COMPARISON OPERATORS

IN THIS LESSON

Comparison operators

Equality operators

Relational operators

COMPARISON OPERATORS

All comparison operators are binary operators

They compare two values and return Boolean true or false

Two types: equality and relational

EQUALITY OPERATORS

===	strict equality/identity/triple equals operator
==	loose equality/equality/double equals operator
!==	strict inequality/nonidentity operator
!=	loose inequality/inequality operator

EQUALITY OPERATORS

===	strict equality/identity/triple equals operator
==	loose equality/equality/double equals operator

Strict equality operator compares operands' value and type

Loose equality operator performs type coercion when types do not match

- type coercion: implicit type conversion, type converted without explicit request

5 === '5'	false	5 == '5'	true
null === undefined	false	null == undefined	true

EQUALITY OPERATORS

!==	strict inequality/nonidentity operator
!=	loose inequality/inequality operator

Strict inequality operator compares type and value

Loose inquality operator performs type coercion before comparing value

5 !== '5'	true	5 != '5'	false

Generally, always use strict versions: **===** and **!==**

RELATIONAL OPERATORS

>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

RELATIONAL OPERATORS

With number operands, relational operators work as you would expect:

5 > 10	false
x = 23; x <= 50	true

String operands are compared in lexicographical order, where a is lower/lesser & z is higher/greater; if characters match, compare next character

'a' <= 'z'	true
'peony' > 'oboe'	true
'aardvark' > 'anteater'	false
'eefffgg' >= 'eefffgh'	false