100 ways Todo list

Different ways a todo list programmed.
Mostly for the purpose to try out different languages, and techniques, whether they are right or not. To reflect on pros/cons of the approach. To use a portfolio kind of work.

(some places might label this as Todo100)

Words:

- Todo, todo task,
- list

Basic Requirements

- **Create - View - Complete**
- Create a todo task
- View all todo tasks
- Complete a task

Basic workflow:

- Start page: show all, and means to “create new”
- Action create new: show free text input
    - Controls:
        - Text input
        - Ok/save/yes - to add task to list
        - cancel/no/discard - to cancel out of “create new”
- List view:

Oddly not required (and if implemented will be marked as bonus feature)

- Long term saving - basic requirements will just be temporary list, as many different ways to save which will be part of a bonus feature.
- Complete task but keep in list. Means to this will be part of bonus features

HTML + Javascript - first round 2019-05

- How i most likely would make a minimal todo html client only
- Tags:
    - No-style
    - Code in javascript, html in html
    - temp-storage
-

Mostly javascript

- Tags: minimal requirements
    - Javascript
    - No-style
- “Mostly”: html used to retrieve javascript file and document page
- Aims: pure javascript (es6, 2019)

HTML Form

jQuery Minimal

- Tags: minimal requirements
    - Javascript
    - no-style
- For comparison with “mostly javascript” as to amount of saved lines of code needing to recreate vs how much bloat it adds.

Mostly javascript + remote save

jQuery AJAX/Remote Save

jQuery UI

Tests - TDD first

Code First database

Java server

c#.net server

Node.js server

Full Javascript (client,

Knockout client

Angular client