Data type for computing with true and false: `boolean`

`boolean` data type

values	true	false	
literals	`true`	`false`	
operations	and	or	not
operator	`&&`	`||`	`!`

Truth-table definitions

a	!a
true	false
false	true

a	b	a && b	a || b
false	false	false	false
false	true	false	true
true	false	false	true
true	true	true	true

Recall first lecture

Q. a XOR b?

A. `(!a && b) || (a && !b)`

Proof

a	b	!a && b	a && !b	(!a && b) || (a && !b)
false	false	false	false	false
false	true	true	false	true
true	false	false	true	true
true	true	false	false	false

Typical usage: Control logic and flow of a program (stay tuned).

Comparison operators

Fundamental operations that are defined for each primitive type allow us to *compare* values.

- Operands: two expressions of the same type.
- Result: a value of type `boolean`.

operator	*meaning*	`true`	`false`
`==`	equal	`2 == 2`	`2 == 3`
`!=`	not equal	`3 != 2`	`2 != 2`
`<`	less than	`2 < 13`	`2 < 2`
`<=`	less than or equal	`2 <= 2`	`3 <= 2`
`>`	greater than	`13 > 2`	`2 < 13`
`>=`	greater than or equal	`3 >= 2`	`2 >= 3`

Examples

non-negative discriminant?	`( b*b - 4.0*a*c ) >= 0.0`
beginning of a century?	`( year % 100 ) == 0`
legal month?	`( month >= 1 ) && ( month <= 12 )`

Typical double values are *approximations* so beware of == comparisons