



## Answersheet

Name: Arashad Ahamad

No. of questions attempted: 7

Submitted at: 4 Apr, 2025 11:59 AM

Assignment: Ternary Operator Explained

Total no. of questions: 7

Total marks: 35

Result: **Not reviewed yet**

Sl. No.	Question	Actions
1	<p>Here is the solution to the question in previous chapters just like we promised. Confusing right?</p> <p>Basically, we are checking if a number is largest in each of the statements. So in this case we already checked that variable a is not largest so it is either b or c and that's the logic behind this.</p> <pre>const a = 5 const b = 8 const c = 6 if (a &gt; b &amp;&amp; a &gt; c) {   console.log(a) } else if (b &gt; c) {   console.log(b) } else {</pre>	<p>5 marks</p> <p>Marks</p> <input type="text" value="0"/>



***console.log(c)***

***}***

**⇒ Now, use this code and try to do the same thing with the ternary operator.**

```
const a = 5 const b = 8 const c = 6 const largestNum = a >= b && a >= c ? a : b >= a && b >= c ? b : c console.log(largestNum)
```

2

**Here is the solution to the question above.**

5 marks

Marks

***const a = 5***

***const b = 8***

***const c = 6***

***console.log(a > b && a > c ? a : b > c ? b : c)***

**⇒ Now, on the line of the console what are the expressions and statements?**

8

3

**What is the fundamental difference between ternary operator and if else statements?**

5 marks

Marks

The ternary operator is a shorthand version of if-else. Both perform the same function—checking conditions—but: 1. The ternary operator allows writing conditions in a single line, making the code concise and clean. 2. The if-else statement is longer but improves readability, especially for complex logic.



4	<p><b>Can you explain why expressions can appear within statements but statements cannot appear within expressions in JavaScript?</b> 5 marks</p> <p>In JavaScript, expressions are those that return a value, whereas statements are those that perform an action. • Expressions can be written inside statements because they produce a value. Example: <code>let result = 10 + 5; // "10 + 5" is an expression that returns 15.</code> <code>console.log(result); // Here, the expression is inside a statement.</code> • Statements cannot be written inside expressions because statements do not return a value; they only perform operations. Example (Invalid): <code>let x = if (true) { console.log("Hello"); }; // Error, because if statement does not return a value.</code> Conclusion: • Expressions always return a value, so they can appear inside statements. • Statements only perform actions, so they cannot be placed inside expressions.</p>	Marks <input type="text" value="0"/>
5	<p><b>Can you give examples of different types of expressions in JavaScript, such as arithmetic expressions and logical expressions?</b> 5 marks</p> <p>• <code>10 &gt; 15</code> • <code>10 &lt;= 20</code> • <code>10 === 10</code> • <code>'hello' + 'world'</code> • <code>true &amp;&amp; false</code> • <code>10 + 5</code></p>	Marks <input type="text" value="0"/>
6	<p><b>What is a statement in JavaScript, and can you provide some examples?</b> 5 marks</p> <p>A statement performs an action in JavaScript. <code>let a = 9; // Variable declaration (action: assigns 9 to 'a')</code> <code>console.log(a); // Prints 'a' (action: outputs 9 to console)</code> <code>if (a &gt; 5) { console.log("Big number"); } // Conditional statement (action: checks condition and prints if true)</code></p>	Marks <input type="text" value="0"/>
7	<p><b>What is an expression in JavaScript, and how does it differ from a statement?</b> 5 marks</p>	Marks <input type="text" value="0"/>



An expression is a piece of code that returns a value. Key Differences Between Expression and Statement:

- Expression: Returns a value (e.g.,  $10 + 5 \rightarrow 15$ ).
- Statement: Performs an action (e.g., `let a = 10;`  $\rightarrow$  Assigns value).
- Expression: Can be inside a statement.
- Statement: Cannot be inside an expression.

0