

TODO-LIST Documentation

@TODO-List is for :

1- Add (todo/task) with (title/name),

description and created date.

2- Remove task.

3- Edit task name or description .

@Collaborators

Fady Bouez - Hossam Shapan - Peter Abdo

////////////////////////////////////

#We all agreed to think about the structure

write it down.

Then everyone will code it from scratch

and make it work.

After that style and clean it as much as we can.

@The Idea Behind Todo List :

> Using local storage as user database to manipulate todos

@Our Logic

(todo/task) is created in local storage with unique key for each . the unique key is the title that user typed.

this unique key is add as Key in local storage and dynamically as ID to (todo div in html).

@Files

#All files are created from scratch

1- index.html > have only parent div that holds all todos

2- style.css > basic style to achieve the idea

3- style.js > add some basic style with js for buttons

4- getters.js > functions to get data from user or local storage (keys, values)

5- **creators.js** > functions to build the shape of todo based on the data from local storage

6- **main.js** >

have the main functions

<send data to local>

(get data from user then add it to local with **logic**

ex: if user add 2 todos with same name)

<get data from local storage>

everytime body **reloads** this function returning all data from local and paint todos **with creator function** based on how many (key,value) in local storage

<edit description , title>

two functions that handle logic to edit title,description

--editing title will treat todo/task **as new task**

--editing description will **just change value of description**

< clearAll , delete todo >

one for clear **all local storage** ,

one for delete **only one todo**

///

// All logics serve to not repeat a task in local storage

Even if user add 2 identical tasks/todos

(with help of a trick (random number as string to task if duplicated))

-----THANK YOU-----

