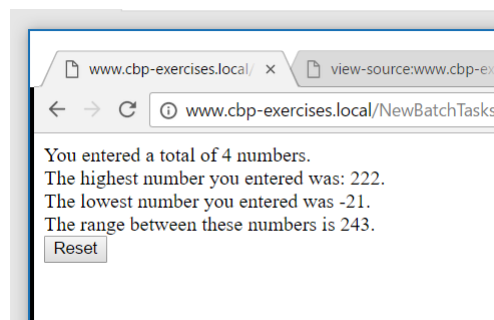
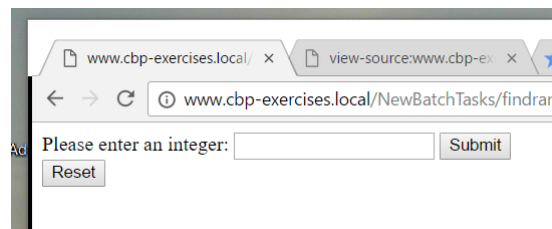


Task 2. Reporting the range between input values: user input (\$_POST) and PHP.

Write a PHP script that registers the input of the user (integers). The user should have the opportunity to provide the program with new integers until he enters the value 0. The page should refresh, evaluate and process the given integer and empty the input field so the user can provide a next integer.

When the number 0 is entered the script should display the largest and the smallest number in the list, as well as the total amount of numbers that he provided. Also it should provide the user with information about the difference between the largest and smallest number.



Additionally, you should make sure that:

- If the user enters only one value before he hits 0, the program should report that value as both the largest and smallest.
- If the user enters 0 on the very first input line, then no values have been entered, and your program should display a message to that effect.
- There should be a reset button to restart the program.
- The input field should disappear when the user submits 0. The reset button should stay.

Step by step programming guide

1. We can start off by creating the form in HTML. With this form the user can provide us with an integer input. We already have to think what variables we need to create and run this program:
 - a. the integer
 - b. the lowest integer so far
 - c. the highest integer so far, and
 - d. the amount of integers already submitted).

(Hint: remember that forms can also pass on hidden values.)

2. We can also create a reset button. Reset can be done by passing on a (hidden) variable using the POST method, but if you can think of another way, feel free to try.
3. Now it is time to write the actual PHP script. The first thing I do is check whether the necessary `$_POST` variables are set. If these are not set there are two possible options: it is the first time the user visits or the user just hit the reset button. Let's start with coding the process of handling a successful response.
 - a. The `$_POST` variables should be requested and set to PHP variables to work with.
 - b. The number that was put in by the user should be evaluated and the specific variables should be adjusted.
 - c. If the number that was put in by the user was 0 the program should start the right actions according to how many inputs have so far been received.
4. Now we code in PHP what will happen if the user hits the reset button. Will there be a variable passed on by the form? How to handle this variable?
5. The last else case should handle the situation in which the user visits for the first time and there are no `$_POST` variables available. The variables should be set to default values.
6. Adjust the program so that when the user hits 0 the input field is no longer shown. This can be done by adding PHP-tags to (around) the HTML part of the page.