LOGICAL OPERATORS

IN THIS LESSON

Logical Operators

Logical And

Logical Or

Logical Not

Double Not

LOGICAL OPERATORS

3 logical operators in JavaScript   && || !

&& and **||** are binary: x && y   x || y

! is unary: !x

LOGICAL AND: &&

A theater wants to have both a general admission ticket price of $20 for ages over 12 and under 65, and a discounted ticket price of $10 for ages 12 and under, or 65 and over

How would you write this logic using an if statement? Pause this video now and try to write an if statement that would work for this, based on what you have learned so far.

LOGICAL AND: &&

```
if (age > 12) {
    if (age < 65) {
        ticketPrice = 20;
    } else {
        ticketPrice = 10;
    }
} else {
    ticketPrice = 10;
}
```

```
if ((age > 12) && (age < 65)) {
    ticketPrice = 20;
} else {
    ticketPrice = 10;
}
```

nucamp

LOGICAL AND: &&

expression	return value
true && true	true
true && false	false
false && true	false
false && false	false

&& tests if the left operand is truthy or falsy

If falsy, it will return the left operand's value

If truthy, it will return the right operand's value

&& returns the last truthy value if **both** operands are truthy; if **either** operand is falsy, then **&&** returns the first falsy value

LOGICAL AND: &&

expression	return value
12 && 'cheetah'	'cheetah'
'cheetah' && 12	12
12 && null	null
undefined && 'cheetah'	undefined
"" && null	""
null && undefined	null

The 6 falsy values: false, null, undefined, empty string, NaN, 0

Each expression returns the value of the first falsy operand

If neither operand is falsy, it returns the last truthy value the right operand)

LOGICAL OR: ||

Your friends are forming a band and you want to join. They tell you that you can join if you play an instrument, **OR** if you can sing

You don't have to both play an instrument **AND** be able to sing - fulfilling either condition makes you eligible to join the band

This is the basic idea of the || operator

LOGICAL OR: ||

expression	return value
true || true	true
true || false	true
false || true	true
false || false	false

|| tests if the left operand is truthy or falsy

If truthy, it will return the left operand's value

If falsy, it will return the right operand's value

|| returns the first truthy value; if neither operand is truthy,
then it will return the last falsy value

COMPARISON OF && AND ||

expression	return value
true && true	true
true && false	false
false && true	false
false && false	false

<table>
<tr><th>expression</th><th>return value</th></tr>
<tr><td>true || true</td><td>true</td></tr>
<tr><td>true || false</td><td>true</td></tr>
<tr><td>false || true</td><td>true</td></tr>
<tr><td>false || false</td><td>false</td></tr>
</table>

COMPARISON OF && AND ||

expression	return value
12 && 'cheetah'	'cheetah'
'cheetah' && 12	12
12 && null	null
undefined && 'cheetah'	undefined
"" && null	""
null && undefined	null

expression	return value
12 || 'cheetah'	12
'cheetah' || 12	'cheetah'
12 || null	12
undefined || 'cheetah'	'cheetah'
"" || null	null
null || undefined	undefined

LOGICAL NOT: !

expression	return value
!true	false
!false	true
!'cat'	false
!(3 > 2)	false
!'nucamp' === true	false
!('nucamp' === true)	true

! is a unary operator (meaning it has one operand) and it is placed just before the operand, no space

Always returns true or false: returns false when its operand's value is truthy, true when its operand's value is falsy

! type coerces its operand to a Boolean value then negates that value

Use parentheses around the operand to ensure **!** operates on the entire operand and not the first value in it

THE DOUBLE NOT OPERATOR: !!

Not a new operator, but a clever use of the ! operator as a shorthand for Boolean conversion

Built-in JavaScript function Boolean() can be used to convert values to Boolean data type:

Boolean(123)	true

Or, use first ! to coerce a value (or expression) to Boolean & negate it:

!123	false

Then use second ! to reverse the negation:

!!123	true