Hasonlóságelemzés java-script-alapon

(Similarity analyses based on java-scripts)

Pitlik Mátyás, Pitlik Marcell, Pitlik László (MY-X team)

Kivonat: A programozás tanulása és tanítása végtelen sok módon történhet. Ezek hatásosságát és hatékonyságát általában véve nem szokás mérni. A programozásra való alkalmasság eleve elvár egy magas fokú szuverenitásra törekvést, az autodidakta tanulás iránti vágyat/készséget. Így vélelmezhetően minden olyan esettanulmány, mely hasonló/azonos funkcionalitások eltérő nyelveken, eltérő logikák mentén történő bemutatását teszi lehetővé lényegében teljes részletgazdagsággal, az a szuverén és autodidakta tanulást érdemben képes illene, hogy legyen támogatni. A cikk célja programozástanulást elszakítani a klasszikus programnyelv-tanulás/tanítás logikájától és elsőként/párhuzamosan arra felhívni a figyelmet, hogy a programozás filozófiája terén is illik fejlődni, nem csak az adott programnyelven belüli virtuóz megoldások mentén. A cikkben a java-script nyelv kerül fókuszba professzionális programozó által készítve a kódot.

Kulcsszavak: COCO, C-URL

Abstract: The learning and teaching in case of creating of source codes can be realized/executed in quasi unlimited ways. The effectivity and/or efficiency of these ways will not be measured in general. The capability of creating source codes assumes (in particular) a kind of struggling toward sovereignty and the high acceptance/desire level of auto-didactical learning methods. Therefore, each deep detailed case study (source code) should be useful for auto-didactive learners/teachers where the same/similar functionalities will be covered through source codes using other programming languages and/or programming structures. The article tries to create significant distances between language-oriented learning/teaching logics and the planning of creating qualitative source codes in a quasi language-independent way in order to catalyse more philosophy behind the source codes. The article focuses on a java-script solution created by a professional programmer.

Keywords: COCO, C-URL

# Bevezetés

A hasonlóságelemzésre már számos megoldás született az elmúlt évtizedekben:

* <http://miau.gau.hu/lps/>
* <https://miau.my-x.hu/myx-free/>
	+ <https://miau.my-x.hu/myx-free/coco/index.html>
	+ <https://miau.my-x.hu/myx-free/index_e.php3?x=e01>
* <https://www.google.com/search?q=sealog+seacon&oq=sealog+seacon>
* <http://miau.gau.hu/mm/Putnoki_Gyula.ppt>

* <https://miau.my-x.hu/miau/255/coco_macro.docx>
* (<https://miau.my-x.hu/miau/255/coco_js.docx>)

Itt és most az időközben kialakított C-URL motor meghívását végző java-script-megoldás kerül bemutatásra:

# A kód



76 92 72 95 78 57

100 85 91 82 78 72

79 99 74 78 87 88

93 56 99 50 82 87

68 100 83 93 50 90

91 56 90 90 53 100

50 91 84 73 99 85

64 74 50 87 74 100

86 64 76 82 87 78

93 83 98 95 88 85

87 57 95 75 85 93

100 54 77 69 62 73

75 81 93 70 59 79

85 58 61 50 95 96

88 66 91 58 87 75

53 91 58 70 87 87

999996 1000002 1000020 1000002 1000002 1000002 1000002 1000002 999988 1000019 1000010 999990 999984 1000002 999988 999995

1000004 999988 999990 999998 999998 999998 999998 999998 1000012 999982 999992 1000009 1000015 999998 1000012 1000007

|  |  |
| --- | --- |
|  | <!DOCTYPE html>  |
|  | <html lang="en">  |
|  |  |
|  | <head>  |
|  | <meta charset="UTF-8">  |
|  | <meta name="viewport" content="width=device-width, initial-scale=1.0">  |
|  | <meta http-equiv="X-UA-Compatible" content="ie=edge">  |
|  | <title>COCO fetch demo</title>  |
|  |  |
|  | <script>  |
|  | function transpose(array) {  |
|  | return array[0].map((\_, col) => array.map(row => row[col]));  |
|  | }  |
|  |  |
|  | function rankArray(array, asc = false) {  |
|  | const compare = asc ? (a, b) => a.val - b.val : (a, b) => b.val - a.val;  |
|  | let sorted = array.map((val, idx) => ({ val, idx }));  |
|  | sorted.sort(compare);  |
|  |  |
|  | let cnt = 0;  |
|  | for (let i = 0; i < sorted.length; ++i) {  |
|  | sorted[i].rank = i + 1 - cnt;  |
|  | if (sorted[i + 1] && sorted[i].val === sorted[i + 1].val) {  |
|  | ++cnt;  |
|  | }  |
|  | else {  |
|  | cnt = 0;  |
|  | }  |
|  | }  |
|  |  |
|  | return array.map((\_, i) => {  |
|  | for (let x of sorted) {  |
|  | if (x.idx === i) return x.rank;  |
|  | }  |
|  | });  |
|  | }  |
|  |  |

|  |  |
| --- | --- |
|  | function coco(inverse = false) {  |
|  | const input = document.getElementById('input').value;  |
|  | const data = input.trim().split('\n').map(x =>  |
|  | x.trim().split(/\s+/).map(parseFloat));  |
|  |  |
|  | const oam = transpose(transpose(data).map(x => rankArray(x, inverse)));  |
|  | oam.forEach(x => x.push(1000000));  |
|  |  |
|  | const str = oam.map(x => x.join('%09')).join('%0D');  |
|  | fetch('https://miau.my-x.hu/myx-free/coco/engine3\_curl\_cors.php', {  |
|  | body: 'modell=Y0&matrix=' + str + '??',  |
|  | headers: {  |
|  | 'Content-Type': 'application/x-www-form-urlencoded',  |
|  | },  |
|  | method: 'POST',  |
|  | // mode: 'cors',  |
|  |  |
|  | })  |
|  | .then(x => x.text())  |
|  | .then(output => {  |
|  | document.getElementById(inverse ? 'output2' : 'output1').innerHTML = output;  |
|  | })  |
|  | .catch(console.log);  |
|  | }  |
|  | </script>  |
|  | </head>  |
|  |  |
|  | <body>  |
|  | <div>  |
|  | <textarea id="input" rows="20" cols="100"></textarea>  |
|  | </div>  |
|  | <div>  |
|  | <button id="button">OK</button>  |
|  | </div>  |
|  | <div>  |
|  | <div id="output1"></div>  |
|  | <div id="output2"></div>  |
|  | </div>  |
|  |  |

|  |  |
| --- | --- |
|  | <script>  |
|  | document.getElementById('button').addEventListener('click', \_ => {coco(); coco(true);});  |
|  | </script>  |
|  | </body>  |
|  |  |
|  | </html> |

Tesztelés: <https://miau.my-x.hu/miau/255/coco_js.html>

# Konklúziók

A megoldások közötti alapvető különbségek a tényleges funkcionalitásokon túl (pl. rangsorolás része, vagy nem része a folyamatnak) a programozás során felvállalt gondolati komplexitásban érhető tetten: vö.

* <https://miau.my-x.hu/miau/255/macro_tanulas_tanitas.docx>
* <https://miau.my-x.hu/miau/254/SUDOKU_SOLVER_CUSTOMIZED_SOURCE_CODE.docx>
* <https://miau.my-x.hu/miau/254/cipher1-2-3.docx>

A kódírás előtt tehát érdemes/illik szinte tetszőleges részletességű terveket készíteni az egyes döntések miértjeinek tételes átgondolása/validálása mellett.