**JavaScript**

The word i18n is a common abbreviation of internationalization the developer community use instead of typing the whole word and trying to spell it correctly. Similarly, a11y is an abbreviation of accessibility. Write a function that takes a string and turns any and all "words" (see below) within that string of length four or greater into an abbreviation following the same rules.

Notes:

* A "word" is a sequence of alphabetical characters. By this definition, any other character like a space or hyphen (e.g. "elephant-ride") will split up a series of letters into two words (e.g. "elephant" and "ride").
* The abbreviated version of the word should have the first letter, then the number of removed characters, then the last letter (e.g. "elephant-ride" => "e6t-r2e").

Example:

abbreviate("elephant-rides are really fun!")

// ^^^^^^^^\*^^^^^\*^^^\*^^^^^^\*^^^\*

// words (^): "elephant" "rides" "are" "really" "fun"

// 123456 123 1 1234 1

// ignore short words: X X

// abbreviate: "e6t" "r3s" "are" "r4y" "fun"

// all non-word characters (\*) remain in place

// "-" " " " " " " "!"

=== "e6t-r3s are r4y fun!"

Question – How do you know that your solution is correct?

**HTML & CSS**

Create an HTML document containing a single element that resizes with the browser window. This element is offset in the browser by 100px on the top, bottom, left and right. This element should contain three side-by-side child elements. These elements are vertically aligned in the middle of the variable-height *blue* parent element.

* The *green* element on the left measures 100px x 200px
* The *purple* element on the right measures 100px x 300px
* The *grey* element in the middle is 400px high, and of variable width with a minimum of 100px

