Intro:

"Today, we begin exploring big O notation, starting with O(n). I've chosen O(n) not for its efficiency but because it's the simplest to understand."

A1:

"Let's break this down with a practical example: a function called logItems. This function will run a for loop and console log each iteration's index."

A2:

"Imagine the function logItems is set to execute a loop 'n' times. If we pass the number 10 to this function, it will run ten times, each time logging the loop's current index from zero to nine."

A3:

"This straightforward demonstration helps us visualize an O(n) operation, where the number of operations, or console logs in this case, is directly proportional to 'n'."

A4:

"Now, let's switch to Chrome DevTools to see this function in action. We'll call logItems with the number ten and observe the outputs, ranging from zero to nine."

A5:

"Reflecting on this example graphically, O(n) can be represented as a straight line on a graph. The horizontal axis represents 'n', and the vertical axis represents the number of operations, illustrating the direct proportionality."

Outro:

"That concludes our introduction to O(n) notation. As we progress, we'll add more complexities to our graph, comparing different big O notations, but for now, this understanding of O(n) sets a solid foundation."