

# Ternary operator

Sure! Let's explore the **Ternary Operator** in JavaScript in a detailed and beginner-friendly way.

---

## ? What is the Ternary Operator?

The **Ternary Operator** is a shorthand for `if...else` statements.

### Syntax:

```
condition ? expressionIfTrue : expressionIfFalse;
```

It's called "**ternary**" because it takes **three operands**:

1. A **condition**
  2. An expression to execute **if true**
  3. An expression to execute **if false**
- 

### Simple Example:

```
let age = 18;  
let message = (age >= 18) ? "You are an adult" : "You are a minor";  
console.log(message); // Output: You are an adult
```

## How It Works

Let's break down this example:

```
(condition) ? (if true) : (if false);
```

In our case:

```
(age >= 18) ? "You are an adult" : "You are a minor";
```

- If `age` is **18 or more**, the condition is true → "You are an adult"
- Otherwise → "You are a minor"

## Equivalent `if...else` :

```
let message;  
if (age >= 18) {  
    message = "You are an adult";  
} else {  
    message = "You are a minor";  
}
```

## More Examples

### Even or Odd

```
let num = 5;  
let result = (num % 2 === 0) ? "Even" : "Odd";  
console.log(result); // Output: Odd
```

### Login Check

```
let isLoggedIn = true;  
let message = isLoggedIn ? "Welcome back!" : "Please log in.";  
console.log(message); // Output: Welcome back!
```

## Nested Ternary (Use with Caution)

You can nest ternary operators, but too much nesting can reduce readability.

```
let marks = 85;  
let grade = (marks >= 90) ? "A" :  
    (marks >= 75) ? "B" :  
    (marks >= 60) ? "C" : "F";  
console.log(grade); // Output: B
```

## 🚫 When NOT to Use Ternary

Avoid using ternary if:

- The logic is complex
- You're nesting multiple levels

Prefer `if...else` for better readability in those cases.

## 📌 Summary

Feature	Description
Syntax	<code>condition ? trueResult : falseResult</code>
Use case	Replace short <code>if...else</code>
Output	Returns a value
Best for	Simple, single-line conditional expressions

Let me know if you'd like a **practice challenge** or want to see how this can be used in JSX (React)!