CHALLENGEs are OPTIONAL

00. Create directory panda/your\_name/hw/hw-2024-06-09

00. Create directory panda/your\_name/projects/<your-nodejs-mysql>

00. Create directory panda/your\_name/projects/09-calculator etc.

0. Projects

00-portfolio - get some ideas, what do you want

00-tic-tac-toe - 30.04.2024

01-is-it-a-prime-number - 30.04.2024

02-css-explanator - 30.04.2024

03-questionnaire - 8.05.2024

04-binary-guessing - 16.05.2024 - small groups

05-string-games - "fix the clock","random quote",

"palindrome" and "gematria" parts - 30.05.2024

06-nim - 23.05.2024 - small groups

07-hangman - 5.06.2024 - small groups

with both "mouse click" and "keyboard" events,

with minimum of the listeners - using

event propagation

08-NodeJS-SQL-Your-Theme - choose your theme and

create database with data

09-calculator - 16.06.2024 - build it with css GRID,

use both "mouse click" and "keyboard" events,

with minimum of the listeners - using

event propagation.

It should perform the actions: +,-,\*,/,%,power,

root, factorial, translating number from binary

to decimal, from decimal to binary, from hexadecimal

to decimal, from decimal to hexadecimal.

Optionally: translating numbers from decimal to

roman and from roman to decimal.

Optionally - more functions.

This project will be evaluated for the themes of:

"algorithms and advanced DOM".

Optional Projects:

- Canvas "Breakout Game"

- "Endless Dangers" - see the description below

- "Dreidel"

- "Memory Game" - see the beginning files and

the example of how to flip card inside

the "projects/optional/memory-game" dir

- "Math Practice" - Optional project of math-practice

with drops containing math exercises and

going down.

When the player types right answer,

the drop should stop or disappear or

make some other response.

- "One page site with smooth scrolling" - like YummyIsrael

1. In parallel with our project "about food"

continue your own project

08-NodeJS-SQL-Your-Theme -

show the page with the form for adding an entity.

Use Cookbook002.

This project in the future is getting evaluated for the theme of

"Basic NodeJS Express MVC CRUD API"

2. Continue the project 05-String-Games,

Add /\* ANAGRAMS \*/ part which will contain

- function getting random word for anagram from

"anagrams" array and showing it.

This function should be triggered

when clicking "New Anagram" button.

- function findAnagrams() finding in the "anagrams" array

all the anagrams of the chosen word

and showing them in the "comp-anagram" input

(remember the alphabetic sorting trick!!!)

- function wich will take the player's guess

(anagrams with commas between them),

compare them to the right list of the anagrams

and respond to the player, like:

This is not an anagram!

You did not find all the anagrams!

Great, you did find all the anagrams!

3. SQL - functions, column aliases, order by:

a) Retrieve customerNames with credit limits rounded

to the nearest whole number (use alias for this column)

b) Concatenate the customer contact's first and last names

along with their phone number and with a comma in between

to look like 'John, Doe, phone: 555-5555'

c) Extract the first three characters of the country name

(SUBSTRING() function)

d) Retreive only the rows for the English speaking countries:

USA, Australia, UK, New Zealand, Canada

e) Retrieve customers from specific cities that their names have

'an' string somewhere inside

f) Retrieve customers within a range of 1370 to 1611 (including)

of sales representative employee numbers or

belonging to a certain set of these numbers: 1165, 1370, 1702

g) Include the ORDER BY clause to specify the sorting order

for any of the above queries: by city - in alphabetic order,

by customerName - in reverse alphabetic order.

4. Callbacks and asynchronous.

a) Write asynchronous function procrastinator() that gets

callback function and delay time (in milliseconds).

It shows Date.now(), writes "I will do it tomorrow"

and uses setTimeout() -

passes to it the callback and the delay.

Use function doLaundry() {

console.log(Date.now())

console.log('Started the LAUNDRY')

setTimeout(()=>{

console.log(Date.now());

console.log('Finished the LAUNDRY')

},1000)

}

Run the function procrastinator() and pass to it

doLaundry() and some delay time.

b) You've got a function:

function getResponseFromPlanet(planet,callback) {

setTimeout(() => {

let responseData = `Response from ${planet}: 1+1=2`

callback(responseData);

},2000)

}

Now run this function: pass to it planet like 'Mars' or 'Venus'

and a callback that should look like an arrow function that gets

message as a parameter and shows it on the console.