

1) logical and (&&) , logical or (||)

Sol. Logical And(\$\$):

Logical AND (&&)expr1 && expr2 Returns expr1 if it can be converted to false; otherwise, returns expr2. Thus, when used with Boolean values, && returns true if both operands are true; otherwise, returns false.

Logical OR (||) expr1 || expr2 Returns expr1 if it can be converted to true; otherwise, returns expr2. Thus, when used with Boolean values, || returns true if either operand is true.

2) thruthy and falsy values in javascript

Sol: thruthy: In JavaScript, a thruthy value is a value that translates to true when evaluated in a Boolean context. All values are thruthy unless they are defined as falsy (i.e., except for false, 0, "", null, undefined, and NaN).

falsy:

A falsy value is a value that translates to false when evaluated in a Boolean context.

3) operators in java script

Sol: **Arithmetic operators:**

| | |
|----|---------------------|
| + | Addition |
| - | Subtraction |
| * | Multiplication |
| / | Division |
| % | Modulus (Remainder) |
| ++ | Increment |
| -- | Decrement |

Comparison operators:

| | |
|-------|------------------------------------|
| = = | Is equal to |
| = = = | Identical (equal and of same type) |
| != | Not equal to 1 |

!== Not Identical
> Greater than
>= Greater than or equal to
< Less than
<= Less than or equal to

Assignment operators:

= Assign
+= Add and assign
-= Subtract and assign
*= Multiply and assign
/= Divide and assign
%= Modulus and assign

Bitwise operators:

& Bitwise AND
| Bitwise OR
^ Bitwise XOR
~ Bitwise NOT
<< Bitwise Left Shift
>> Bitwise Right Shift
>>> Bitwise Right Shift with Zero

Logical operators:

&& Logical AND
|| Logical OR
! Logical Not

Special Operators:

(?:) Conditional Operator returns value based on the condition. It is like if-else.
, Comma Operator allows multiple expressions to be evaluated as single statement.
delete Delete Operator deletes a property from the object.
in In Operator checks if object has the given property
instanceof checks if the object is an instance of given type
new creates an instance (object)
typeof checks the type of object.
void it discards the expression's return value.

yield checks what is returned in a generator by the generator's iterator.

4) operational precedences.

Sol. Operator precedence determines the order in which operators are evaluated. Operators with higher precedence are evaluated first.

| | |
|--------|----------------------------------|
| () | brackets |
| ++ - - | increment/ decrement |
| * / % | Multiplication, Division, Modulo |
| + _ | Addition, Subtraction |