

```
HTMLSearch(currentHTMLElements, columnWords, n, chapterHTMLElements, isSearchingFirst){
```

```
MutableArray wordsToCheck;
```

```
if(isSearchingFirst == true){
    for (i = 0; i < n ; i++){
        wordsToCheck.add(columnWords[i]);
    }
}
else {
    for (i = columnWords.count - 1; i > columnWords.count - n; i--) {
        wordsToCheck.add(columnWords[i]);
    }
}
```

```
MutableArray validElements;
```

```
//iterate through all currentHTMLElements
for (i = 0; i < currentHTMLElements.count ; i++){

    HTMLObject localCopy = currentHTMLElements[i];
    MutableArray wordsForThisElement;

    //iterate through the wordsToCheck
    for (j = 0; j < n ; j++){

        String currentWord = wordsToCheck.[j];

        //Track the numberOfOccurrencesForEachWord
        MutableArray numberOfOccurrencesForEachWord;

        for (k = 0; k < n ; k++){
            //If the current
            if(currentWord == localCopy.wordArray[k]){
                numberOfOccurrencesForEachWord.add(i);
            }
        }

        if(numberOfOccurrencesForEachWord.count != 0){

            wordsForThisElement.add(numberOfOccurrencesForEachWord);

        }
    }

    //Check that the number of wordsForThisElement is the same as the depth of
    recursion. If so, add this Element to the validElements array
    if (wordsForThisElement.count == n) {
        validElements.add(localCopy);
    }
}

if (validElements.count > 1) {
    // More than one element contains all words at this level, increase n by 1.
    return HTMLSearch(validElements, columnWords, n+1, chapterHTMLElements,
        isSearchingFirst);
}
```

```
else if(validElements.count == 0){
    //A worst case senario has occured, pop off the first or last word and start again.

    if(isSearchingFirst){
        columnWords.remove[0];
    }
    else{
        columnWords.remove[columnWords.count];
    }

    return HTMLSearch(chapterHTMLElements, columnWords, 1, chapterHTMLElements,
isSearchingFirst);
}
else{
    //HTML Element found
    return validElements[0];
}
}
```