**PART 1**

**Query**

**geometry, Area\_Acres,Length\_mil**

**var** rows = data[**'rows'**];

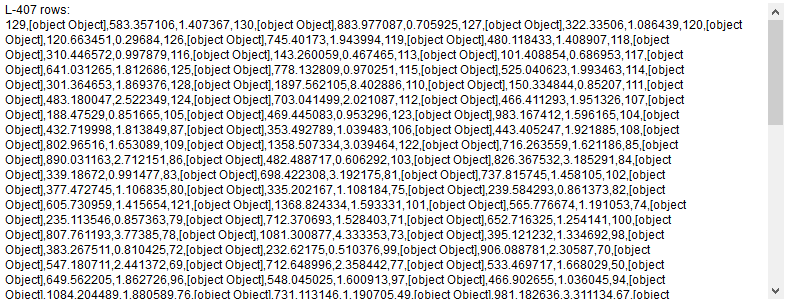
**Alert 1.**

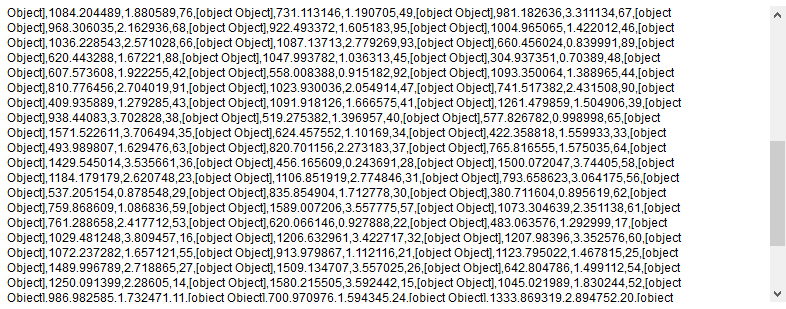
This information comes from the GIS map

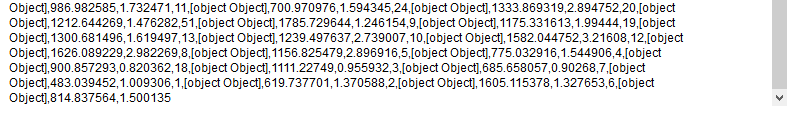
**rows**

alert(**"L-407 rows"** + rows);

Subbasin, geometry-coord., Area, Length







**Alert 2.**

alert(rows.**length**);*//E:*

row: 130

alert(**"Type rows: "** + **typeof 'rows'**);

String

**Alert 3.**

alert(rows[0]);

**129,[object Object],583.357106,1.407367**

alert(**"Type row[0]: "** + **typeof 'rows[0]'**);

String

alert(rows[0][1]);

row[0][1]: [object Object]

alert(rows[0][1][**'geometry'**]);

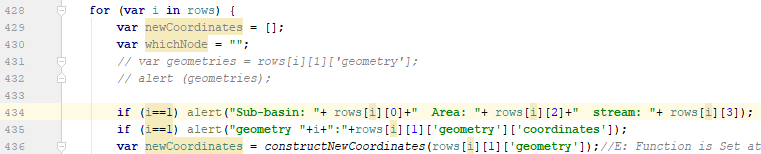
row[0][1]['geometry']: [object Object]

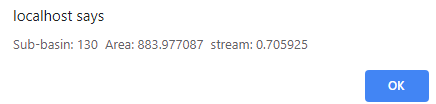
alert(**"row[0][1][1]: "** + rows[0][1][1]);

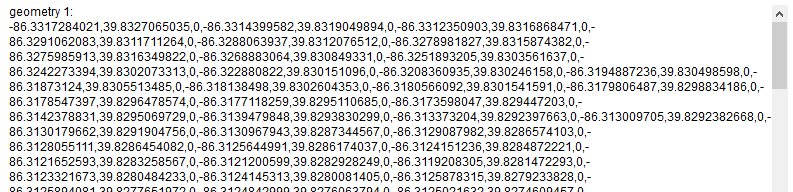
row[0][1][1]: undefined

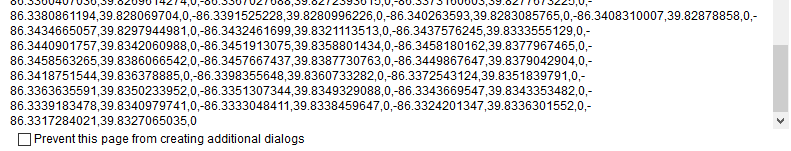
alert (**"Sub-basin: "**+ rows[i][0]+**" Area: "**+ rows[i][2]+**" stream: "**+ rows[i][3])

alert (**"geometry "**+i+**":"**+rows[i][1][**'geometry'**][**'coordinates'**]);



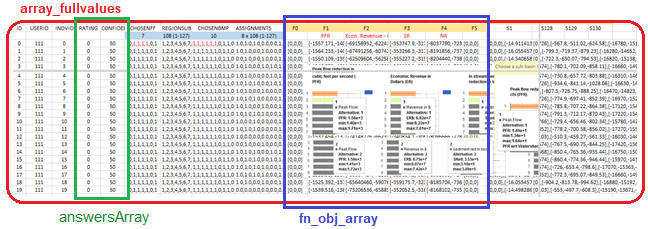






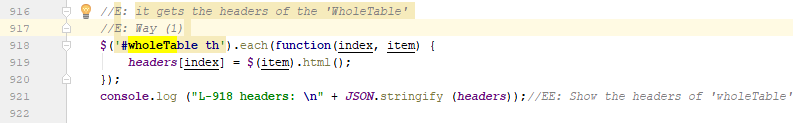
**PART 2 – executed in g2.php**

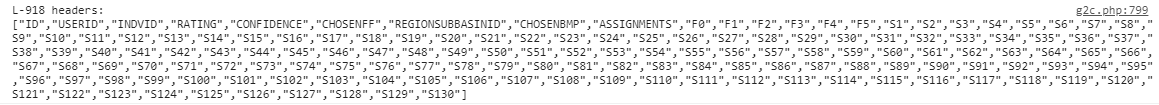
**GROUP 1**

****

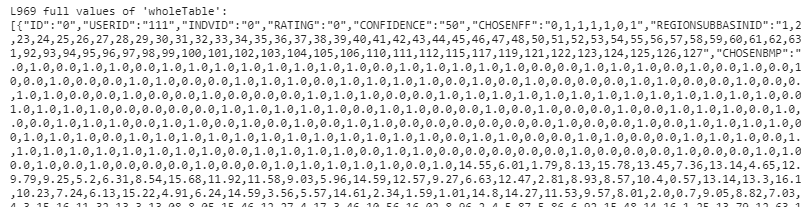
**headers**

BMPArray

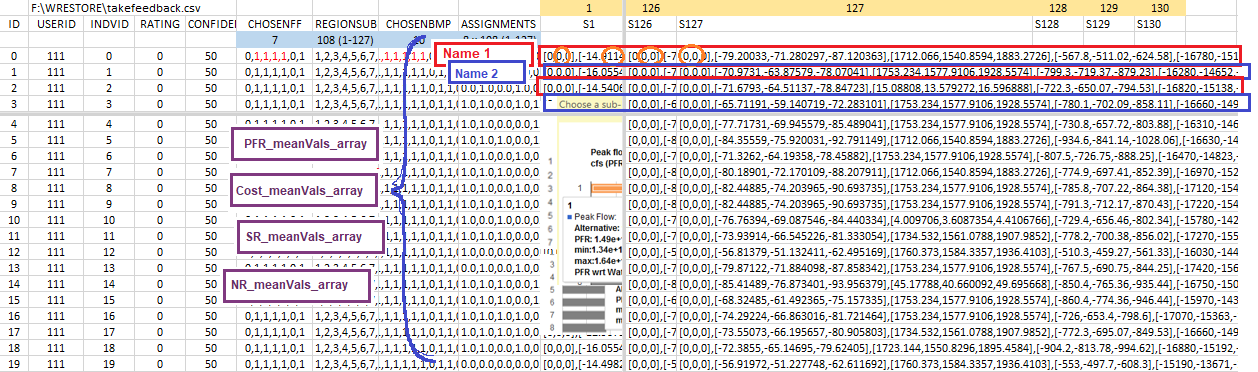
****

****

**console**.log(**"L.969 full values of 'wholeTable':"** + ***JSON***.stringify (***array\_fullvalues***));

****

**PFR\_meanVals\_array, Cost\_meanVals\_array, SR\_meanVals\_array, NR\_meanVals\_array**

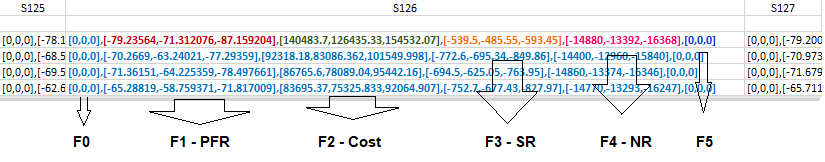
****

NR\_meanVals\_array

SR\_meanVals\_array

Cost\_meanVals\_array

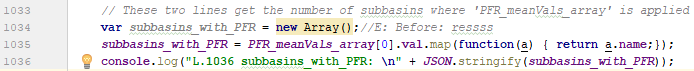
PFR\_meanVals\_array

****

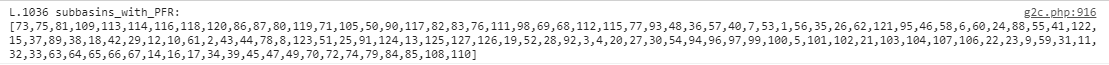
**It means alternative 1**

****

**Subbasins\_with\_PFR**

****

**console**.log(**"L.1036 subbasins\_with\_PFR: \n"** + ***JSON***.stringify(***subbasins\_with\_PFR***));

****

51+53+23 = 127 subbasins in total

**subBMP**

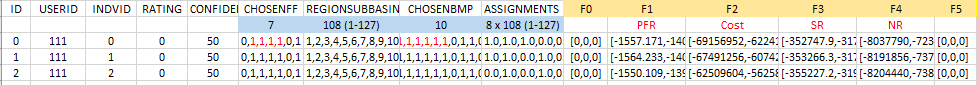
**bmpArray = [subBMP-alt1, subBMP-alt2, subBMP-alt3, …, subBMP-alt20]**

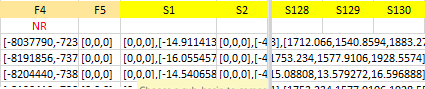
*//E: 'bmpArray' gets the 'CHOSENBMP' of the current alternative starting from alternative 0 to 20. For example, for alternative 1 is:*

****

**GROUP 2**

**Frame**

****

****

**Assignation of variables:**

***bmpArray* =** CHOSENBMP

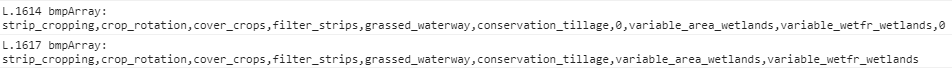
****

This ‘bmpArray’ will be modified by names instead numbers below

***subBasinArrayStart* =** REGIONSUBBASINID

**subBasinArrayStart =** REGIONSUBBASINID as array, for the current alternative, starts from 0 to 20. For example, this is for alternative 1 (array[0]).

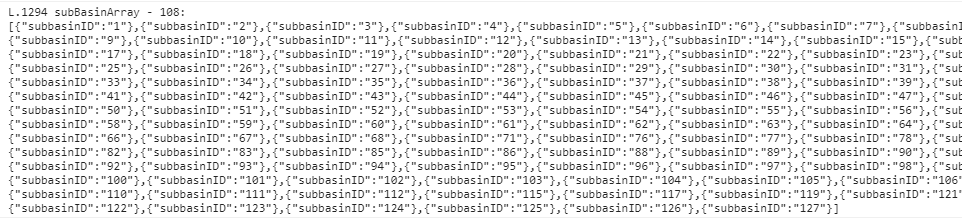
****

****

**assArray =** ASSIGNMENTS for the current alternative, starts from 0 to 20

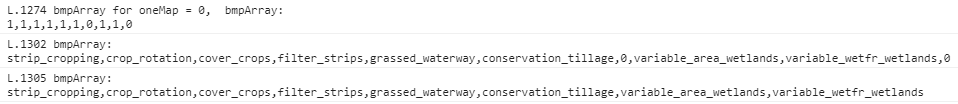
**subBasinArray** = obtained as function of ‘subBasinArrayStart’.

This variable ‘**subBasinArray**’ will be extended below

****

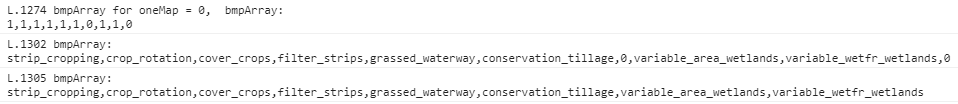
***bmpArray* =** CHOSENBMP

This ‘bmpArray’ was initially assigned as numbers (L.1302 ‘bmpArray’ is modified by names instead numbers, see below), now it was changed as Conservation-Practice names.



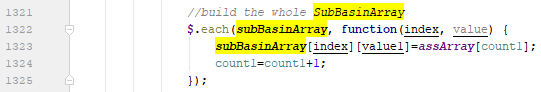
***bmpArray* =** CHOSENBMP

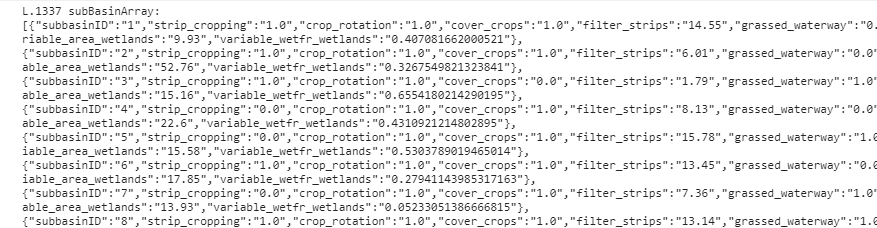
****



Start a second loop

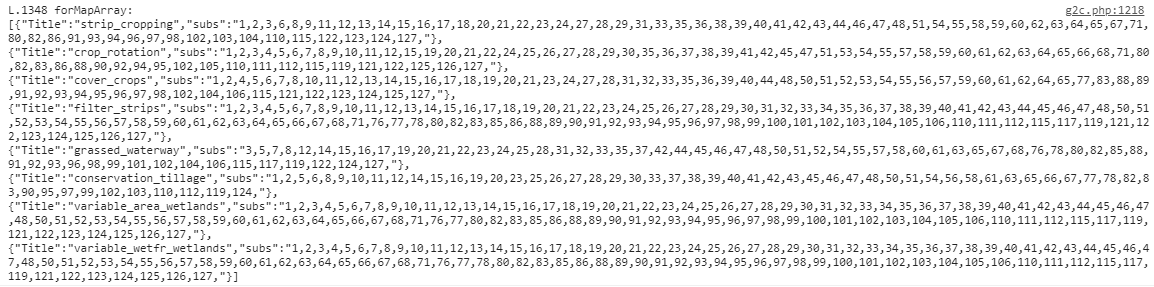
**subBasinArray.** It is extended by taking information from “assArray”

****

****

**forMapArray**

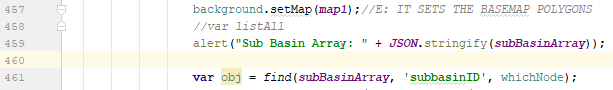
**console**.log(**"L.1348 forMapArray: \n"**+ ***JSON***.stringify(***forMapArray***));

**

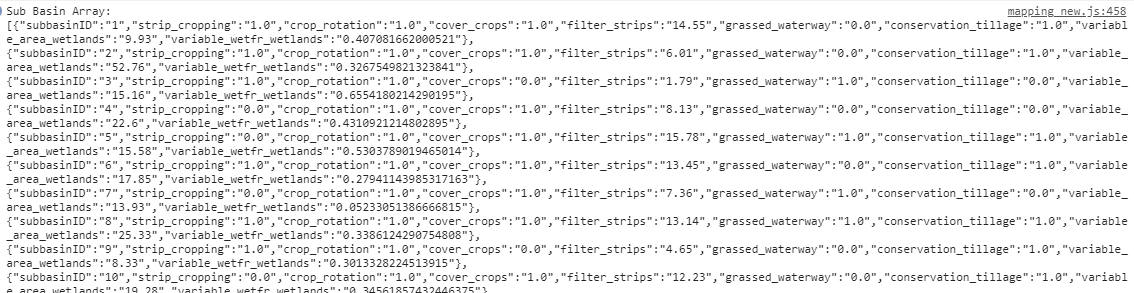
**PART3: Executed in mapping\_new.js**

**subBasinArray**

This information comes from MySQL Database. It gives values for each of 108 subbasins



It gives 130 times this



**PART 3**

*alert (obj);*

Note: ‘find’ function searches the match watershed



It gives this, 130 times



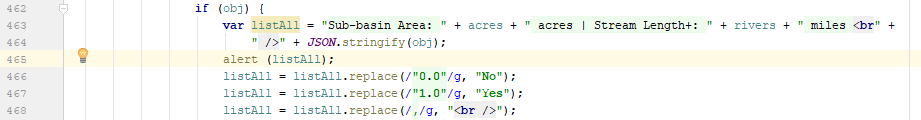
alert (***JSON***.stringify(obj, **null**, 4));



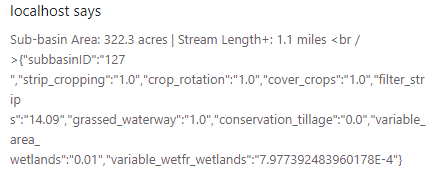
It gives this, 130 times



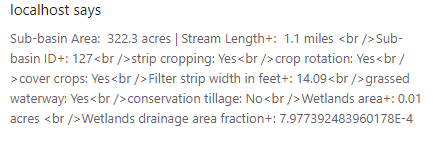
*alert (listAll);*

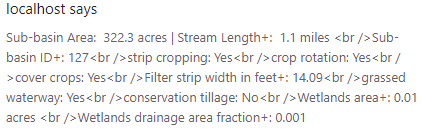


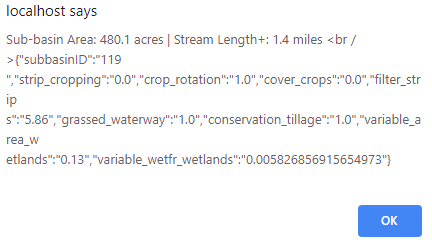
Before

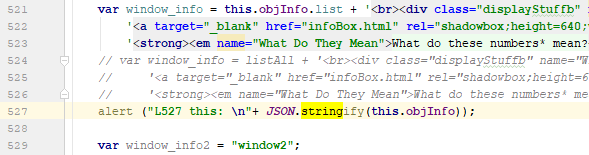


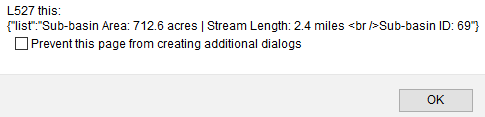
After



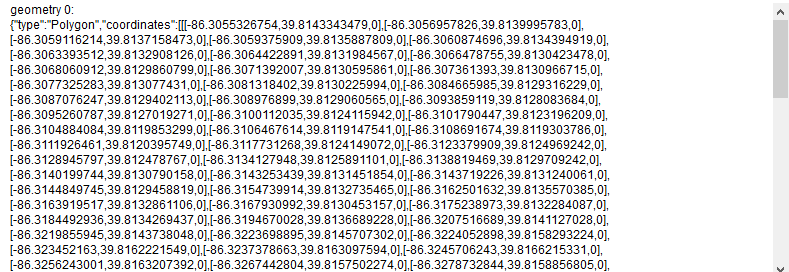


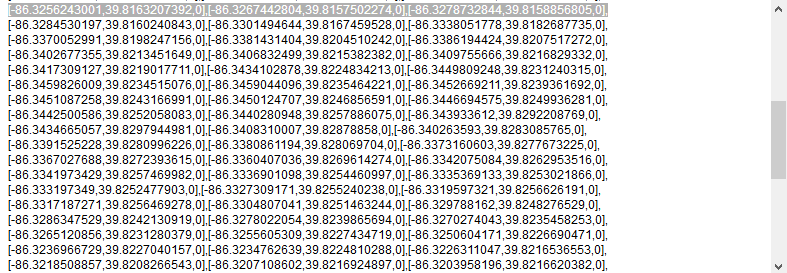
, The blue box is “obj”

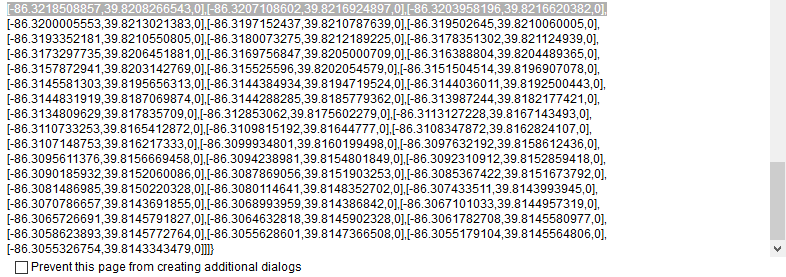




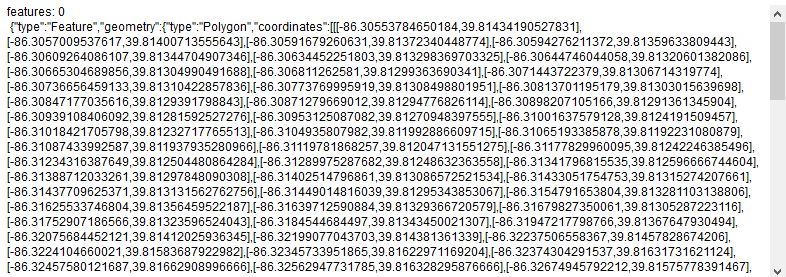


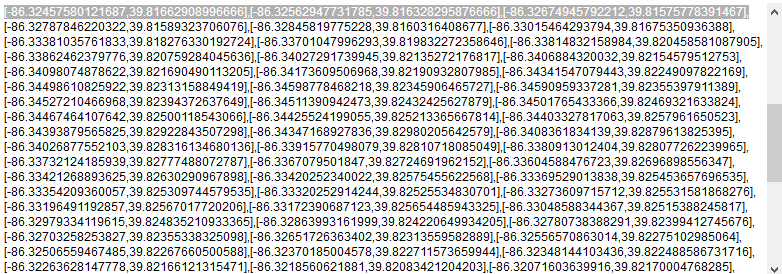


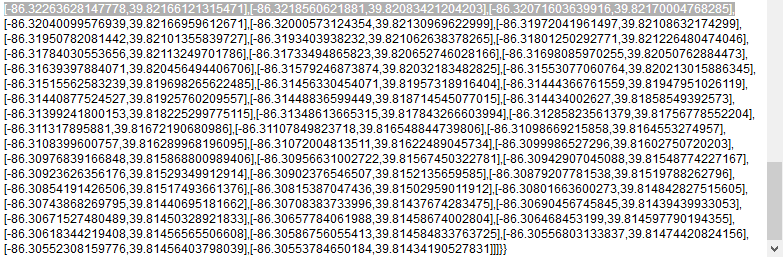




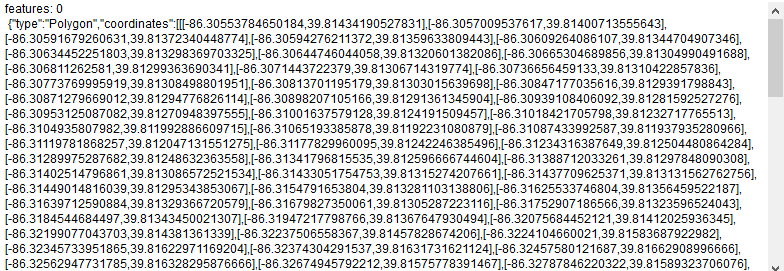
**if** (i==0) alert(**"features: "** + i + **" \n "** + ***JSON***.stringify(***map\_data***.**features**[i]));

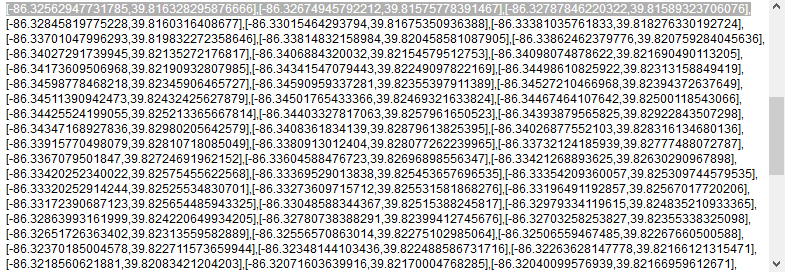






**if** (i==0)alert(**"features: "**+i+**" \n "**+***JSON***.stringify(***map\_data***.**features**[i].**geometry**));





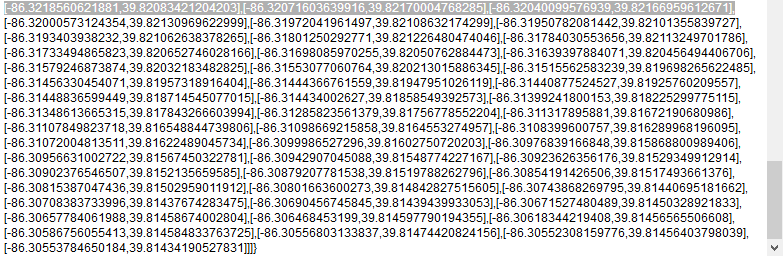


Chart libraries

<https://naver.github.io/billboard.js/demo/#Chart.StackedBarChart>

<https://naver.github.io/billboard.js/>

<https://geoviz.ceoas.oregonstate.edu/neocarto/>

Stack overflow

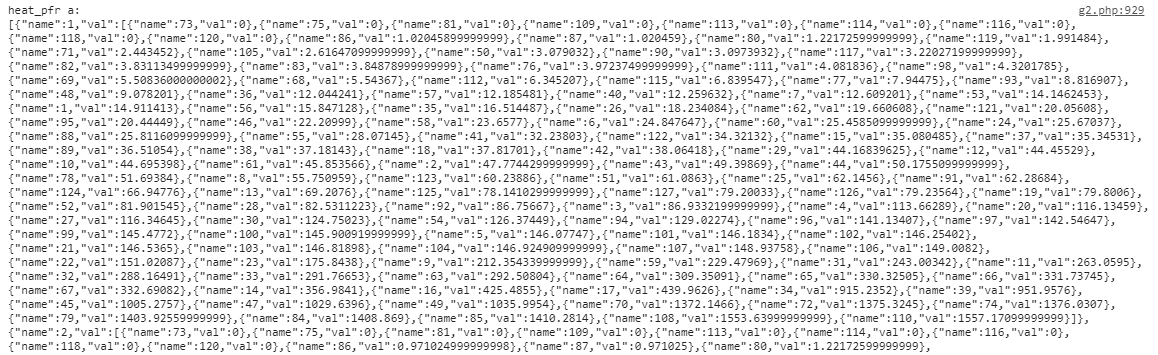
<https://stackoverflow.com/questions/10888958/country-boundries-using-google-map-api-v3/37092260#37092260>

<https://stackoverflow.com/questions/39106230/style-multiple-geojson-files-with-the-google-maps-javascript-api-v3-data-layer/39107656>

chrome does not update

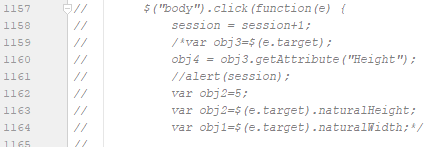
<https://groups.google.com/forum/#!topic/google-chrome-developer-tools/gysw_3qgMMs>

**console**.log (**"heat\_pfr a: \n"** + ***JSON***.stringify (***heatpfra***))

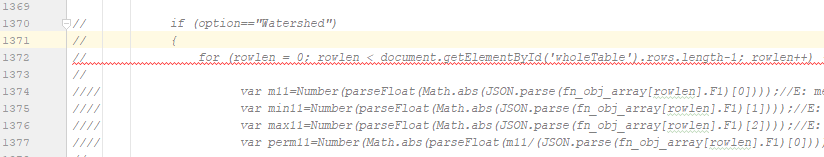


NOT SURE ABOUT THESE CHANGES

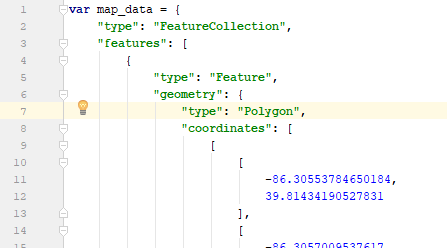
Change 1

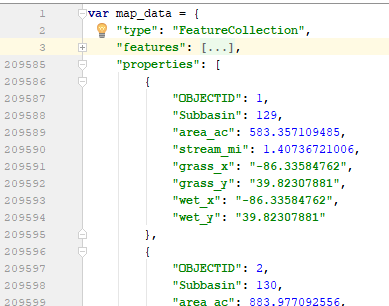


Change 2



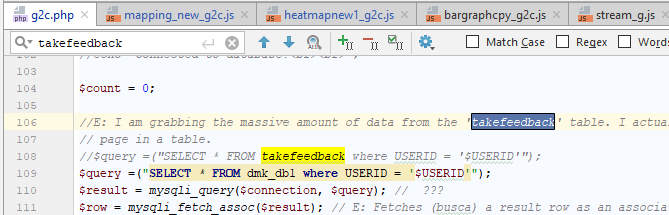
Json for polygons





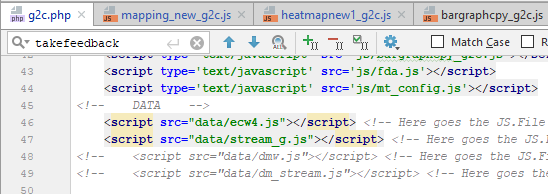
**CHANGES for Dairy McKay**

1. The database (mentioned in MySQL) “takefeedback” for ecw

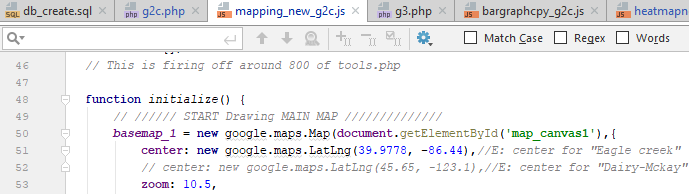


1. DATA (2 lines) L.46 L.47

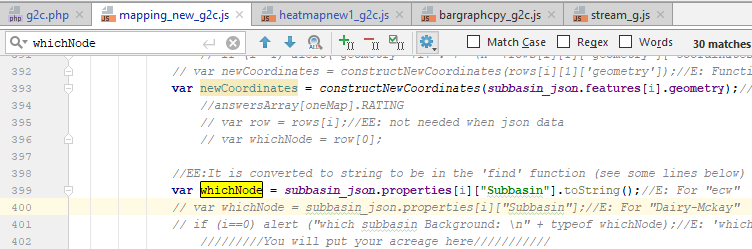
Change the JSON (js) data



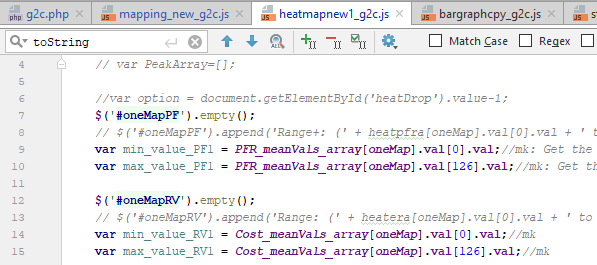
1. Map center position L.51



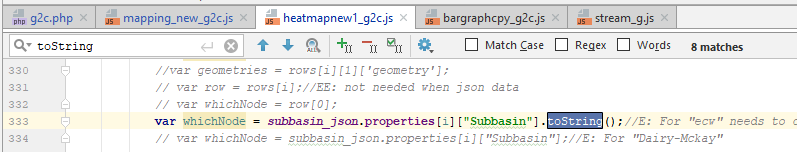
1. The type of variable “toString” (1 line) in “mapping\_new\_g2.js”



1. Change the length of “PFR\_meanVals\_array” from to 126 to 38 (4 times)



1. The type of variable “toString” (8 lines)

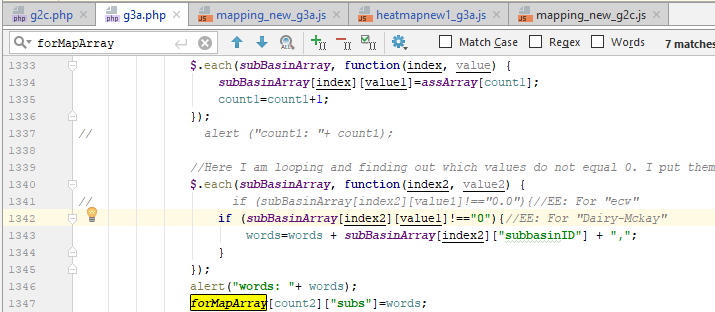


1. Make sure that json data of watershed and stream have the following characteristic:

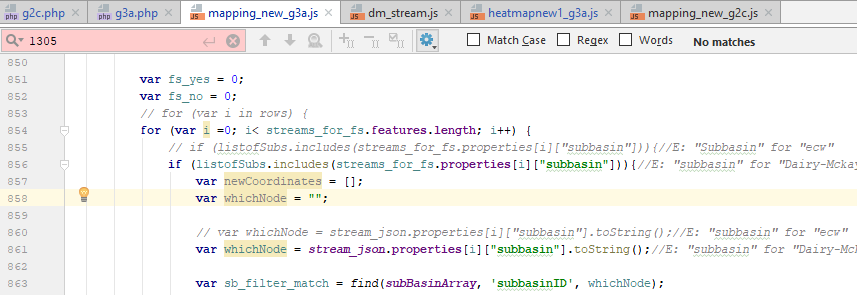
* No “multipolygon”.
* Kip the same name of features, such as: “Subbasin”, “area\_ac”, etc.

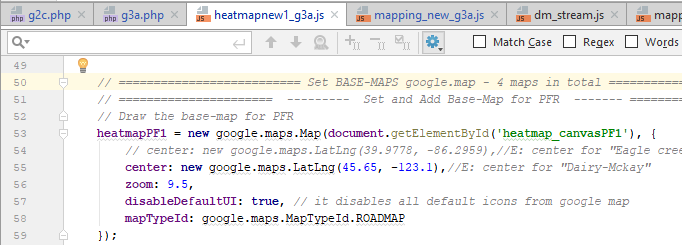
1. Make this change in “g3.php”. “0.0” for “ecw” and just “0” for “Dairy-Mckay”



1. These two lines of code in “mapping\_new.js” are case sensitive. Change “Subbasin” for “ecw” by “subbasin” for “Dairy McKay”, because the JSON data for “ecw” considers “Subbasin” with uppercase and “Dairy McKay” with lowercase



1. Change the center of coordinates of each heatmap in PFR, P, SR, and NR heatmap. The zoom can be 9.5 or 10. See below.



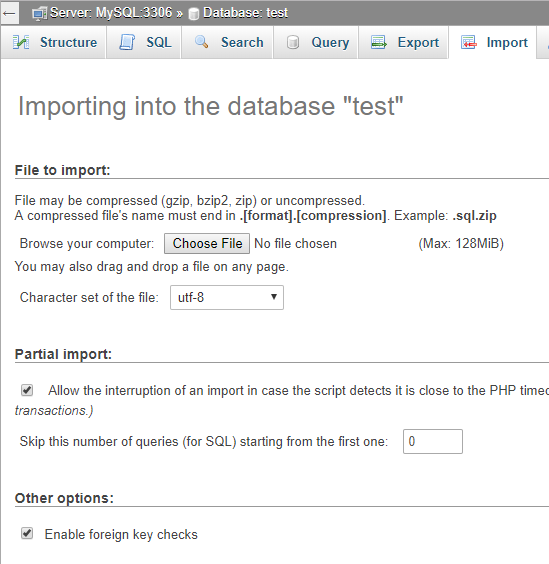
1. dsds

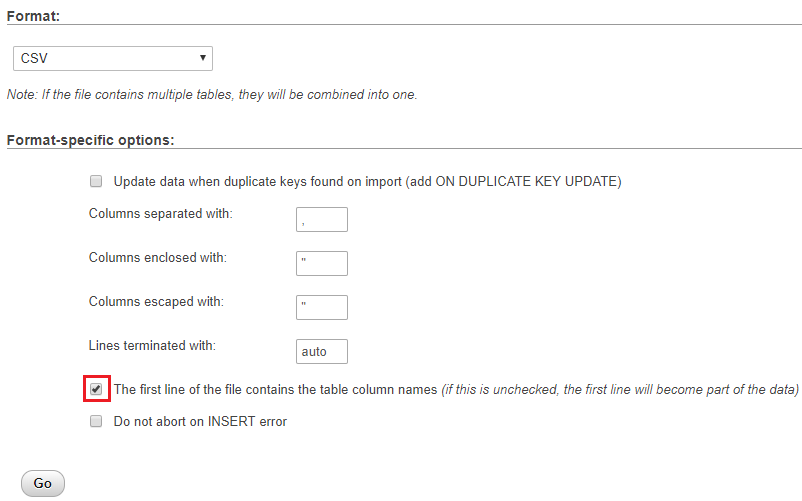
**CHANGES for using JSON Database**

1. In g2.php, Line 917 switch ‘headers’ array calculation
2. In g2.php, Line 938 switch ‘array\_fullvalues’, ‘answerArray’, and ‘fn\_obj\_array’ calculation
3. In g2.php, Line 1044 switch ‘PFR\_meanVals\_array’, ‘Cost\_meanVals\_array’, ‘SR\_meanVals\_array’, and ‘NR\_meanVals\_array’ calculation.
4. In ‘g2.php’, from Line 85 to 197, turn off the <?php part

**How to insert a CSV/EXCEL file into MySQL**

1. Save your excel file as CSV with your desired header
2. Open the MySQL and go to the Database where you want to import the CSV data.
3. Once you are inside of the database, click on “Import”.
4. In the new window, choose your CSV file (upload your CSV file), and check the box “The first of the line contains ….” (see the red box below). By clicking this box, headers of you CSV file will be setup as header on your MySQL table as well.





1. Click “Go”. Then, a table with a default name will be create.
2. Rename the table with a desired name using this query (notice the apostrophe key is located below the `esc` key).

RENAME **TABLE `**tb1` **TO `**tb2`, `tb3` **TO `**tb4`;

VERIFICATION OF STRUCTURE OF DATABASES

NEXT Tasks

1. Run WRESTORE for Dairy-Mckay watershed (done)
2. Set the json file (data.js) into the virtual web (done)
3. Convert the big database DDBB into json (done)
4. Implement Bootstrap
5. Set a Quit option at the end of the alternative 20
6. Insert the user’s rating into MySQL for the alternative in curse.
7. Corrections for DR. Kristen experiment

token

6b40d9a9fb47d07bf794c911c9c9183cb28c5580

Curso de bootstrap para ver classes

<https://www.youtube.com/watch?v=7s1RjItUBqU&index=1&list=PLL0TiOXBeDagsYUYFO9WMwDtuDuoGZVB9>

Use google drive as database and to access through JavaScript

<https://www.youtube.com/watch?v=-7YH6rdR-tk>

<https://bytutorial.com/tutorials/google-api/introduction-to-google-drive-api-using-javascript?fbclid=IwAR3OMDB3FZsG8sQSnH7bbFXuMpgWa3u-oLltqgO5JuqEBwdzOn9pjXwZbFI>

<https://advancedweb.hu/2015/05/26/accessing-google-drive-in-javascript/>

my page

<https://noayarae.github.io/Custom-basemap-leaflet/index.html>

GIT Atlassian-tutorial

<https://www.atlassian.com/git/tutorials/saving-changes/gitignore>

<https://confluence.atlassian.com/bitbucket/checkout-a-branch-into-a-local-repository-313466957.html>

JSON validator

<https://jsonlint.com/>