

# Pseudocode

# Think Programmatically

- Let's take a look at how thinking programmatically helps you tell a computer exactly what to do.
- If you asked people to identify the shortest character in the *Lord of the Rings* series by looking at the lineup below, they would simply compare the characters' heights and pick Frodo, the little Hobbit on the right.



# Think Programmatically (Continued)

- Computers work a little differently — they break things down into more specific steps.
- So how might you instruct a computer to pick the shortest character from *Lord of the Rings*?
- Your first step might be:
  1. "Find the height of character A."
  2. What might the next step be?

# Think Programmatically (Continued)

- Here are some possible instructions you could give a computer to complete this task:
  1. Find the height of character A.
  2. Find the height of character B.
  3. Compare height of character A to character B.
  4. Choose the character with shorter height.
  5. Find the height of character C.
  6. Compare the height of character C to answer from Step 4.
  7. Choose the character with the shorter height.
  8. Repeat steps 5-7 for all remaining characters.
  9. Label the last character who is determined to be shorter "shortest."

# Think Programmatically (Continued)

- As you can see, it takes a computer many more steps to do what humans do intuitively.
- As we begin to write our JavaScript code, we need to continuously think like a programmer so we can make fewer mistakes and identify errors before they become a problem.
- One way to do this is through the use of **pseudocode**.

# What is Pseudocode?

- Pseudocode is a technique that lets you write out your program in simple, step-by-step instructions using plain English rather than a programming language.
- There are no special commands, so errors and omissions are easy to identify.
- It allows you to break down a program into simple steps, like a computer would, without getting caught up or distracted by syntax.
-

# Using Pseudocode

- The goal when using pseudocode is to think through the problem and break it down into simple steps, which can then be written out in code.
- As there are no formal rules for writing pseudocode, each programmer's version will vary slightly.
- When writing pseudocode, organize thoughts into ideas that relate to one another.

# Writing Pseudocode

- Let's take a look at the pseudocode for a program that lets a player know whether or not he or she has passed the current level in a game:
- `// Passing score = 150 points`
- `// Get the player's score`
- 
- `// If the player's score is greater than or equal to the passing score`
- `// Display message "Current level: Passed"`
- `// Otherwise`
- `// Display message "Current level: Failed"`

You'll notice in the example above that each line starts with `//`.

There may be times when we'll want to write "notes," or **comments**, in our code.

Comments tell a computer to completely ignore that line (or lines) in the JavaScript file so that they will not be interpreted.

Putting `//` at the beginning of the line when writing pseudocode allows the pseudocode to be in the same location as the actual code but to be ignored during processing of the source code.



# Comments in JavaScript

- There are two kinds of comments in JavaScript:
- **multi-line** comments (`/* comment */`), which can run across multiple lines
- `/*`
- Here is a multi-line comment.
- This is for longer comments or notes.
- `*/`
- 
- 
- **single-line** comments (`// comment`), which only comment out everything following it on the current line.
- `// Here is a single line comment`

# What Role Does Pseudocode Play in Development?

- Pseudocode is helpful during the initial design or planning stages, especially when you are working with a team.
- It helps you break problems apart before you begin worrying about the syntax. It also gives you a step-by-step process to follow for solving the problem at hand. This is helpful to both your teammates AND future-you, when you revisit your code!
- It allows you to clearly explain concepts and lines of code, even when speaking with non-developers.
- It's also easier to identify problems early on if you use pseudocode as your first step.
-

# What Role Does Pseudocode Play in Development? (Continued)

- As tempting as it may be to jump right into a problem and start writing code, you'll find that taking a few minutes to plan will save you valuable time in the future.
- You'll often find that during the planning process, you or your team member will come up with additional functions or design changes. Starting with pseudocode allows you to incorporate these new ideas without having to rewrite a lot of code.
-

# Pseudo Code: Example 1

- Let's take a look at an example of pseudocode. Sticking with the *Lord of the Rings* theme...
- Suppose you were given the following assignment:
  - Create a button. Every time users click on the button, the Ring of Power gets put on Frodo's finger. This action makes Frodo disappear. If users click on the button again, the Ring of Power gets taken off of Frodo's finger, and Frodo reappears.
- Your pseudocode for the assignment might look like this:
  - `// When the button is clicked`
  - `// If ring is not on Frodo's finger`
  - `// Put ring on Frodo.`
  - `// Make Frodo disappear.`
  - `// If ring is on Frodo's finger`
  - `// Take ring off of Frodo.`
  - `// Make Frodo appear.`
  -
- As you can see, you don't need to know any JavaScript to write pseudocode. You can write out the steps in plain language.

# Pseudo Code: Example 2

- Let's take a look at another example. Suppose you were given the following assignment:
- Create a website with a pop-up survey that appears when the homepage loads. The user must check off the box "I love *Lord of the Rings!*" and click "Submit" in order to move on to the next page.
- How would you write the pseudocode for this assignment?
- Take out a piece of paper, and write down your own pseudocode. Continue reading this lesson when you're done.

# Pseudo Code: Example 2 (Continued)

- Your pseudocode for the assignment might look like this:
- `// When user enters website`
- `// Pop up survey.`
- `// When user clicks the check box`
- `// Highlight SUBMIT button.`
- `// When user clicks SUBMIT`
- `// IF the check box has been clicked, close the survey.`
- `// otherwise, prompt them to answer the survey`
- 
- Your pseudocode may have looked different, but that's totally fine as long as you covered all the steps required for the computer to understand the task.
- JavaScript is a language with a different syntax than other languages you may have previously used to code. This means there is a lot of room for error, so utilizing pseudocode can be especially helpful when it's your first time coding in JS.

# Pseudo Code: Example 3

- Let's take a look at one final example. Suppose you were given the following assignment:
- Create a program that lets a player know whether or not he or she has passed the current level in a game. In order to pass the level, the player must have a score of 150 or higher.
- How would you write the pseudocode for this assignment?
- Take out a piece of paper, and write down your own pseudocode. Continue reading this lesson when you are done.

# Pseudo Code: Example 3 (Continued)

- Your pseudocode for the assignment might look like this:
- `// Passing score = 150 points`
- `// Get the player's score`
- 
- `// If the player's score is greater than or equal to the passing score`
- `// Display message "Current level: Passed"`
- `// Otherwise`
- `// Display message "Current level: Failed"`
- 
- Again, it's okay if your pseudocode looks different from the sample code above. Just make sure you've covered all of the conditions.



# Summary

- Breaking down problems in this fashion might take a bit of time at first. But once you gain some practice, it'll become second nature.
- Many of the programs you'll be working on in the following chapters are challenging. Instead of trying to tackle a problem all at once, take a few minutes to **think about the overall goal of the program, and then break that answer down into series of short instructions.** It'll be time well spent.
- Watch the video on the next slide to hear developers talk about the importance pseudocode plays in their workflow.

# Video: How Do You Use Pseudocode?

- [MyGA | General Assembly](#)

# Exercise

- Now it's your turn to practice writing pseudocode!
- Try your hand at writing pseudocode for this program. Remember, the goal is to think through the problem and break it down into simple steps.
- You can use a pen and paper or a text editor to complete this exercise.
-

# Exercise Instructions

- We'd like to create a program that checks to see whether or not movie patrons are eligible for a discount and then displays the appropriate ticket price.
- Here is the pricing structure:
  - 1) If patrons are 65 or older, they are eligible for senior discounts. The ticket price for seniors is \$6.00.
  - 2) If patrons are 25 or younger, they are eligible for student discounts. The ticket price for students is \$8.00.
  - 3) All other tickets cost \$10.00.
- Stuck? Check out the solutions in the Study Guide at the end of this lesson. Not stuck? Look at you – a certified pseudocoding wizard! Nicely done. See you in the next lesson.
-