# IT 328 Pair Program 1

For this pair-program, you will be collaborating with a partner, but you will each be typing your own code on your own computer, and committing to your own Git repository as you go. Before you start coding:

* Log into GitHub and create a new repository called pp1. Copy the URL.
* Launch PhpStorm and choose Start SSH Session under the Tools menu.
* Navigate to your 328 directory.
* Clone your repo.
* In the editor, right click the pp1 directory on the right (remote), and choose Download from Here.
* Do all of your work from the local view in the left pane!

After completing *each step* below, test your changes. Then add, commit and push your changes to GitHub. Do a git status *after each command*. You should end up with a total of 8 commits.

### Steps

1. Create an index.php file in your pp1 directory and give it a title and header “Pair Program 1.” Preview the page in a browser. Commit your changes.
2. In a PHP block, define a function **printArr()** that takes an array as a parameter and prints the array, one item per line. Then, outside the function, define an array called $numbers, which contains 7, 9, 8, 9, 8, 8, 6. Print the array using your function. Commit your changes.
3. Move your print function into an include file called functions.php in the same directory. Make sure that it still works. Commit your changes.
4. In your functions file, define a function called **largest()** that takes an array as a parameter and returns the largest value in the array. Test the function from your index page. Commit!
5. In your functions file, define a function called **average()** that takes an array as a parameter and returns the average of the values in the function. Test the function from your index page. Commit your changes.
6. In your functions file, define a function called **removeDups()** that takes an array as a parameter and returns an array with duplicates removed. So, given the $numbers array, removeDups() would return [7, 9, 8, 6]. (Order doesn’t matter.) Test the function from your index page. Commit your changes.
7. In your include file, define a function called **distribution()** that takes an array as a parameter and returns an associative array with each value from the original array, and the number of times that value occurs in the original array. The keys should be sorted.   
     
   So, given the $numbers array: [7, 9, 8, 9, 8, 8, 6]  
   distribution() would return [6=>1, 7=>1, 8=>3, 9=>2]  
     
   Test the function from your index page. Commit your changes.