

Aufgabe 3: Wörter aufräumen

Vorwort

Alle Methoden sind mit HTML und Javascript implementiert.

Das Ausführbare Programm kann man bei folgende Adresse finden:

https://tnthunter22.github.io/Bwinf_2020/Aufgabe%203/Aufgabe_3.HTML

Lösungsidee

Mein Lösungsidee war , dass ich eine tabelle erstelle und dann bei der tabelle ablese elches wort auf welchen platz kommt und das mit 1 oder 0 markieren,

	_ _ t	_ _ _ t	m _ _
ist	1	0	0
mit	1	0	1
test	0	1	0

und dann wenn ein wort auf ein kodierte wort kann und kein anderes wort auf das kodierte wort kann, ändere ich jede 1 in der reihe und spalte zu einer 0.

	_ _ t	_ _ _ t	mit
ist	1	0	0
mit	0	0	0
test	0	1	0

Umsetzung

Schritt 1: Datei lesen & Tabelle erstellen

Schritt 2: In der Tabelle möglichkeiten für die kodierten Wörter suchen, wenn es mehrere wörter für eine Spalte gibt dann überspringen

Schritt 3: die richtigen wörter speichern

Schritt 4: Schritt 2 und 3 wiederholen bis es keine Wörter mehr gibt

Schritt 5: Die Gespeicherten wörter in dem kodierten Satz ersetzen

Beispiele

Aufgabe 3

Datei auswählen raetsel0.txt

Read File

Input Text

1)

h, a, r, e, b!

2)

arbeit eine für je oh was

Result

oh je, was für eine arbeit!

Aufgabe 3

Datei auswählen raetsel1.txt

Read File

Input Text

1)

m a e s e . D a i u e n u l h h e .

2)

Am in als das Das die und sehr Leute viele wurde wurde Anfang machte wütend falsche Schritt Richtung angesehen Universum erschaffen allenthalben

Result

Am Anfang wurde das Universum erschaffen. Das machte viele Leute sehr wütend und wurde allenthalben als Schritt in die falsche Richtung angesehen.

Quellcode

```
var countOfWordsFound = 0;
var clearCheckTable = function(checktable,row,col){
    for(var o = 0 ;o < checktable.length;o++){
        checktable[row][o] = 0;
        checktable[o][col] = 0;
    }
}

var checkerLines = function(checktable,foundWords,fillWords){
    for(var i = 0;i < checktable.length;i++){
        var countOfPossibilities = 0;
        var foundindex = 0;
        for(var j = 0;j < checktable[i].length;j++){
            if(checktable[i][j] === 1){
                foundIndex = j;
                countOfPossibilities++;
            }
        }
        if (countOfPossibilities === 1){
            foundWords[i] = fillWords[foundIndex];
            countOfWordsFound++;
            clearCheckTable(checktable,i,foundIndex);
        }
    }
}

var checkerColumns = function(checktable,foundWords,fillWords){
    for(var i = 0;i < checktable.length;i++){
        var countOfPossibilities = 0;
        var foundindex = 0;
        for(var j = 0;j < checktable[i].length;j++){
            if(checktable[j][i] === 1){
                foundIndex = j;
                countOfPossibilities++;
            }
        }
        if (countOfPossibilities === 1){
            foundWords[foundIndex] = fillWords[i]
            countOfWordsFound++;
            clearCheckTable(checktable,foundIndex,i);
        }
    }
}

var checkerDups = function(checktable,foundWords,fillWords){
    for(var i = 0;i < checktable.length;i++){
        foundIndex = -1;
        for(var j = 0;j < checktable[i].length;j++){
            if(checktable[i][j] === 1){
                foundIndex = j;
                break;
            }
        }
        if (foundIndex > -1){
            foundWords[i] = fillWords[foundIndex]
            clearCheckTable(checktable,i,foundIndex);
        }
    }
}
```

```

    }
}
var countChecker = function(fillWords,missingText){
    var checktable = [];
    for(var i = 0;i < missingText.length;i++){
        checktable[i] = [];
        loopFillWords:
        for(var j = 0;j < fillWords.length;j++){
            checktable[i][j] = 1;
            if(missingText[i].length === fillWords[j].length){
                for(var h = 0;h < missingText[i].length;h++){
                    if(missingText[i].charAt(h) === "_"){
                        continue;
                    }else if(missingText[i].charAt(h) !== fillWords[j].charAt(h)){
                        checktable[i][j] = 0;
                        continue loopFillWords;
                    }
                }
            }else{
                checktable[i][j] = 0;
            }
        }
    }
}
return checktable;
}

var checker = function(missingText,fillWords,output){
    var checktable = countChecker(fillWords,missingText);
    var allWordcount = fillWords.length;
    countOfWordsFound = 0;
    var iterations = 0;
    foundWords = [];
    while(allWordcount > countOfWordsFound && iterations++ < allWordcount * 10){
        checkerLines(checktable,foundWords,fillWords);
        checkerColumms(checktable,foundWords,fillWords);

    }
    checkerDupes(checktable,foundWords,fillWords);
    for(var i = 0;i < fillWords.length;i++){
        output = output.replace(missingText[i],foundWords[i]);
    }
    return output;
}

```


