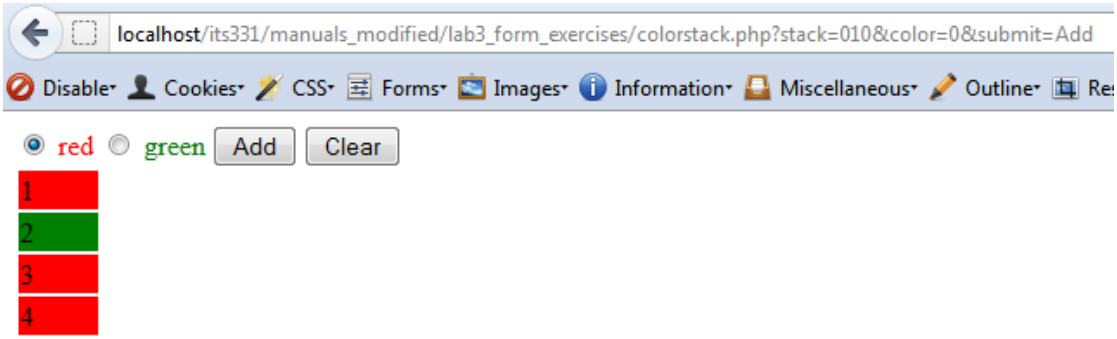
School of Information and Computer Technology
Sirindhorn International Institute of Technology
Thammasat University
ITS331 Information Technology Laboratory I

Laboratory #3: HTML Forms and PHP Exercises

Exercises

1. Write a PHP script with an HTML form that lets the user stack the selected color onto previously selected color list, after a button click. Use radio button and submit buttons as shown.

For example, here is a screenshot after adding "red", "green", "red", and "red".



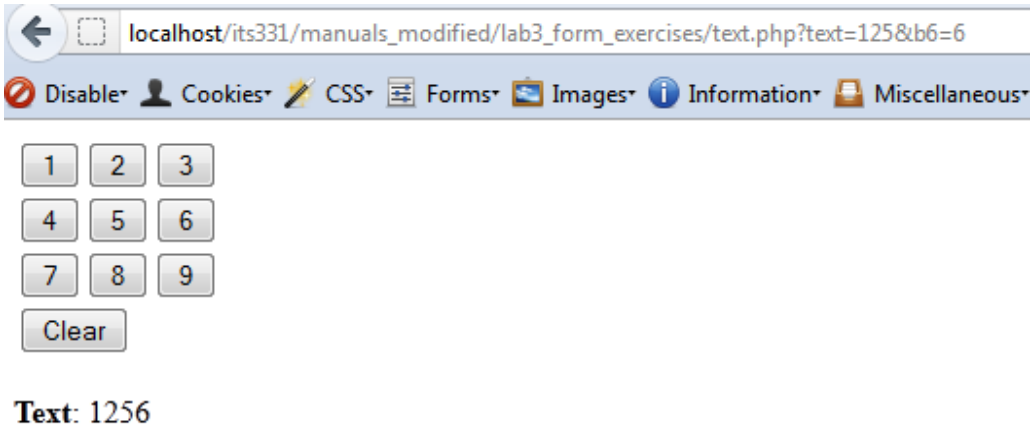
After clicking the "Add" button, the newly added color should be put at the bottom of the stack, along with its current number (starting from 1). Clicking "Clear" should remove all colors from the stack.

HINTS:

- Use `<table>` and use `<tr><td bgcolor="red">$i<td></tr>` for each red color selected.
- Both "Add" and "Clear" buttons are `<input type="submit" name="……">`. Make sure to set a different name for each button. The name can be used to check which button was pressed.
- The two colors may be represented with, say, 0 for red and 1 for green. So, the color list can be represented with a 0-1 string. In the above example, the string representation is 0100.
- Set the form action to the same PHP page.
- See "Keeping State in PHP" section

2. Write a PHP script with an HTML form that allows the user to click a number (submit button) from 1 to 9. The selected number is then appended to the number list selected so far.

For example, here is a screenshot after clicking "1", "2", "5" and "6".



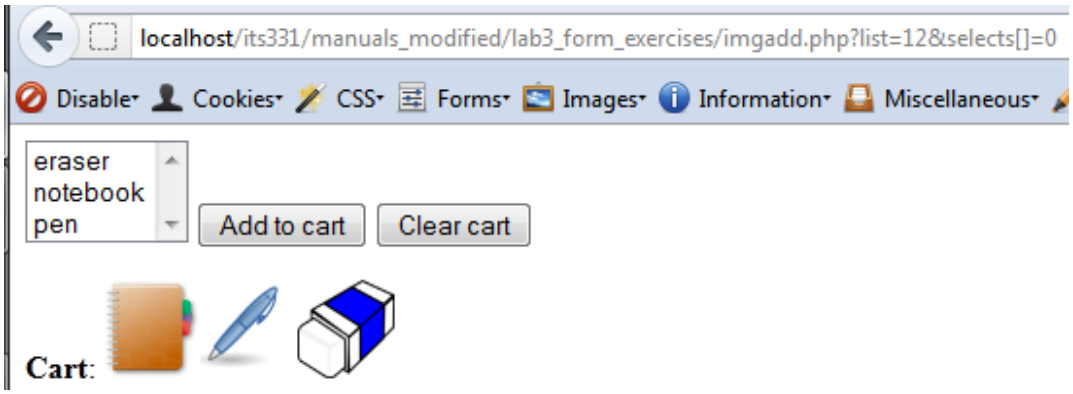
On each click of a number button, the clicked number should be immediately appended to the number list. Clicking "Clear" button should delete the entire number list.

HINTS:

- Each number button is a submit button having the code of the form `<input type="submit" name="b$i" value="$i">`. For example, for button "3", the HTML code would be `<input type="submit" name="b3" value="3">`
- Use `<table>` to organize the buttons.
- Set the form action to the same PHP page.
- See "Keeping State in PHP" section

3. Write a PHP script with an HTML form to simulate a simple shopping cart. Clicking "Add to cart" button should add the selected item to the cart. Display selected items with images.

For example, here is a screenshot after adding "notebook", "pen" and "eraser".



Clicking "Clear cart" should remove all items in the cart (i.e., remove all the displayed images).

HINTS:

- Represent each item as a number and keep the selected items as a string. For example, 0 represents eraser, 1 represents notebook, and 2 represents pen. So, in the above example, the items in the cart can be summarized as "120".
- Set the form action to the same PHP page.
- See "Keeping State in PHP" section

Keeping State in PHP

HTML, by its nature, is stateless. That is, calling the same HTML page over time will produce the same content regardless of what the user did to the page in the past. By using PHP, there are many ways to keep states so that a page presents the content based on what the user did in the past.

1. Using cookie which is a file stored on the client-side
2. Using PHP session which is a temporary global session of variables. So, the variables can be used across subsequent page accesses, as long as the browser is not closed.
3. **Using hidden field of HTML form**, which is what you learned in this lab.

In this exercise, you are required to use hidden fields.

Hidden fields are declared as follows:

```
<input type="hidden" name="data" value="<?=$v?>">
```

In this code snippet, the hidden field has a name "data", and its value is the value of `$v`. To keep state across subsequent page accesses, `$v` should be kept to the hidden field on each access every time there is a modification. The general usage is shown next.

```
<?php
    $v = $_GET['data']; // data = name of the hidden field
    $v = ... // update $v
?>
<!DOCTYPE html>
<html>
<body>
<form action="....php" method="GET">
    <input type="hidden" name="data" value="<?=$v?>">
    ...
    ...
</form>
</body>
</html>
```

It can be seen that, in the beginning of the code, $v is set to the value in the hidden field. $v is then updated as needed. Then, in the <form>, the value of $v is kept for the next access by setting it as the value of the hidden field. In this way, the value of $v is always kept (in the hidden field).