

A. Identify Reusable Code

Review your existing scripts from previous assignments. Find sections of code that repeat or perform a specific, self-contained task.

```
//generate random number between 0 and 5  
var rand1 = Math.floor(Math.random() * 6);
```

```
//generate random number between 0 and 5  
var rand2 = Math.floor(Math.random() * 6);
```

One section of code in the featured script that gets repeated multiple times is generating a random number between 0 and 5.

```
//each random number corresponds to choosing a different review  
if(rand1 === 0) {  
    randomReview = review1;  
}  
//review with most likes is weighted higher  
else if(rand1 === 1 || rand1 === 2) {  
    randomReview = review2;  
}  
else if(rand1 === 3) {  
    randomReview = review3;  
}  
else if(rand1 === 4) {  
    randomReview = review4;  
}  
else {  
    randomReview = review5;  
}
```

```
//each random number corresponds to a different list  
//list with most likes is weighted higher  
if(rand2 === 0 || rand2 === 1) {  
    randomList = list1;  
}  
else if(rand2 === 2) {  
    randomList = list2;  
}  
else if(rand2 === 3) {  
    randomList = list3;  
}  
else if(rand2 === 4) {  
    randomList = list4;  
}  
else {  
    randomList = list5;  
}
```

Another section of code in the featured script that gets repeated multiple times is this block of code that picks a random list given a random number and 5 lists.

```
//use getElementById to change the header title top Top 5 Users  
var username_doc = document.getElementById("top_username");  
username_doc.textContent = ("Top 5 Users");
```

This code in my top_user script is self contained and doesn't rely on any other code in the script.

B. Plan Your Functions

Note which sections can be turned into functions. Consider where parameters, arguments and return values could make your functions more flexible and efficient.

All three of the sections I showed above can be turned into functions.

1. For the random number function it could be made more flexible by taking in a parameter that specifies the range. The function can then return a random number between 0 and the end of the range.
2. For the function which chooses a random list, the parameters for the function are the random number and 5 lists to choose from. The return value is going to be one of the 5 lists which will be chosen randomly.
3. For the function that changes the heading, no parameters or return values are necessary as the function only performs one self contained task.

C. Document Your Plan

Answer the following questions in your document:

1. *Describe the specific code segments you plan to modularize into functions. Explain what each function will do and whether it will have parameters, arguments, or return values.*

Segment 1: This code segment generates two random numbers between 0 and 5. This can be turned into a function with a parameter that specifies the range, and returns a random number within that range.

Segment 2: This code segment picks a random list out of 5 lists. This can be turned into a function with a parameter for the random number that picks the list, and a parameter that contains the 5 possible lists that can be chosen. It will return one of the 5 possible lists. The list or review with the most likes is weighted higher, with number 0 and 1 choosing it as the featured item.

Segment 3: This code segment gets the element with "top_username" id and changes its text content to "Top 5 Users". This can be turned into a self contained function with no parameters, no arguments, and no return values.

2. *Discuss why you chose to convert these sections into functions. Focus on how modularization improves your code's readability, reduces repetition, and simplifies debugging or future updates.*

I chose to convert these sections into functions to reduce repetition, especially for the first two code segments for generating a random number and getting a random list. This also helps to improve readability as the code is not cluttered with repeated statements. Being able to specify the range for the random number function also allows for future updates if I ever want to choose between any number of random elements. I chose to convert the code segment which changes the text content of a header because it improves readability by putting self contained code in its own section.

3. *Propose any new functions that could enhance your site's features or improve user interaction. For each new function, describe its purpose, what parameters it will take, whether it will return a value, and how it fits into your overall project.*

One new function idea is one that puts a random highlight on one of the top rated games on the homepage to show that it is the most popularly reviewed game currently. This could be accomplished by a function that picks a random image to put a gold highlight onto. This function does not take any parameters and will not return a value. This function increases interactivity and engagement as it shows what game is the most popularly reviewed at the moment, which makes users want to look at reviews for that game.