
TECH
ELEVATOR
Module 1, Week 1

What is one word you’d use to describe the past week?

Housekeeping

- Strongly encourage getting through all reading/quiz material **before Sat** (even though exercises are due Sunday).
- Expectation is that you're spending 2+ hours a day.
- If you complete the exercises early in the week, circle back and go over your work to reinforce the concepts. Or slack me and I'll be able to give you tasks that can reinforce the weeks material or prepare you for upcoming material.
- Exercises
  - Awesome to get 100% of tests passing, but remember that 50% (a score of 2) and 90% (a score of 3) are great, too
  - You can submit multiple times

Housekeeping

- Office hours/student time
- Pushed week-1 starting lecture code if you want to follow along.
- Sign up for a 1 on 1 with me
- Check out Reign
  - https://reign.techelevator.com/
  - Password TechElevatorStudent

Questions from week 0

Any questions on...

- How to get help
- IntelliJ (how to open a project, run test code, etc.)
- Git
- Navigating the file system
- Using the command line
- Finding reading/quiz content in the LMS or code in GitLab
- Any other topics?

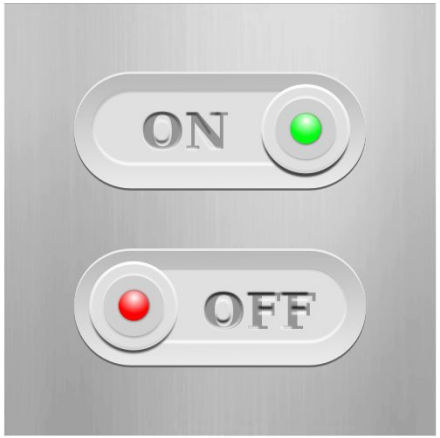
Agenda

- Learning objectives
  - Opening project/ new project
  - Declare and assign values to variables
  - Write arithmetic expressions and understand how they are evaluated in code
  - Identify the right primitive data types to model data in a real-world domain
  - Understand and use casting
- Show sample app
- Introduce pair programming

Data types - Boolean

On/off switch
Values: true or false
Examples:

- Is the feature enabled?
- Is the user above the age of 13?



Data types - int, long, short, byte

Store numeric data (integers)
Values: depends on type
Examples:

- 1, 900, -4893

int is by far the most common

Data types - float, double

Store numeric data with decimal points
Values: depends on type
Examples:

- 1.0, 3.333333

Double is more common than float

Data types - char vs. String

char

- Stores a Unicode character (letter, punctuation, symbols)
- Use single quotes for a char literal
- Primitive data type
- Examples: ‘B’, ‘f’, ‘{’

String

- Stores text
- Use double quotes for a string literal
- Not a primitive data type, but very common
- Examples: “The cow jumped over the moon!”, “gobbledy-gook”

Brainstorming

What are some examples of everyday computer interactions that depend on arithmetic?

- adding up the total price of your Amazon shopping cart
- counting how many followers someone has on Twitter
- determining the remaining runtime of a YouTube video
- statistics displayed about weather or sports
- calculating the position of moving objects in a video game

Code Examples

Breakout rooms:

- I'll post questions in Chat
- Talk about answers together* and jot them down
  - Write the expression
  - Write down what you think it will evaluate to, and why
- We'll come back together and test them out

*Make sure everyone has a chance to think about the answers individually

Example question: What is 4 divided by 2?
Answer: expression is `4 / 2`, which evaluates to `2`

Agenda

- Learning objectives
  - Build boolean expressions using arithmetic, comparison, and logical operators
  - Write conditional code using boolean expressions
  - Read and understand complex code expressions
  - Understand the use and implications of code blocks and variable scope
- Show sample app
- Introduce pair programming

Comparison operators

Operator	Meaning
==	Equal To
!=	Not Equal To
>	Greater Than
<	Less Than
>=	Greater Than or Equal To
<=	Less Than or Equal To

When used in an expression, these comparison operators always evaluate to a boolean `true` or `false` value

Logical operators

Operator	Meaning
!	NOT
&&	AND
||	OR
^	XOR

Individually

A	B	A && B	A || B	!A && !B
TRUE	TRUE			
TRUE	FALSE			
FALSE	TRUE			
FALSE	FALSE			

Small groups

- Breakout rooms of 3-4 people each
- Make a copy of the slide I'll share in Slack
- One person, share your screen
- Fill in the truth table together for each expression
- Take note of any items that tripped you up

A	B	!(A || B)	!A || !B	A ^ !B	(A && B) || !A
TRUE	TRUE				
TRUE	FALSE				
FALSE	TRUE				
FALSE	FALSE				

A	B	!(A || B)	!A || !B	A ^ !B	(A && B) || !A
TRUE	**TRUE**	FALSE	FALSE	TRUE	TRUE
TRUE	**FALSE**	FALSE	TRUE	FALSE	FALSE
FALSE	**TRUE**	FALSE	TRUE	FALSE	TRUE
FALSE	**FALSE**	TRUE	TRUE	TRUE	TRUE

Code examples

- Comparison operators
- Putting together logical and comparison operators to form expressions

Agenda

- Learning objectives
  - Write arithmetic expressions and understand how they're evaluated
  - Read and understand complex code expressions
  - **Write conditional code using boolean expressions**
- Show sample app
- Introduce pair programming

Code examples

Agenda

- Learning objectives
  - Write arithmetic expressions and understand how they're evaluated
  - Read and understand complex code expressions
  - Write conditional code using boolean expressions
- **Introduce pair programming**

Pair Programming

- Each Saturday (11am-1pm ET)
- You'll be assigned to a small group of 3-4 people (different each week)
- You'll have a separate repo every week for pair exercises
- **"Driver":** one person shares their screen and is the only one typing
- **"Navigators":** everyone else, helps work through the exercise
- Rotate so that each person has the chance to drive
- Academic support team will drop in and out to help
- Give feedback on your group assignments (see instructions in Slack) each week
- See Zoom link in Slack (it's called "Pair Exercise Zoom")