# Lesson 6: Boolean Expressions

Download, unzip, and move the Lesson6 folder to your Desktop. Remember to cd into your Desktop, and then the folder.

# But first... Task #2 from last time

- Open the lesson 5 folder in Atom
- cd into your Lesson5 directory
- Take 8 minutes
    - We'll walk through the solution after
- What Math function will help us solve this?
- What can we use to determine the structure of the output? (ie line, square, cube).

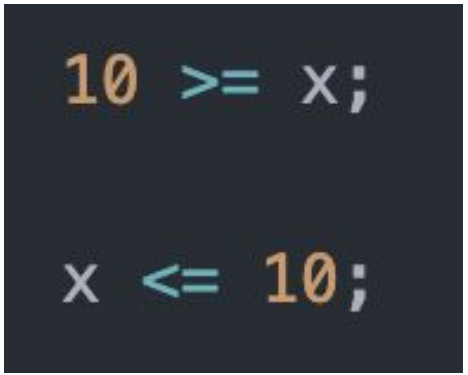
# Boolean Comparators

- We've already seen these, think back to (x>5)... we're comparing!
- There are a bunch, but we are really on ever testing 2 things:
  - If two values are equal
  - If one is greater than the other
- When using greater/less than OR equal to, the = always comes second!
- 3 "=" checks for type as well as value

| Operator | Description | Example |
|---|---|---|
| == | Equal to | x == 5 |
| === | Equal value and type | x === '5' |
| != | Not equal to | x != 55 |
| !== | Not equal to value and type | x !== '5' |
| > | Greater than | x > 1 |
| < | Less than | x < 10 |
| >= | Greater than or equal to | x >= 5 |
| <= | Less than or equal to | x <= 5 |

# Comparisons are often symmetrical!

- Spot the difference between the two conditions on the right
- Does it make the code behave any differently?
  - This really halves the number of operators if you think about it
- When comparing values for equality, remember to use 2 equals signs (==)!
  - 1 = means you are setting a value
  - 2 means you are comparing values
  - 3 means you are comparing value AND type



# Task #1

- Open up Lesson6/main.js in Atom
  - Get it ready to run in your terminal
- Read the instructions carefully
- Take 10 minutes to solve the problem
  - Send any questions in the chat

# Boolean Operators

- Arithmetic operators perform operations on numbers
- Boolean operators perform operations on boolean values!
  - In your computer, logic gates perform these processes on a physical level
- They allow us to make decisions when given answers to many YES or NO questions

**AND**

**OR**

**NOT**

# AND Operator

- We saw this with the rectangle examples last week
  - Used if we only want to proceed when both conditions are true
- In code, we use && to mean AND
- Think of this just like a normal math operator
  - We take two inputs and produce a single output, either true or false

```
let x = (5>4)&&(6>5);
//true
let y = (5<4)&&(6>5);
//false
let z = true&&false;
//false
```

# OR Operator

- Very similar to the AND operator, except this time the whole expression will be true at long as one or more of the sub-expressions are true
- This works logically if we say it out loud
  - “I want to do this if either this OR that is true”
- In Javascript, we denote this operator with ||
- See the examples to the right for how we use the operator

```
let a = (5>4)||(6>5);
//true
let b = (5<4)||(6>5);
//true
let c = (5<4)||(6<5);
//false
let d = true||false;
//true
let d = false||false;
//false
```

# NOT Operator

- This one is the easiest
- It simply inverts the boolean value of whatever it precedes
- Denoted by ! - standard across languages
- Very commonly used
- They can get tricky when used with larger expressions (see right)

```
let f = !false;
//true
let g = !true;
//false
let h = !(4<5)||(6<5);
//??
```

# Task #2

- Scroll down to Task #2 in main.js
- Read the instructions carefully, and begin
- Take 10 minutes
    - Throw any questions into the chat

# Review

Today we learned:

- Boolean Comparators
  - <, >, >=, <=, ==
- Boolean Operators
  - &&, ||, !
- How to evaluate boolean expressions

# Wrap-Up

Email questions to info.codedelaware@gmail.com

Next class: Next Monday, 1/18

Congrats! You're halfway through the course!