili olup, geliştiricilere web sayfalarına işlevsellik ekleme imkanı sağlar. İlk olarak web sitelerini daha etkileşimli hale getirmek amacıyla oluşturulmuş ve istemci tarafında çalışan bir betik dili olarak geliştirilmiştir. Bu, kodun uzaktaki bir sunucu yerine doğrudan kullanıcının tarayıcısında çalıştırılması anlamına gelir.

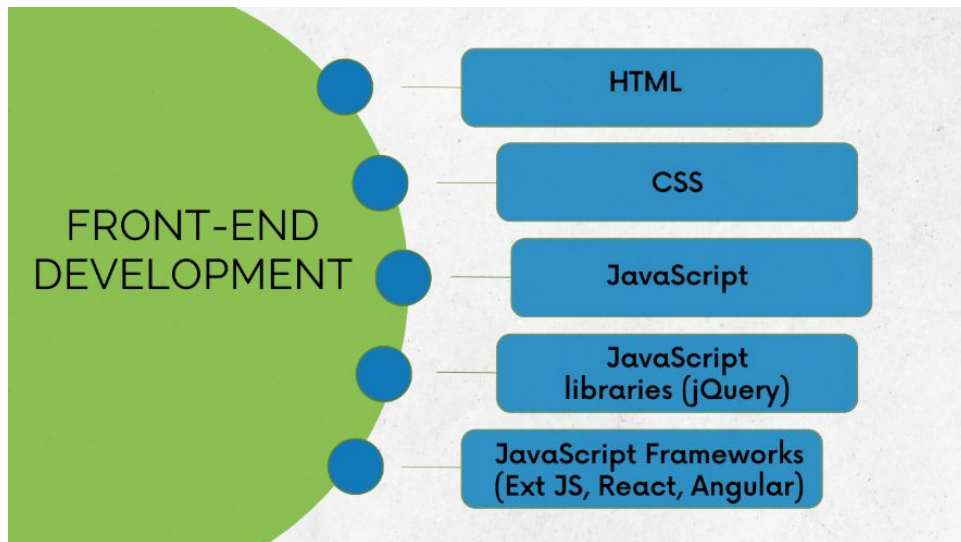


Where is javascript used?

Companies who use JavaScript on:			
<u>Front-End</u>	<u>Back-End</u>	<u>Mobile Apps</u>	<u>Desktop Apps</u>
Facebook Google Quora Uber	Walmart LinkedIn PayPal Uber	Facebook Instagram Uber Skype	Microsoft VS Code WhatsApp Slack
<i>99% of all top websites</i>	<i>many more...</i>	<i>many more...</i>	<i>many more...</i>

Importance of javascript in front-end development

- Etkileşimli Kullanıcı Deneyimi
- Dinamik İçerik Güncellemeleri
- Sayfa Yeniden Yüklemeden Veri İletişimi
- Zengin Web Uygulamaları
- Kullanıcı Girdisi ve Form Yönetimi
- Animasyonlar ve Görsel Etkiler



JavaScript Syntax

Syntax (Sözdizimi), bir programlama dilinde kodların doğru ve anlaşılır bir şekilde yazılmasını sağlayan kurallar bütünüdür.

- Değişken Tanımlama
- Satır Sonlandırma
- Büyük/Küçük Harf Duyarlılığı
- Yorum Satırları
- Fonksiyon Tanımlama
- Koşullu İfadeler
- Döngüler
- Operatör Kullanımı
- Noktalı Virgül Kullanımı
- Boşluk ve Girinti Kullanımı

```
Syntax
const myName = "John";
const myAge = 30;

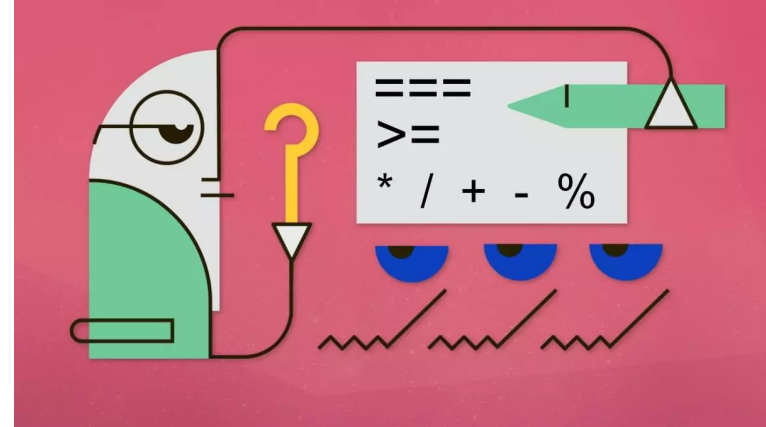
const greeting = () => {
  console.log(`My name is ${myName} and I am ${myAge} years old.`);
};

greeting();
```

Operators

Operatörler, değişkenler ve değerler üzerinde işlem yapmamızı sağlayan özel sembollerdir.

- Arithmetic (Aritmetik Operatörler)
- Assignment (Atama Operatörleri)
- Comparison (Karşılaştırma Operatörleri)
- Logical (Mantıksal Operatörler)
- Bitwise (Bit Düzeyinde Operatörler)
- Ternary (Üçlü Operatör)
- Type (Tür Operatörleri)



@sumusiriwardana

JAVASCRIPT OPERATOR CHEATSHEET

Arithmetic Operators

Operator	Description	Example
+	Adds two numeric operands.	<code>2 + 3 // 5</code>
-	Subtract right operand from left operand.	<code>3 - 2 // 1</code>
*	Multiply two numeric operands.	<code>2 * 3 // 6</code>
/	Divide left operand by right operand.	<code>10 / 5 // 2</code>
%	Returns remainder of two operands.	<code>5 % 3 // 2</code>
++	Increase operand value by one.	<pre>let x = 5; console.log(++x, x); // 6, 6 let y = 5; console.log(y++, x); // 5, 6</pre>
--	Decrease value by one.	<pre>let x = 5; console.log(--x, x); // 4, 4 let y = 5; console.log(y--, x); // 5, 4</pre>
**	Raise one operand to the power of the other operand.	<code>5 ** 3 // 125</code>

Assignment Operators

Operator	Description	Longform	Example
=	Assigns right operand value to the left operand.		<pre>let x = 20; x = 20</pre>
+=	Sums up left and right operand values and assigns the result to the left operand.	<code>x = x + 5</code>	<code>x += 5 // 25</code>
-=	Subtract right operand value from the left operand value and assigns the result to the left operand.	<code>x = x - 5</code>	<code>x -= 5 // 15</code>
*=	Multiply left and right operand values and assigns the result to the left operand.	<code>x = x * 5</code>	<code>x *= 5 // 100</code>
/=	Divide left operand value by right operand value and assign the result to the left operand.	<code>x = x / 5</code>	<code>x /= 5 // 4</code>
%=	Get the remainder of left operand divide by right operand and assign resulted remainder to the left operand.	<code>x = x % 5</code>	<code>x %= 5 // 0</code>

Comparison Operators

Operator	Description	Example
==	Returns true if the operands are equal	<pre>let x = 5; x == '5' // true</pre>
!=	Returns true if the operands are not equal	<code>x != '6' // true</code>
===	Returns true if the operands are equal and of the same type	<code>x === 5 // true</code>
!==	Returns true if the operands are the same type but are not equal or are different types	<code>x !== 5 // false</code>
>	Returns true if the left operand is greater than the right operand	<code>x > 5 // false</code>
>=	Returns true if left operand is greater than or equal to the right operand	<code>x >= 5 // true</code>
<	Returns true if the left operand is less than the right operand	
<=	Returns true if the left operand is less than or equal to the right operand	<code>x <= 5 // true</code>

Logical Operators

Operator	Description	Example
&&	Returns true if both the operands are true, else returns false	<pre>3 > 2 && 2 > 1 // true 3 < 2 && 2 > 1 // false</pre>
	Returns true if either of the operands is true; returns false if both are false	<pre>3 < 2 2 > 1 // true 3 < 2 2 < 1 // false</pre>
!	Returns true if the operand is false and vice-versa.	<pre>!true // false !false // true</pre>

Ternary Operators

Operator	condition ? expressionIfTrue : expressionIfFalse
Description	If the condition is true, the first expression will be executed. Otherwise, the second expression will be executed.
Example	<code>(2 === 2) ? 'x' : 'y'; // 'x'</code>

typeof Operator

Operator	typeof
Description	Returns a string indicating the data type
Example	<pre>console.log(typeof true); // 'boolean' console.log(typeof 300); // 'number' console.log(typeof 'Pete'); // 'string'</pre>

Data Types

JavaScript'te veri tipleri, değişkenlerin alabileceği farklı türdeki değerleri belirleyen kategorilerdir.

- **Primitive Types (Temel Veri Tipleri)**
 - String
 - Number
 - Boolean
 - Undefined
 - Null
 - BigInt
- **Reference Types (Referans Veri Tipleri)**
 - Object
 - Array
 - Function

Non-zero value



null



0



undefined



Variables

Değişkenler, verileri saklamak ve manipüle etmek için kullanılan isimlendirilmiş alanlardır.

- **let**
 - **Kapsam (Scope):** Blok seviyesinde geçerlidir (block-scoped).
 - **Değer değişikliği:** Değeri daha sonra değiştirilebilir.
- **const**
 - **Kapsam (Scope):** Blok seviyesinde geçerlidir (block-scoped).
 - **Değer değişikliği:** Değeri bir kez atandıktan sonra değiştirilemez.
- **var**
 - **Kapsam (Scope):** Fonksiyon seviyesinde geçerlidir (function-scoped) ve daha eski bir yöntemdir.
 - **Değer değişikliği:** Değeri değiştirilebilir, ancak modern JavaScript'te **let** ve **const** tercih edilir.



Arrays

Diziler, birden fazla değeri tek bir değişkende sıralı şekilde saklamak için kullanılan veri yapılarıdır ve her bir öge, indeks numarasıyla erişilebilir.

- **Methods (Array Metotları)**

- length
- shift()
- unshift()
- pop()
- push()
- join()
- concat()

- **Iteration (Tekrarlama Metotları)**

- forEach
- map()
- filter()
- reduce()
- every()
- some()

```
pets.shift()
removes

pets = [ "cats" , "dogs", "birds" ]

pets.unshift("mice")
inserts

pets.push("mice")
inserts

pets.pop()
removes
```

Objects

Objeler, anahtar-değer çiftlerinden oluşan ve verileri yapılandırılmış bir şekilde saklamak için kullanılan veri yapılarıdır. Objeler, farklı türdeki verileri bir arada tutabilir.

- **Obje Oluşturma**
- **Obje Özelliklerine Erişim**
- **Obje Özelliklerine Değer Atama**
- **Yeni Özellik Ekleme**
- **Obje Özelliğini Silme**
- **Obje İçinde Fonksiyonlar**
- **this Anahtar Kelimesi**
- **Obje Kopyalama**
- **Objelerin Geçerli Özelliklerini Listeleme**

```
objects.js

const person = {
  firstName: "John",
  lastName: "Doe",
  age: 20,
  fullName: () => {
    return `${person.firstName} ${person.lastName}`;
  }
};

person.firstName // John
person.lastName  // Doe

person.fullName(); // John Doe
```

What is JSON ?

JSON (JavaScript Object Notation), veri değişimi için kullanılan hafif, metin tabanlı bir veri formatıdır. Özellikle sunucu ile istemci arasında veri iletimi için yaygın olarak kullanılır. JSON, insan tarafından okunabilir ve makine tarafından kolayca işlenebilir olduğu için veri paylaşımında tercih edilir.

Metin Tabanlı: JSON, sadece metin formatında veri içerir, bu da taşınmasını ve saklanmasını kolaylaştırır.

Anahtar-Değer Çiftleri: JSON, verileri **anahtar-değer (key-value)** çiftleri şeklinde saklar.

Yapısal ve Hiyerarşiktir: JSON, **diziler (arrays)** ve **nesneler (objects)** gibi veri yapılarını destekler, bu da karmaşık verilerin temsil edilmesini sağlar.

Dil Bağımsızdır: JSON, JavaScript'ten türemiş olsa da, çoğu programlama diliyle uyumludur ve veri iletimi için evrensel bir format olarak kullanılır.

```
{
  "Name": "John",
  "Age": 34,
  "StateOfOrigin": "England",
  "Pets": [
    {
      "Type": "Cat",
      "Name": "MooMoo",
      "Age": 3.4
    },
    {
      "Type": "Squirrel",
      "Name": "Sandy",
      "Age": 7
    }
  ]
}
```

JSON

- **JSON ile Objelerin Arasındaki Fark**
 - JSON, sadece veri taşıma amacıyla kullanılan bir formatken, JavaScript nesneleri daha dinamik ve işlevsel olabilir, yani fonksiyonlar içerebilir.
 - JSON sadece veri içermeli, fonksiyonlar içeremez.
- **JSON Metotları**
 - `JSON.stringify()` metodu, bir JavaScript objesini JSON formatına dönüştürür.
 - `JSON.parse()` metodu, JSON formatındaki bir veriyi JavaScript objesine dönüştürür.

```
{
  "name": "John",
  "age": 20,
  "height": 1.92,
  "gender": "M",
  "salary": 70000,
  "married": true,
  "children": [
    {"name": "Tom", "age": 9, "gender": "M"},
    {"name": "Ava", "age": 7, "gender": "F"}
  ]
}
```

Homework

prompt() ile kullanıcıdan isim, yaş ve meslek almalı ve bir nesne (object) içinde saklamalı.

Bir dizi (array) kullanarak ürünleri sepete eklemeli ve listelemeli. (name, price)

reduce() metodunu kullanarak sepetin toplam fiyatını hesaplamalı.

Kullanıcıdan dinamik olarak ürün eklemesini isteyebiliriz (*)

Bir ürünü sepetten çıkarmalarını sağlayacak bir fonksiyon nasıl yazılabilir bunu araştırmalı ve yapabiliyorsa yapmalı. (*)

```
Adınız nedir? Ahmet
Yaşınız kaç? 25
Mesleğiniz nedir? Yazılımcı
Kullanıcı Bilgileri: { name: 'Ahmet', age: 25, job: 'Yazılımcı' }

Sepete eklemek istediğiniz ürünü yazın: Laptop
Ürünün fiyatı: 25000
Laptop ürünü sepete eklendi. Fiyat: 25000 TL

Sepete eklemek istediğiniz ürünü yazın: Mouse
Ürünün fiyatı: 500
Mouse ürünü sepete eklendi. Fiyat: 500 TL

Sepete eklemek istediğiniz ürünü yazın: q
Sepetiniz: [ { product: 'Laptop', price: 25000 }, { product: 'Mouse', price: 500 } ]
Toplam Fiyat: 25500 TL
```