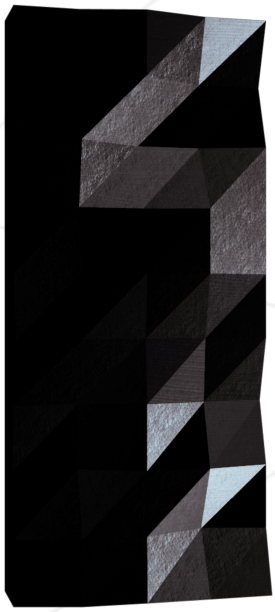


Javascript, jQuery and fun

Fun stuff

Maksud Aghayev maghayev@student.us.org
42 Staff pedago@42.fr

Summary: This document should get you started in writing your on interactive web page, using Javascript and HTML/CSS manipulations

Contents

Chapter I

Foreword

“Don’t Panic.” — Douglas Adams, The Hitchhiker’s Guide to the Galaxy

“Space is big. You just won’t believe how vastly, hugely, mind-bogglingly big it is. I mean, you may think it’s a long way down the road to the chemist’s, but that’s just peanuts to space.” — Douglas Adams, The Hitchhiker’s Guide to the Galaxy

“For a moment, nothing happened. Then, after a second or so, nothing continued to happen.” — Douglas Adams, The Hitchhiker’s Guide to the Galaxy

“You know," said Arthur, "it’s at times like this, when I’m trapped in a Vogon airlock with a man from Betelgeuse, and about to die of asphyxiation in deep space that I really wish I’d listened to what my mother told me when I was young." "Why, what did she tell you?" "I don’t know, I didn’t listen.” — Douglas Adams, The Hitchhiker’s Guide to the Galaxy

A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools. — Douglas Adams, The Hitchhiker’s Guide to the Galaxy

Chapter II

Introduction

JavaScript became very important in web development and making web pages interactive, further down the road JavaScript became quite popular in different areas apart from pure web interactions and evolved into its own ecosystem.
This project will require you to build basic interactions in couple exercises, and later connect your project from HTML/CSS to this one and add quite a bit of interactions to it.

Chapter III

Goals

In this project we will learn basic HTML/CSS manipulations, data management, and add persistent data for complete cycle of front end development.
We will cover vanilla JavaScript as part of jQuery JavaScript library to eliminate some overhead of using vanilla JS. But be aware, this project will require mixing of both to succeed.
In this project we will cover additional topics of persistent storage, event-driven web page and one of 2 the most popular ways to store data, JSON

Chapter IV

General instructions

- You may build on top of your previous project, and we `strongly` encourage extending your previous project but not changing it drastically. But you are free to come up with new project, but then you will have to face requirements of previous project and this one.
- Application of persistent storage can be done in various forms, it is up to you to choose which one to use. You might find pointers to what to look into in different subjects in different sections, or look for direct links to. Be cautions, as it is stated in The Hitchhiker's Guide to the Galaxy:
  "Exactly!" said Deep Thought. "So once you do know what the question actually is, you'll know what the answer means."

Chapter V

Mandatory part

- You are required to use jQuery 3.x, how you include it is up to you. We suggest using jQuery CDN link to version 3.x
- It is mandatory to persist data after refresh, closing browser window or some other none hard resets of the page/browser.
- It is crucial to look into and use appropriately JSON to store, send and use data about each tile, and for bonus user management.

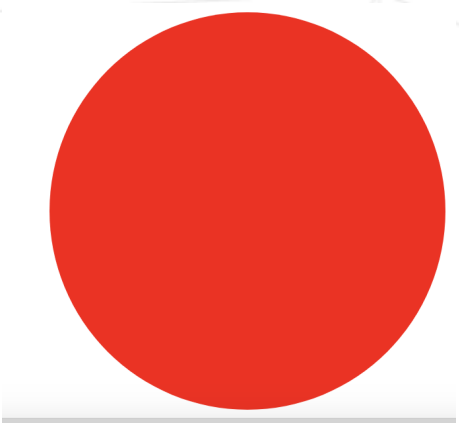
Let's rock'n'roll:

`On page 42 of Harry Potter and the Philosopher's Stone, Harry discovers he's a wizard.`

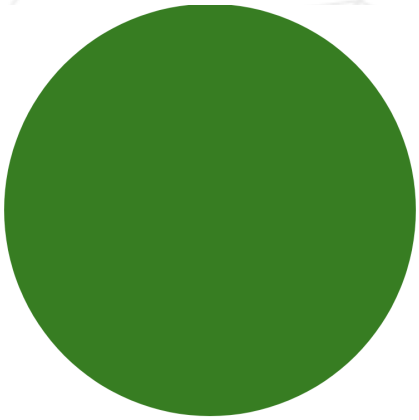
`Exercise 00:`

Folder to turn in: *ex00*
Files to turn in: *index.html*
General look over:
Build a page with just circle on it, initially red.
Each click on the circle will change its color and increase its size by 10px, each mouseleave will decrease its size by 5px and change its color.

- Initial circle has to be 300x300px
- Three colors to use: red, green and blue
- Mouseleave - decrease 5px, change color to next in queue.
- Mouse click on a circle - increase by 10px and change color to next in queue
- If circle reaches 500px, deflate it to 300px
- And vise-versa, if circle is about to hit level below 150px, inflate it to 300px



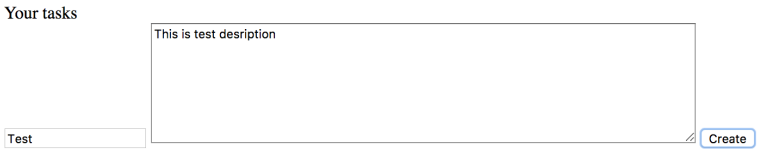
```
width: 300px;
height: 300px;
```

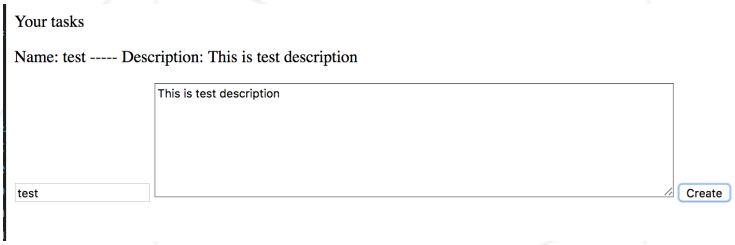


```
width: 360px;
height: 360px;
background-color: green;
```

`Exercise 01`:
Folder to turn in: *ex01*
Files to turn in: *index.html*
General look over:
You will have to build a page, no design needed, that will allow user to add, remove and see all past information in to do list. Yes, yes, we are building basic TO-DO list that will allow user interact with it. In here we already moving our glaze towards JSON, local storage(could be cache, local storage techniques, local DB builds, Cookies - choose your preference, but be wise about your chose).m?mm????

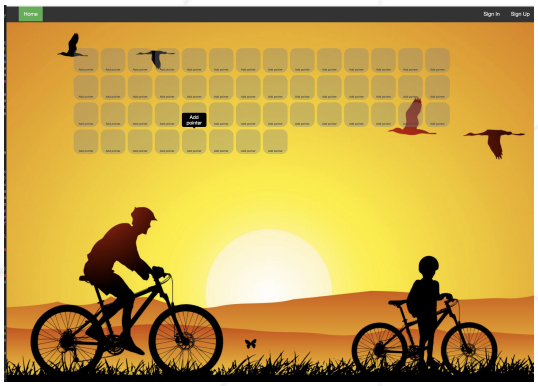
- You have to have a form that will add items to a To Do list
- As soon as item got added, update the list
- As soon as item was deleted, update the list
- You have to persist data even after browser close, window close and page refresh.
- As part of the mandatory part, you will implement on load list display, if any
- And obvious one, you have to handle errors :)

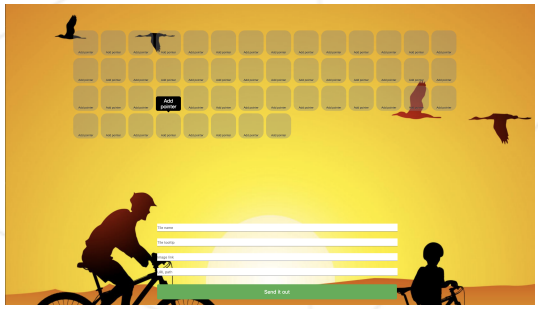


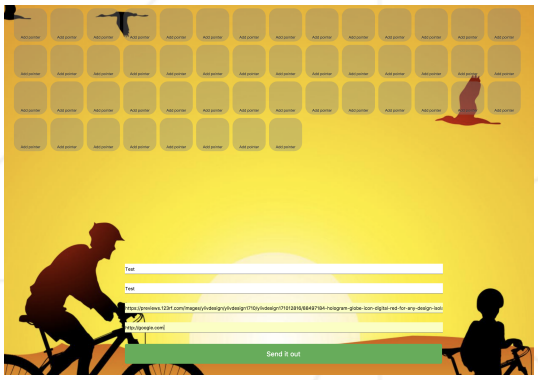


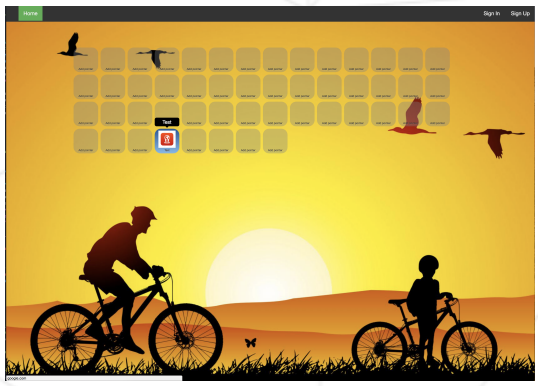
`Exercise 02`:
Folder to turn in: *ex02*
Files to turn in: *
General look over:
Here you will use all knowledge gained from previous exercises to bring you previous project to live action. Whats needs to be done is to allow changes to each cell by adding data into it, be able to change each cell that is filled already.

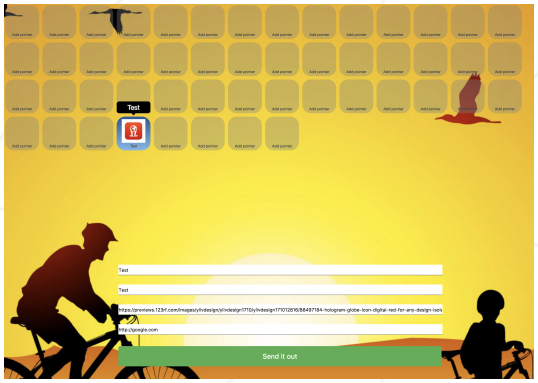
- Left click on a grid item has to bring forth a form that will have at least
  - Name text of an item
  - Tooltip text for an item
  - Link to icon, web based or local file
- Right click, or contextmenu, have to bring a form that is pre-filled with data specified in previous point
- As soon as item got added, update the list
- As soon as item was edited, update the list
- You have to persist data even after browser close, window close and page refresh.
- As part of the mandatory part, you will implement on load list display, if any
- And obvious one, you have to handle errors :)

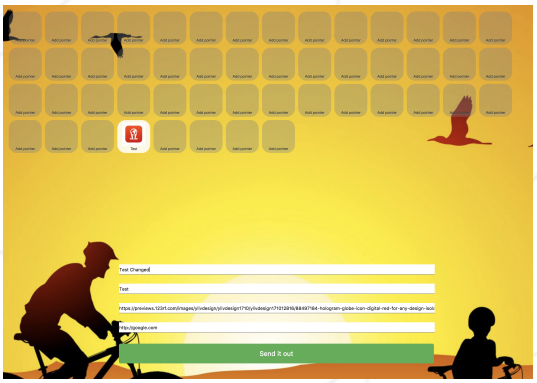









Test
Test
Test
Send it out


Test Changed
Test
Send it out

Chapter VI

Bonus part

Once you have completed whole project and met all requirements you may start working on bonus part.
This part of the project will look into user management and management of content based on user.
Folder to turn in: *bonus*
Files to turn in: *
So closer to the task:

Extend your exercise 3 and add login, registration and log out systems.
Users shall not be able to access main content of the site without login. How you will handle registration and what happens right after user registered, be it redirect to main content or to login page, is solely up to you.
You have to meet this criteria to get bonus:

- You have to build login page with two fields: `username` and `password`
- You have to build registration page, which will consist of same fields as login.
- Log out should log out current signed user and display login/registration page.
- Each user should be able to maintain his/her data of each grid item, depending what your grid items do.

`You may use single interactive page with a lot of mouse listeners to handle each case separately, but be smart about it, it is really easy to misplace click event handling. As follow up this jQuery has to be incorporated into this, how is up to you.`

Chapter VII

Turn-in and peer-evaluation

This part describes the conditions and instructions regarding the turn-in and the peer-evaluation of the project. If your project does not require odd turn-in or peer-evaluation instructions, feel free to use the following paragraph as it is:

Turn your work in using your `GiT` repository, as usual. Only work present on your repository will be graded in defense.