জাভাস্ক্রিপ্টে **অপারেটর**গুলি বিভিন্ন অপারেশন (যেমন গাণিতিক গণনা, তুলনা, লজিকাল অপারেশন) সম্পাদন করতে ব্যবহৃত হয়। জাভাস্ক্রিপ্টে বিভিন্ন ধরনের অপারেটর রয়েছে, যেমন গাণিতিক অপারেটর, তুলনা অপারেটর, লজিকাল অপারেটর, এসাইনমেন্ট অপারেটর ইত্যাদি। নিচে জাভাস্ক্রিপ্টের সমস্ত অপারেটরগুলির বিস্তারিত ব্যাখ্যা দেওয়া হলো:

**১. গাণিতিক অপারেটর (Arithmetic Operators):**

গাণিতিক অপারেটরগুলি সংখ্যার উপর গাণিতিক অপারেশন সম্পাদন করে।

| **অপারেটর** | **বর্ণনা** | **উদাহরণ** | **ফলাফল** |
| --- | --- | --- | --- |
| + | যোগ | 10 + 5 | 15 |
| - | বিয়োগ | 10 - 5 | 5 |
| \* | গুণ | 10 \* 5 | 50 |
| / | ভাগ | 10 / 5 | 2 |
| % | ভাগশেষ | 10 % 3 | 1 |
| \*\* | ঘাত (Power) | 2 \*\* 3 | 8 |
| ++ | ইনক্রিমেন্ট | let x = 5; x++ | 6 |
| -- | ডিক্রিমেন্ট | let x = 5; x-- | 4 |

**২. এসাইনমেন্ট অপারেটর (Assignment Operators):**

এসাইনমেন্ট অপারেটরগুলি ভেরিয়েবলে মান অ্যাসাইন করে।

| **অপারেটর** | **বর্ণনা** | **উদাহরণ** | **সমতুল্য** |
| --- | --- | --- | --- |
| = | মান অ্যাসাইন | let x = 10; | x = 10 |
| += | যোগ এবং অ্যাসাইন | x += 5; | x = x + 5 |
| -= | বিয়োগ এবং অ্যাসাইন | x -= 5; | x = x - 5 |
| \*= | গুণ এবং অ্যাসাইন | x \*= 5; | x = x \* 5 |
| /= | ভাগ এবং অ্যাসাইন | x /= 5; | x = x / 5 |
| %= | ভাগশেষ এবং অ্যাসাইন | x %= 5; | x = x % 5 |
| \*\*= | ঘাত এবং অ্যাসাইন | x \*\*= 2; | x = x \*\* 2 |

**৩. তুলনা অপারেটর (Comparison Operators):**

তুলনা অপারেটরগুলি দুটি মান তুলনা করে এবং true বা false রিটার্ন করে।

| **অপারেটর** | **বর্ণনা** | **উদাহরণ** | **ফলাফল** |
| --- | --- | --- | --- |
| == | সমান | 10 == "10" | true |
| === | স্ট্রিক্টলি সমান | 10 === "10" | false |
| != | অসমান | 10 != 5 | true |
| !== | স্ট্রিক্টলি অসমান | 10 !== "10" | true |
| > | বড় | 10 > 5 | true |
| < | ছোট | 10 < 5 | false |
| >= | বড় বা সমান | 10 >= 10 | true |
| <= | ছোট বা সমান | 10 <= 5 | false |

**৪. লজিকাল অপারেটর (Logical Operators):**

লজিকাল অপারেটরগুলি একাধিক শর্ত যুক্ত করতে ব্যবহৃত হয়।

| **অপারেটর** | **বর্ণনা** | **উদাহরণ** | **ফলাফল** |
| --- | --- | --- | --- |
| && | AND | true && false | false |
| ` |  | ` | OR | `true |  | false` | true |
| ! | NOT | !true | false |  |  |  |  |

**৫. কন্ডিশনাল (টার্নারি) অপারেটর (Ternary Operator):**

এটি একটি শর্টহ্যান্ড if-else স্টেটমেন্ট।

**সিনট্যাক্স:**

javascript

Copy

condition ? expression1 : expression2;

**উদাহরণ:**

javascript

Copy

let age = 20;

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

console.log(status); // আউটপুট: Adult

**৬. টাইপ অপারেটর (Type Operators):**

টাইপ অপারেটরগুলি ডেটা টাইপ চেক করতে ব্যবহৃত হয়।

| **অপারেটর** | **বর্ণনা** | **উদাহরণ** | **ফলাফল** |
| --- | --- | --- | --- |
| typeof | ভেরিয়েবলের টাইপ রিটার্ন করে | typeof "Hello" | "string" |
| instanceof | অবজেক্টের টাইপ চেক করে | [] instanceof Array | true |

**৭. বিটওয়াইজ অপারেটর (Bitwise Operators):**

বিটওয়াইজ অপারেটরগুলি সংখ্যার বাইনারি ফর্মে অপারেশন সম্পাদন করে।

| **অপারেটর** | **বর্ণনা** | **উদাহরণ** | **ফলাফল** |
| --- | --- | --- | --- |
| & | AND | 5 & 1 | 1 |
| ` | ` | OR | `5 | 1` | 5 |
| ^ | XOR | 5 ^ 1 | 4 |  |  |
| ~ | NOT | ~5 | -6 |  |  |
| << | লেফট শিফট | 5 << 1 | 10 |  |  |
| >> | রাইট শিফট | 5 >> 1 | 2 |  |  |
| >>> | আনসাইনড রাইট শিফট | -5 >>> 1 | 2147483645 |  |  |

**৮. স্ট্রিং অপারেটর (String Operators):**

স্ট্রিং অপারেটরগুলি স্ট্রিং কনক্যাটেনেশন করতে ব্যবহৃত হয়।

| **অপারেটর** | **বর্ণনা** | **উদাহরণ** | **ফলাফল** |
| --- | --- | --- | --- |
| + | স্ট্রিং কনক্যাটেনেশন | "Hello" + " World" | "Hello World" |

**৯. কোমা অপারেটর (Comma Operator):**

কোমা অপারেটর একাধিক এক্সপ্রেশন একসাথে এক্সিকিউট করে এবং শেষ এক্সপ্রেশনের মান রিটার্ন করে।

**উদাহরণ:**

javascript

Copy

let x = (5, 10); // x = 10

**১০. ডিলিট অপারেটর (Delete Operator):**

ডিলিট অপারেটর অবজেক্টের প্রপার্টি ডিলিট করে।

**উদাহরণ:**

javascript

Copy

let person = { name: "John", age: 25 };

delete person.age; // age প্রপার্টি ডিলিট হয়

console.log(person); // আউটপুট: { name: "John" }

**সারসংক্ষেপ:**

জাভাস্ক্রিপ্টে বিভিন্ন ধরনের অপারেটর রয়েছে যা বিভিন্ন অপারেশন সম্পাদন করতে ব্যবহৃত হয়। নিচে অপারেটরগুলির একটি সংক্ষিপ্ত তালিকা দেওয়া হলো:

| **অপারেটর টাইপ** | **উদাহরণ অপারেটর** | **ব্যবহার** |
| --- | --- | --- |
| গাণিতিক অপারেটর | +, -, \*, / | গাণিতিক গণনা |
| এসাইনমেন্ট অপারেটর | =, +=, -= | ভেরিয়েবলে মান অ্যাসাইন |
| তুলনা অপারেটর | ==, ===, >, < | দুটি মান তুলনা |
| লজিকাল অপারেটর | &&, ` |  | , !` | লজিকাল শর্ত যুক্ত করা |
| কন্ডিশনাল অপারেটর | ? : | শর্টহ্যান্ড if-else |  |  |
| টাইপ অপারেটর | typeof, instanceof | ডেটা টাইপ চেক করা |  |  |
| বিটওয়াইজ অপারেটর | &, ` | , ^, ~` | বাইনারি অপারেশন |  |
| স্ট্রিং অপারেটর | + | স্ট্রিং কনক্যাটেনেশন |  |  |
| কোমা অপারেটর | , | একাধিক এক্সপ্রেশন একসাথে এক্সিকিউট করা |  |  |
| ডিলিট অপারেটর | delete | অবজেক্টের প্রপার্টি ডিলিট করা |  |  |

এই অপারেটরগুলি ব্যবহার করে আপনি জাভাস্ক্রিপ্টে বিভিন্ন ধরনের অপারেশন সম্পাদন করতে পারবেন। 😊

**instanceof**

জাভাস্ক্রিপ্টে **instanceof অপারেটর** ব্যবহার করে আপনি চেক করতে পারেন যে একটি অবজেক্ট নির্দিষ্ট ক্লাস বা কনস্ট্রাক্টরের একটি ইনস্ট্যান্স কিনা। এটি একটি বুলিয়ান মান (true বা false) রিটার্ন করে।

**সিনট্যাক্স:**

javascript

Copy

object instanceof constructor;

* object: যে অবজেক্টটি চেক করা হবে।
* constructor: যে ক্লাস বা কনস্ট্রাক্টরের ইনস্ট্যান্স চেক করা হবে।

**উদাহরণ:**

**১. বিল্ট-ইন অবজেক্ট চেক করা:**

javascript

Copy

let arr = [1, 2, 3];

console.log(arr instanceof Array); // true

console.log(arr instanceof Object); // true (Array হল Object এর একটি ইনস্ট্যান্স)

console.log(arr instanceof String); // false

let str = "Hello";

console.log(str instanceof String); // false (প্রিমিটিভ স্ট্রিং)

console.log(new String("Hello") instanceof String); // true (অবজেক্ট স্ট্রিং)

**২. কাস্টম ক্লাস চেক করা:**

javascript

Copy

class Car {

constructor(brand) {

this.brand = brand;

}

}

let myCar = new Car("Toyota");

console.log(myCar instanceof Car); // true

console.log(myCar instanceof Object); // true (Car হল Object এর একটি ইনস্ট্যান্স)

**৩. ইনহেরিটেন্স চেক করা:**

javascript

Copy

class Vehicle {

constructor(type) {

this.type = type;

}

}

class Car extends Vehicle {

constructor(brand) {

super("Car");

this.brand = brand;

}

}

let myCar = new Car("Toyota");

console.log(myCar instanceof Car); // true

console.log(myCar instanceof Vehicle); // true (Car হল Vehicle এর একটি ইনস্ট্যান্স)

console.log(myCar instanceof Object); // true (Vehicle হল Object এর একটি ইনস্ট্যান্স)

**৪. ফাংশন কনস্ট্রাক্টর চেক করা:**

javascript

Copy

function Person(name) {

this.name = name;

}

let john = new Person("John");

console.log(john instanceof Person); // true

console.log(john instanceof Object); // true

**কাজের নিয়ম:**

* instanceof অপারেটর প্রোটোটাইপ চেইন চেক করে। যদি অবজেক্টটি কনস্ট্রাক্টরের প্রোটোটাইপ চেইনে থাকে, তাহলে true রিটার্ন করে।
* এটি শুধুমাত্র অবজেক্টের জন্য কাজ করে। প্রিমিটিভ ডেটা টাইপ (যেমন string, number, boolean) এর জন্য false রিটার্ন করে।

**সারসংক্ষেপ:**

* instanceof অপারেটর ব্যবহার করে আপনি চেক করতে পারেন যে একটি অবজেক্ট নির্দিষ্ট ক্লাস বা কনস্ট্রাক্টরের একটি ইনস্ট্যান্স কিনা।
* এটি বিল্ট-ইন অবজেক্ট (যেমন Array, Object), কাস্টম ক্লাস, এবং ইনহেরিটেন্স চেক করার জন্য ব্যবহৃত হয়।

এই অপারেটরটি ব্যবহার করে আপনি অবজেক্টের টাইপ চেক করতে পারবেন এবং আপনার কোডে টাইপ সেফটি নিশ্চিত করতে পারবেন। 😊