Booleans - True or false?

True or False?

```
bool isWaterBlue = true;
bool isWaterRed = false;
```

A boolean (**bool**) are used to display values that can be:

- **Yes** or **No**
- **On** or **Off**
- **True** or **False**

Logical Operators

```
bool isWaterNotRed = !isWaterRed; // true
```

NOT

You can inverse a booleans value by using an exclamation point ! (read it as **NOT**)

- **!true** -> **NOT true** -> **false**
- **!false** -> **NOT false** -> **true**

AND

You can combine two **bool** values with **&&** (read: AND) or || (read: OR).

```
bool isOfFullAge = true;
bool hasEnoughMoney = true;
// You are only allowed, if you are of full age AND you have enough money:
bool mayBuyBeer = isOfFullAge && hasEnoughMoney; // true
```

<- **&&** only returns **true** if **both** values are **true**.

```
bool hasEnoughMoney = false;
bool canGetALoan = true;
bool mayBuyBeer = hasEnoughMoney || canGetALoan; // true
```

<- || returns **true** if at least **one** of the incoming values are **true**.

Negation

```
!(!A) = A
```

```
!(A && B) = !A || !B
```

```
!(A || B) = !A && !B
```

Negating logical operators:

- **NOT -> !(!A) = A**
  - **The negation of “Not allowed to play” is**
  - **“Allowed to play**
- **AND -> !(A && B) = !A || !B**
  - **The negation of “Touches Ground” AND “Press Jump” is**
  - **“Does not touch ground” OR “Does not press jump”**
- **OR -> !(A || B) = !A && !B**
  - **The negation of “Has ID” OR “Has Passport” is**
  - **“Has No ID” AND “Has No Passport”**

Toggle

```
bool playerTurn = true;
// ...
playerTurn = !playerTurn;
// ...
playerTurn = !playerTurn;
```

You can use ! to toggle a **bool**´s value between the two states:

- For a light switch -> On or Off
- A checkbox in a menu -> Ticked or Not
- Or who’s turn it is -> Player´s or AI´s turn
- or just about any scenario where something can have two states..

Comparison Operators

Comparison operators are used to compare two values and return true or false.

- bool isGreater = 10 > 9; // true
- bool isEqual = 10 == 9; // false
- bool isNotEqual = 10 != 9; // true
- bool isLess = 10 < 9; // false
- bool isGreaterOrEqual = 10 >= 9; // true
- bool isLessOrEqual = 10 <= 9; // false

Or you can even check if something is in between two values by making use of the AND to do two comparisons:

- bool isMorning = time > 5 && time < 12;

Negation of Comparison operators

Input Gold	Gold > 100	Gold < 100	Gold <= 100
50	❌	✅	✅
100	❌	❌	✅
150	✅	❌	❌

You can simplify the negation of comparison operators:

- **!(money > 100)** is same as **money <= 100**
- **!(money < 100)** is same as **money >= 100;**
- **!(money == 100)** is same as **money != 100;**

Think about it:

- What is the opposite of having more than 100 gold?
- It's not having less than 100 gold.
- It's having less than or exactly 100 gold;

You can read the both ways, because the opposite of the opposite of a statement is the original statement:

- **!(!(A))** is the same as **A**
- **!(!(true)) = !(false) = true**

Combining both

Negation:

- bool canBuyBeer = age >= 20 && money > 50;
- bool canNotBuyBeer = !(age >= 20 && money > 50);
- bool canNotBuyBerr = age < 20 || money <= 50;

Simplification:

- bool jump = (canJump && pressSpace) || (canJump and click);
- bool jump = canJump && (pressSpace || click);
- bool canBuy = hasMoney || (!hasMoney && canTakeLoan);
- bool canBuy = hasMoney || canTakeLoan);

Goal

```
Output:What's your age?
Input:31
Output:You are a child: False
Output:You are a teenager: False
Output:You are an adult: True
```

- Create a console project called **E10Boolean**
- Ask the user for their age, save it to a variable named **age**
- First, do a few age checks:
  - Save a bool variable named **isChild**, whether the age is **between** 0 and 12
  - Save a bool variable named **isTeenager**, whether the age is **between** 13 and 19
  - Save a bool variable named **isAdult**, whether the age is **greater** than 19
- Then print them all to the console like this:
  - **You are a child: true**
  - etc.. see sample on the left