

Assignment Operator (=)

◆ কেন Assignment operator?

Assign মানে → মান দিয়ে দেওয়া / বসিয়ে দেওয়া

= কোনো হিসাব করে না ❌

সে শুধু একটা মান ভেরিয়েবলের মধ্যে রাখে ✅

👉 উদাহরণ:

```
var c = 20;
```

কেন = কে Assignment operator বলে?

কারণ 👉 = হিসাব করে না,

👉 সে শুধু মানটা ভেরিয়েবলের মধ্যে “চুকিয়ে দেয় / বসিয়ে দেয়”।

ইংরেজি শব্দ **Assign** মানে:

| কাউকে কিছু দিয়ে দেওয়া / নির্ধারণ করে দেওয়া

এখন এই লাইনটা দেখো 👉

```
var c = 20;
```

এটার মানে কিন্তু ২০ সমান c না ❌

এটার মানে হলো 👉

| "২০ মানটা c নামের ভেরিয়েবলে দিয়ে দাও"

তাই `=` এখানে **assignment operator**।

কেন Arithmetic operator না?

Arithmetic operator হলে হিসাব করত:

`5 + 3`// হিসাব \rightarrow 8

`10 - 4`// হিসাব \rightarrow 6

কিন্তু `=` কোনো হিসাব করে না ❌

সে শুধু বলে:

এই মানটা ওখানে রাখো

খুব সহজ তুলনা 🧠

ধরো,

- একটা খালি বাক্স = `c`
- তুমি বাক্সে ২০ ঢুকিয়ে দিলে

`c = 20;`

এখানে `=` মানে 🖐️ বাক্সে ঢোকাও

এই জন্যই নাম **assignment operator**।

Mathematical = আর Programming = এক না !

গণিতে:

$5 = 5$ (সমান)

কিন্তু প্রোগ্রামিং-এ:

```
c = 20; // মান ঢোকানো
```

সমান চেক করতে চাইলে আলাদা operator আছে:

```
== // value compare
```

```
=== // value + type compare
```

এক লাইনে মনে রাখো 🔥

| = কে assignment operator বলে কারণ সে মান "assign" করে, সমান চেক করে না।

3 Compound Assignment Operator (`+=` , `=` , `=` , `/=` , `%=`)

👉 Arithmetic + Assignment একসাথে

◆ `+=` মানে কী?

```
c += d;
```

এর পুরো লেখা:

```
c = c + d;
```

👉 উদাহরণ:

```
var c = 20;
```

```
var d = 10;
```

```
c += d;
```

```
console.log(c);// 30
```

🧠 কম্পিউটার ভাবে:

"c-এর পুরাতন মান নাও,
তার সাথে d যোগ করো,
তারপর নতুন মানটা আবার c-তেই রেখে দাও"

✨ অন্য Compound Operators

Operator	পুরো মানে
<code>x += y</code>	<code>x = x + y</code>
<code>x -= y</code>	<code>x = x - y</code>
<code>x *= y</code>	<code>x = x * y</code>
<code>x /= y</code>	<code>x = x / y</code>
<code>x %= y</code>	<code>x = x % y</code>

5 Mathematical `=` vs Programming `=`

❌ গণিতে:

5 = 5 (সমান)

✅ প্রোগ্রামিং-এ:

`c = 20;` // মান assign

👉 সমান চেক করতে:

`==` // value check

`===` // value + type check