Coding Challenge Level 2: Solution

Let's review the 4-step framework for solving this coding challenge.

1. **Understand and appreciate the problem statement.**

- Did you fully understand the problem?
- Do you know what inputs are required?
- Do you know what the desirable output is?
- Does it give you enough information?

2. **Break down the problem into functions.**

- Define the problem statement into various functions or modules.
- Does each module solve a specific subset of the problem?
- Can the functions/modules be reused?

3. **Apply common patterns or language constructs that you already know for the problem.**

- Do you see a common pattern that you can use?
- Do you see a loop to use or a language function that might make things easy?
- Pseudocode your proposed solution.

4. **Start coding.**

- This is the best place to start really coding.
- Ignore difficult parts for now – begin by solving a simpler sub-part.
- Run the code as you work and let errors show you where you can correct the code.
- Start with input gathering, then processing logic, and then output.
- Compare your solution for its accuracy and completeness relative to the problem statement.
- If time permits, optimize the solution by looking at various alternatives to shorten the code or increase its efficiency.

Javascript Solution

```
function removeDuplicates(arr) {
  var usedItems = [];
  arr.forEach(function(item){
    if (!usedItems.includes(item)){
      usedItems.push(item);
    }
  });
  return usedItems;
}
```

Python Solution

```
def removeDuplicates(arrlst):
    result = []
    for item in arrlst:
        if item not in result:
            result.append(item)
    return result
```

Procedure

In the above code:

- We first defined a function called *removeDuplicates* which would take input as an array.
- The input variables can contain any number of strings or integers.
- We create an empty array to hold the new array that will contain non-duplicate items.
- We iterate through the array.
- We compare each item of the array to the items in the new empty array. If it doesn’t exist in the new array, we push this item. If the item exists, the code would ignore it and go to the next array item.
- In JavaScript, we use inbuilt *push* function.
- In Python, we use inbuilt *append* function.
- Finally, we show the new array as the output.

Author

Rama

Other Contributor(s)

Marty Hale-Evans, Instructional Designer