**🧠 JavaScript কী?**

**JavaScript** হলো একটি প্রোগ্রামিং ভাষা যা ওয়েবসাইটে "বুদ্ধি" যোগ করে।  
তুমি যখন কোনো বাটনে ক্লিক করো, ফর্ম পূরণ করো, বা কোনো লেখা বদলে যায় – এসব কিছু **JavaScript** দিয়েই করা হয়।

## ✅ ১. alert() → স্ক্রিনে বার্তা দেখায়

javascript

Copy code

alert("হ্যালো দুনিয়া!");

### ➤ ব্যাখ্যা:

* alert() হল একটি **পপ-আপ বার্তা**।
* যখন এই কোড চালাবে, তখন স্ক্রিনে একটি মেসেজ বক্স আসবে – “হ্যালো দুনিয়া!”

📌 এটা সাধারণত ব্যবহার করা হয় ইউজারকে কিছু জানাতে।

📌 এটা তোমার ব্রাউজারে একটি মেসেজ বক্স দেখাবে। যেমন – সতর্কতা বা শুভেচ্ছা।

## ✅ ২. console.log() → কনসোলে বার্তা দেখায়

javascript

Copy code

console.log("আমি জাভাস্ক্রিপ্ট শিখছি");

### ➤ ব্যাখ্যা:

* এটি স্ক্রিনে কিছু দেখায় না। ব্রাউজারের **ডেভেলপার কনসোলে** (F12 টিপে Console ট্যাব) দেখা যায়।
* প্রোগ্রাম ঠিকভাবে কাজ করছে কিনা তা বোঝার জন্য আমরা এটা ব্যবহার করি।

📌 এই কোডটি ব্রাউজারের কনসোল-এ (Inspect > Console) মেসেজ দেখাবে। ডেভেলপাররা এটা দিয়ে পরীক্ষা করে।

## ✅ ৩. ভেরিয়েবল (let, var, const)

javascript

Copy code

let নাম = "রাহিম";

var বয়স = ২৫;

const দেশ = "বাংলাদেশ";

### ➤ ব্যাখ্যা:

| **শব্দ** | **কাজ** |
| --- | --- |
| let | সাধারণ ভেরিয়েবল, পরিবর্তন করা যায় |
| var | পুরনো স্টাইল, এখন কম ব্যবহার হয় |
| const | একবার মান দিলে পরিবর্তন করা যায় না |

📌 **ভেরিয়েবল** হলো এমন একটি "ঘর" যেখানে তুমি তথ্য রেখে দাও।

📌 ভেরিয়েবল মানে হলো **তথ্য জমিয়ে রাখা**। তুমি নাম, বয়স বা যেকোনো তথ্য রাখার জন্য let, var বা const ব্যবহার করতে পারো।

**✅ ৪. ফাংশন (Function)**

javascript

CopyEdit

function বল\_স্বাগতম(নাম) {

console.log("স্বাগতম, " + নাম);

}

বল\_স্বাগতম("রাহিম");

### ব্যাখ্যা:

* ফাংশন হলো একটি **কাজের প্যাকেট**।
* তুমি যখন বল\_স্বাগতম("রাহিম") লেখো, তখন সেটি ফাংশন চালায়।

📌 এটি বারবার ব্যবহার করার জন্য খুব উপকারী।

📌 ফাংশন হলো এমন একটি কোড ব্লক যেটা তুমি বার বার চালাতে পারো। এতে ইনপুট (parameter) নিতে পারো এবং কাজ করাতে পারো।

**✅ ৫. If-Else (শর্ত)**

javascript

CopyEdit

let নাম্বার = ৭০;

if (নাম্বার >= ৫০) {

console.log("তুমি পাশ করেছো");

} else {

console.log("তুমি ফেল করেছো");

}

### ➤ ব্যাখ্যা:

* if মানে "যদি"।
* যদি নাম্বার ৫০ বা তার বেশি হয় → "পাশ করেছো"
* না হলে (else) → "ফেল করেছো"

📌 আমরা সিদ্ধান্ত নিতে if/else ব্যবহার করি।

📌 এটা শর্ত দেখে সিদ্ধান্ত নেয়। যদি ৫০ বা তার বেশি হয় তাহলে "পাশ", নাহলে "ফেল"।

**✅ ৬. লুপ (Loop)**

for (let i = ১; i <= ৫; i++) {

console.log("নম্বর: " + i);

}

### ➤ ব্যাখ্যা:

* for লুপ ১ থেকে ৫ পর্যন্ত বার বার ঘুরে কাজ করবে।
* i++ মানে প্রতিবার i এক করে বাড়ে।

📌 যেসব কাজ বার বার করতে হয় সেগুলোর জন্য **লুপ খুব দরকারি।**

📌 এই কোড ১ থেকে ৫ পর্যন্ত সংখ্যা দেখাবে। for লুপ বার বার একই কাজ করতে সাহায্য করে।

## ✅ ৭. ইভেন্ট (Event) → ক্লিক করলে কিছু হবে

html

Copy code

<button onclick="sayHello()">ক্লিক করো</button>

<script>

function sayHello() {

alert("তুমি বাটনে ক্লিক করেছো!");

}

</script>

### ➤ ব্যাখ্যা:

* onclick ইভেন্ট যখন বাটনে ক্লিক করো, তখন sayHello() ফাংশন চালায়।
* এবং alert() দিয়ে বার্তা দেখায়।

📌 ইভেন্ট মানে হলো ইউজার কিছু করলে সেই অনুযায়ী কাজ।

## ✅ ৮. HTML-এ টেক্সট বদলানো

html

Copy code

<p id="demo">পুরানো লেখা</p>

<button onclick="changeText()">টেক্সট বদলাও</button>

<script>

function changeText() {

document.getElementById("demo").innerHTML = "নতুন লেখা!";

}

</script>

### ➤ ব্যাখ্যা:

* getElementById("demo") দিয়ে HTML-এ যেই আইডি আছে, সেটা খুঁজে আনে।
* .innerHTML মানে টেক্সট বদলানো।

📌 এতে করে JavaScript দিয়ে HTML-এর লেখা বা স্টাইল বদলানো যায়।

## ✅ ৯. ফর্ম যাচাই (Form Validation) – সহজ উদাহরণ

html

Copy code

<input type="text" id="name">

<button onclick="checkName()">যাচাই করো</button>

<script>

function checkName() {

let name = document.getElementById("name").value;

if (name == "") {

alert("নাম লিখো");

} else {

alert("ধন্যবাদ, " + name);

}

}

</script>

### ➤ ব্যাখ্যা:

* ইউজার কিছু না লিখলে বার্তা দিচ্ছে: "নাম লিখো"
* কিছু লিখলে বলে: "ধন্যবাদ"

📌 ফর্ম সাবমিট হওয়ার আগে যাচাই করার জন্য ব্যবহার হয়।

**✅ ১০. HTML-এ JavaScript ব্যবহার**

html

CopyEdit

<button onclick="sayHello()">ক্লিক করো</button>

<script>

function sayHello() {

alert("তুমি বাটনে ক্লিক করেছো!");

}

</script>

📌 **ব্যাখ্যা**:  
এই কোডে, তুমি বাটনে ক্লিক করলে JavaScript চালু হবে এবং একটি মেসেজ দেখাবে।

**✅ উপসংহার:**

| **বিষয়** | **কি কাজ করে** |
| --- | --- |
| alert() | স্ক্রিনে মেসেজ দেখায় |
| console.log() | কনসোলে তথ্য দেখায় |
| let/const | ভেরিয়েবল রাখে |
| function | কোড প্যাক করে চালানো যায় |
| if/else | শর্ত দিয়ে সিদ্ধান্ত নেয় |
| for | বারবার কাজ করতে |
| onclick | ক্লিক করলে কাজ হয় |
| DOM (getElementById) | HTML-এর কিছু বদলাতে |

**🔁 সংক্ষেপে মনে রাখো:**

| **বিষয়** | **ব্যাখ্যা** |
| --- | --- |
| alert() | স্ক্রিনে বার্তা দেখায় |
| console.log() | কনসোলে বার্তা দেখায় |
| let, var, const | ভেরিয়েবল তৈরি করে |
| function | কোডের পুনঃব্যবহারযোগ্য ব্লক |
| if, else | শর্ত অনুযায়ী কাজ |
| for | লুপ চালাতে ব্যবহৃত |

😊

## 🔍 কন্ডিশন (Condition) কী?

**Condition** মানে:  
👉 "যদি কিছু সত্য হয়, তাহলে কিছু করো — না হলে অন্য কিছু করো।"

JavaScript-এ এটি আমরা লিখি:

javascript

Copy code

if (শর্ত) {

// শর্ত সত্য হলে যা হবে

} else {

// শর্ত মিথ্যা হলে যা হবে

}

## 🧪 উদাহরণ ১: পাশ না ফেল?

javascript

Copy code

let নাম্বার = ৪৫;

if (নাম্বার >= ৫০) {

console.log("তুমি পাশ করেছো");

} else {

console.log("তুমি ফেল করেছো");

}

### ✅ ব্যাখ্যা:

* নাম্বার >= ৫০ → এটা একটি **শর্ত** (condition)
* যদি সত্য হয় (যেমন ৬০, ৭০), তাহলে "পাশ"
* না হলে (যেমন ৪৫, ৩০), তাহলে "ফেল"

## 🔁 কন্ডিশনের অংশগুলো:

| **চিহ্ন** | **অর্থ** |
| --- | --- |
| == | সমান কিনা (value) |
| === | একদম সমান (value + type) |
| != | সমান না |
| < | ছোট |
| > | বড় |
| <= | ছোট বা সমান |
| >= | বড় বা সমান |

## 🧪 উদাহরণ ২: বয়স যাচাই

javascript

Copy code

let বয়স = ১৮;

if (বয়স >= ১৮) {

console.log("তুমি ভোট দিতে পারো");

} else {

console.log("তুমি এখনো ছোট");

}

## 🧠 Else If ব্যবহার করলে?

javascript

Copy code

let নাম্বার = ৭৫;

if (নাম্বার >= ৮০) {

console.log("A+");

} else if (নাম্বার >= ৭০) {

console.log("A");

} else if (নাম্বার >= ৬০) {

console.log("A-");

} else {

console.log("Fail");

}

### ✅ ব্যাখ্যা:

* এখানে আমরা ধাপে ধাপে যাচাই করছি:  
  আগে ৮০+, না হলে ৭০+, না হলে ৬০+, না হলে ফেল।

📌 else if এর মানে হল: "আগেরটা সত্য না হলে এটাও দেখো"।

## 🧪 উদাহরণ ৩: দুটি ভেরিয়েবল যাচাই

javascript

Copy code

let userName = "admin";

let password = "1234";

if (userName === "admin" && password === "1234") {

console.log("লগইন সফল");

} else {

console.log("ভুল ইউজারনেম অথবা পাসওয়ার্ড");

}

### ✅ ব্যাখ্যা:

* && মানে **এবং** (দুটোই সত্য হতে হবে)
* যদি ইউজারনেম ও পাসওয়ার্ড ঠিক হয় = সফল

## ✅ গুরুত্বপূর্ণ লজিক্যাল অপারেটর:

| **অপারেটর** | **মানে** |
| --- | --- |
| && | AND – দুইটা শর্তই সত্য হলে |
| ` |  |
| ! | NOT – সত্যকে মিথ্যা করে, মিথ্যাকে সত্য |

## 🧪 ছোট উদাহরণ: OR ব্যবহার

javascript

Copy code

let দিন = "শনিবার";

if (দিন === "শুক্রবার" || দিন === "শনিবার") {

console.log("আজ ছুটি");

} else {

console.log("আজ অফিস");

}

📌 এখানে যদি যেকোনো একদিন হয় — তাহলে ছুটি দেখাবে।

## 🔚 উপসংহার

### শিখলো:

* if, else, else if কিভাবে কাজ করে
* তুলনা চিহ্ন (==, >, <, !=, ===)
* &&, ||, ! এর ব্যবহার

## ✅ ১. ফর্ম যাচাই (Form Validation)

📌 উদ্দেশ্য:  
ফর্ম সাবমিট করার আগে চেক করা যে ইউজার ঠিকভাবে তথ্য লিখেছে কিনা।

### 🎓 উদাহরণ: নাম না লিখলে সতর্ক করো

html

Copy code

<form onsubmit="return validateForm()">

<input type="text" id="name" placeholder="তোমার নাম লিখো">

<button type="submit">সাবমিট</button>

</form>

<script>

function validateForm() {

let name = document.getElementById("name").value;

if (name === "") {

alert("দয়া করে নাম লিখুন");

return false; // ফর্ম সাবমিট হতে দেবে না

}

alert("ধন্যবাদ, ফর্ম সাবমিট হয়েছে!");

return true;

}

</script>

### 🧠 ব্যাখ্যা:

* onsubmit="return validateForm()": ফর্ম সাবমিটের আগেই validateForm() ফাংশন চালায়
* যদি ইনপুট খালি থাকে → alert() দেখায় এবং ফর্ম বন্ধ করে দেয়
* সব ঠিক থাকলে → সাবমিট হয়

## ✅ ২. API কল (API Call)

📌 উদ্দেশ্য:  
**API** মানে হল অন্য সার্ভার বা ডেটা সোর্স থেকে তথ্য আনা।

### 🎓 উদাহরণ: Public API দিয়ে ইউজারের ডেটা আনা

html

Copy code

<button onclick="loadUser()">ইউজার দেখাও</button>

<p id="output"></p>

<script>

function loadUser() {

fetch("https://jsonplaceholder.typicode.com/users/1")

.then(response => response.json())

.then(data => {

document.getElementById("output").innerHTML =

"নাম: " + data.name + "<br>" +

"ইমেইল: " + data.email;

})

.catch(error => {

console.log("ভুল হয়েছে:", error);

});

}

</script>

### 🧠 ব্যাখ্যা:

* fetch() ব্যবহার করে API থেকে ডেটা আনা হয়
* .then() দিয়ে ডেটা পাওয়ার পর কী করতে হবে তা বলে
* .catch() দিয়ে যদি কোনো ভুল হয় তা ধরে ফেলা হয়

📌 এটি AJAX বা API কলের আধুনিক ও সহজ উপায়

## ✅ ৩. গেম বানানো (Game Making)

📌 উদ্দেশ্য:  
HTML + CSS + JavaScript দিয়ে ছোট ছোট গেম বানানো যায়।

### 🎮 🎓 উদাহরণ: "Guess the Number" গেম

html

Copy code

<h2>১ থেকে ১০ পর্যন্ত নাম্বার ভাবো!</h2>

<input type="number" id="guess">

<button onclick="checkGuess()">জাচাই করো</button>

<p id="result"></p>

<script>

let secret = Math.floor(Math.random() \* 10) + 1;

function checkGuess() {

let userGuess = document.getElementById("guess").value;

if (userGuess == secret) {

document.getElementById("result").innerHTML = "🎉 সঠিক! তুমি জিতে গেছো!";

} else {

document.getElementById("result").innerHTML = "❌ ভুল! আবার চেষ্টা করো।";

}

}

</script>

### 🧠 ব্যাখ্যা:

* Math.random() দিয়ে একটি **গোপন সংখ্যা** বানানো হয়
* ইউজার যখন ইনপুট দেয়, সেটি আমরা checkGuess() ফাংশনে যাচাই করি
* মিললে জয়, না মিললে ট্রাই আবার

## ✅ সংক্ষেপে:

| **বিষয়** | **কাজ** |
| --- | --- |
| **Form Validation** | ইনপুট যাচাই করে ফর্ম সাবমিট কন্ট্রোল করে |
| **API Call** | বাইরের সার্ভার থেকে ডেটা নিয়ে আসে (যেমন ইউজার লিস্ট) |
| **Game Making** | ইন্টার‍্যাকটিভ গেম বানায় (যেমন Guess the Number) |

**চলো দেখি, JavaScript দিয়ে কী কী করা যায়, সহজভাবে — উদাহরণ সহ:**

**🔥 ১. ওয়েবসাইটে ইন্টার‌্যাকশন তৈরি করতে**

📌 উদাহরণ:

* বাটনে ক্লিক করলে কিছু ঘটে (যেমন: pop-up)
* ফর্ম ভেরিফিকেশন (input ভুল হলে দেখানো)
* টেক্সট/ছবি/স্টাইল পরিবর্তন

javascript

Copy code

document.getElementById("btn").onclick = function() {

alert("তুমি ক্লিক করেছো!");

}

🧠 এটা দিয়ে ওয়েবসাইট "বুদ্ধিমান" হয়।

**🔥 ২. ডায়নামিক ওয়েব অ্যাপ বানাতে**

📌 উদাহরণ:

* চ্যাট অ্যাপ (Messenger, WhatsApp Web)
* লাইভ নিউজ/ভোট/কমেন্ট দেখানো
* Instagram বা Facebook-এর নিউজফিড

💡 এই কাজে JavaScript + HTML + CSS + Backend (Node.js বা অন্য API) দরকার হয়।

**🔥 ৩. গেম বানাতে (Game Development)**

📌 উদাহরণ:

* Guess the Number গেম
* স্নেক গেম
* বড় 2D/3D গেম (Canvas বা Phaser.js দিয়ে)

🧠 JavaScript দিয়ে সহজ ও মাঝারি লেভেলের ব্রাউজার গেম বানানো যায়।

**🔥 ৪. মোবাইল অ্যাপ বানাতে (React Native)**

📌 JavaScript দিয়ে মোবাইল অ্যাপ বানানো যায় React Native দিয়ে:

* Android + iOS অ্যাপ
* একই কোড দুই প্ল্যাটফর্মে চলে

🎯 উদাহরণ: Facebook, Instagram, Skype (সব React Native দিয়ে বানানো)

**🔥 ৫. ডেস্কটপ অ্যাপ বানাতে (Electron.js)**

📌 উদাহরণ:

* Visual Studio Code
* Discord
* Spotify (ডেস্কটপ অ্যাপ)

📦 JavaScript + Electron.js দিয়ে ডেস্কটপ অ্যাপ তৈরি করা যায় — Windows, Mac, Linux সব জায়গায় চলে।

**🔥 ৬. সার্ভার বানাতে (Node.js)**

📌 JavaScript শুধু ব্রাউজারেই নয়, সার্ভারেও চলে (Node.js দিয়ে)।  
এটা দিয়ে বানানো যায়:

* API
* ব্যাকএন্ড সার্ভার
* ফাইল হ্যান্ডলিং, ডেটাবেজ কানেকশন

🧠 এর মানে, JavaScript দিয়ে ফুল ওয়েব অ্যাপ বানানো যায় – frontend + backend সব।

**🔥 ৭. API কল করে ডেটা আনা/দেখানো**

📌 উদাহরণ:

* আবহাওয়ার তথ্য দেখানো
* ইউজার প্রোফাইল/পোস্ট লোড
* লাইভ খেলা/ভোট রেজাল্ট

javascript

Copy code

fetch("https://api.example.com/data")

.then(res => res.json())

.then(data => console.log(data));

**🔥 ৮. Machine Learning / AI (TensorFlow.js)**

📌 হ্যাঁ! এখন JavaScript দিয়েও মেশিন লার্নিং করা যায় TensorFlow.js দিয়ে।

* ছবি চেনা
* ভয়েস রিকগনিশন
* Recommendation system

**✅ সংক্ষেপে: JavaScript দিয়ে কী কী করা যায়?**

| **কাজের ধরণ** | **কী কী করা যায়** |
| --- | --- |
| ওয়েব | ইন্টার‍্যাকটিভ ওয়েবসাইট, ফর্ম, ইভেন্ট |
| গেম | স্নেক গেম, কার্ড গেম |
| মোবাইল | React Native দিয়ে Android/iOS অ্যাপ |
| ডেস্কটপ | Electron.js দিয়ে অ্যাপ |
| সার্ভার | Node.js দিয়ে Backend |
| API | ডেটা আনা ও দেখানো |
| ML/AI | TensorFlow.js দিয়ে সহজ মেশিন লার্নিং |

**JavaScript-এর Reserved Keywords (সংরক্ষিত শব্দ)** গুলো দিচ্ছি — এগুলো এমন শব্দ যেগুলো JavaScript-এ আগে থেকেই নির্দিষ্ট কাজে ব্যবহৃত হয়, তাই তুমি এগুলো **ভেরিয়েবল, ফাংশনের নাম ইত্যাদি হিসেবে ব্যবহার করতে পারো না**।

## ✅ JavaScript Reserved Keywords List

### 🔹 **ভেরিয়েবল ও ডেটা টাইপ সম্পর্কিত**

javascript

CopyEdit

var

let

const

typeof

new

instanceof

void

### 🔹 **নিয়ন্ত্রণ কাঠামো (Control Structures)**

cpp

CopyEdit

if

else

switch

case

default

for

while

do

break

continue

return

### 🔹 **ফাংশন ও অবজেক্ট**

javascript

CopyEdit

function

class

this

super

constructor

extends

import

export

await

async

### 🔹 **বুলিয়ান ও লজিক্যাল**

javascript

CopyEdit

true

false

null

undefined

NaN

in

delete

### 🔹 **ব্যতিক্রম ও ত্রুটি (Errors & Try-Catch)**

csharp

CopyEdit

try

catch

finally

throw

### 🔹 **Scope ও Execution Context**

javascript

CopyEdit

with

yield

arguments

eval

### 🔹 **Future Reserved (ECMAScript এর জন্য রিজার্ভ রাখা)**

vbnet

CopyEdit

enum

implements

interface

package

private

protected

public

static

### ⚠️ বিশেষ দ্রষ্টব্য:

| **শব্দ** | **ব্যবহার** |
| --- | --- |
| let, const | ES6 থেকে শুরু |
| await, async | অ্যাসিনক্রোনাস কাজের জন্য |
| eval | জাভাস্ক্রিপ্ট কোড রান করায়, কিন্তু নিরাপদ নয় |
| enum, interface | ভবিষ্যতের জন্য রিজার্ভ, এখনো ব্যবহার করো না |

## ❌ ব্যবহার না করার নিয়ম:

javascript

CopyEdit

let if = 5; // ❌ ভুল

const return = "hi"; // ❌ ভুল

function var() {} // ❌ ভুল

➡️ এসব করলে কোড **এরর দিবে** কারণ এগুলো **জাভাস্ক্রিপ্ট নিজেই ব্যবহার করে**।

## 📦 Data Type কী?

**Data type** মানে হলো:  
➡️ "তথ্য বা ডেটা কোন ধরণের?"  
JavaScript-এ প্রতিটি ভেরিয়েবলের একেক ধরণের মান থাকতে পারে — সংখ্যা, লেখা, সত্য/মিথ্যা, ইত্যাদি।

## ✅ JavaScript-এ ২ ধরনের Data Type:

### 1. **Primitive Types** – সহজ, একক মান

### 2. **Non-Primitive (Reference) Types** – জটিল, অবজেক্ট বা কালেকশন টাইপ

## 🔹 1. Primitive Data Types (7টি)

| **টাইপ** | **উদাহরণ** | **ব্যাখ্যা** |
| --- | --- | --- |
| String | "Hello" | টেক্সট/লেখা |
| Number | 100, 3.14 | সংখ্যা (পূর্ণসংখ্যা বা দশমিক) |
| Boolean | true, false | সত্য/মিথ্যা |
| Null | null | ইচ্ছাকৃত খালি মান |
| Undefined | undefined | মান এখনো দেওয়া হয়নি |
| Symbol | Symbol("id") | ইউনিক মান তৈরি করতে |
| BigInt | 123456789012345678901n | বড় সংখ্যা রাখতে |

### 🧪 উদাহরণ:

javascript

CopyEdit

let name = "Rahim"; // String

let age = 25; // Number

let isStudent = true; // Boolean

let x = null; // Null

let y; // Undefined

let id = Symbol("user"); // Symbol

let bigNum = 123456789n; // BigInt

## 🔹 2. Non-Primitive (Reference) Data Types

| **টাইপ** | **উদাহরণ** | **ব্যাখ্যা** |
| --- | --- | --- |
| Object | {name: "Rahim", age: 20} | কী-ভ্যালু জোড়া |
| Array | [1, 2, 3] | উপাদান তালিকা |
| Function | function greet() {} | কাজের প্যাকেট |

### 🧪 উদাহরণ:

javascript

CopyEdit

let person = {

name: "Rahim",

age: 25

}; // Object

let numbers = [10, 20, 30]; // Array

function greet() {

console.log("Hello!");

} // Function

## 🧠 typeof দিয়ে চেক করা যায় টাইপ:

javascript

CopyEdit

typeof "Hello"; // "string"

typeof 100; // "number"

typeof true; // "boolean"

typeof undefined; // "undefined"

typeof null; // ❗️ "object" (bug, কিন্তু প্রচলিত)

typeof [1, 2]; // "object"

typeof function(){}; // "function"

## ✅ সংক্ষেপে মনে রাখো:

| **ধরন** | **টাইপ** |
| --- | --- |
| Primitive | string, number, boolean, null, undefined, symbol, bigint |
| Non-Primitive | object, array, function |

**JavaScript**-এ বিভিন্ন ধরণের **Operators (অপারেটর)** আছে যা দিয়ে আমরা ভেরিয়েবলের উপর গণনা, তুলনা, লজিক এবং অনেক কিছু করতে পারি।

চলো আমরা শিখি —  
👉 **কত প্রকার অপারেটর আছে?**  
👉 প্রতিটির **ব্যাখ্যা ও উদাহরণ** সহ।

**✅ JavaScript-এ অপারেটরের প্রকারভেদ (Types of Operators)**

| **অপারেটরের নাম** | **কাজ** |
| --- | --- |
| 1. Arithmetic Operators | গাণিতিক কাজ |
| 2. Assignment Operators | মান বসানো |
| 3. Comparison Operators | তুলনা করা |
| 4. Logical Operators | সত্য-মিথ্যা যাচাই |
| 5. String Operators | স্ট্রিং জোড়া লাগানো |
| 6. Conditional (Ternary) Operator | শর্টকাট শর্ত |
| 7. Type Operators | টাইপ চেক করা |
| 8. Bitwise Operators | বিট লেভেল অপারেশন |

**🔢 ১. Arithmetic Operators (গাণিতিক)**

javascript

CopyEdit

let a = 10, b = 3;

a + b // 13 (যোগ)

a - b // 7 (বিয়োগ)

a \* b // 30 (গুণ)

a / b // 3.33... (ভাগ)

a % b // 1 (ভাগশেষ)

a \*\* b // 1000 (ঘাত বা power)

**📝 ২. Assignment Operators (মান বসানো)**

javascript

CopyEdit

let x = 5; // =

x += 2; // x = x + 2

x -= 1; // x = x - 1

x \*= 3; // x = x \* 3

x /= 2; // x = x / 2

x %= 4; // x = x % 4

📌 ডানে যে মান আছে তা বাম পাশের ভেরিয়েবলে বসায়।

**⚖️ ৩. Comparison Operators (তুলনা করা)**

javascript

CopyEdit

5 == "5" // true (মান একই)

5 === "5" // false (টাইপ ভিন্ন)

5 != 3 // true (সমান না)

5 !== "5" // true (টাইপ ভিন্ন ও মানও না)

5 > 3 // true

5 < 3 // false

5 >= 5 // true

5 <= 4 // false

📌 == মান চেক করে  
📌 === মান ও টাইপ দুইটাই চেক করে

**🔍 ৪. Logical Operators (AND, OR, NOT)**

javascript

CopyEdit

true && true // true (AND - দুটি সত্য হলে সত্য)

true || false // true (OR - যেকোনো একটি সত্য হলেই সত্য)

!true // false (NOT - উল্টো করে দেয়)

📌 এগুলো শর্ত মিলাতে ব্যবহার হয় if বা while-এ

**🧵 ৫. String Operators (স্ট্রিং জোড়া)**

javascript

CopyEdit

let firstName = "Rahim";

let lastName = "Uddin";

let fullName = firstName + " " + lastName; // "Rahim Uddin"

📌 + দিয়ে দুটি টেক্সট (string) জোড়া যায়।

**❓ ৬. Conditional / Ternary Operator (ছোট if/else)**

javascript

CopyEdit

let age = 18;

let status = (age >= 18) ? "Adult" : "Minor";

console.log(status); // Adult

📌 উপরে বলা হয়:  
**"যদি age >= 18 হয়, তাহলে Adult নয়তো Minor"**

**🔎 ৭. Type Operators**

javascript

CopyEdit

typeof "Hello" // "string"

typeof 123 // "number"

instanceof Array // চেক করে কোনও ভেরিয়েবল array কিনা

📌 ডেটার ধরন বোঝার জন্য ব্যবহার হয়।

**⚙️ ৮. Bitwise Operators (বিট লেভেল গণনা – উন্নত বিষয়)**

javascript

CopyEdit

5 & 1 // 1 (AND)

5 | 1 // 5 (OR)

5 ^ 1 // 4 (XOR)

~5 // -6 (NOT)

5 << 1 // 10 (বাম সিফট)

5 >> 1 // 2 (ডান সিফট)

📌 সংখ্যাকে বাইনারি (0/1) তে রূপান্তর করে কাজ করে।

(এই অংশ জটিল, চাইলে আমি সহজ করে আলাদা করে বুঝিয়ে দিতে পারি।)

**🧠 সংক্ষেপে মনে রাখো:**

| **টাইপ** | **উদাহরণ** | **কাজে লাগে** |
| --- | --- | --- |
| Arithmetic | +, -, \*, /, % | সংখ্যা নিয়ে |
| Assignment | =, +=, -= | মান বসাতে |
| Comparison | ==, ===, > | তুলনায় |
| Logical | &&, |  |
| String | + | স্ট্রিং জোড়ায় |
| Ternary | ? : | ছোট if-else |
| Type | typeof | টাইপ বুঝতে |
| Bitwise | &, | , ^ |