**Тестовые задания по HTML, CSS и JS**

**Задание #1**

Используя только HTML и CSS (без картинок) нужно изобразить на одной странице флаги государств:

*- Украины,*

*- Японии,*

*- Исландии,*

*- Чехии.*

Флаги размещаем один под другим с небольшими отступами, размер каждого флага 300 пикселей в ширину. Важно соблюсти пропорции между элементами флага.

**Задание #2**

**HTML+CSS:**

1) Расположите на одной странице изображения из папки task-2 в виде сетки 4 на 3 с равномерными отступами между ними. Область контента с картинками центрирована и занимает 80% ширины экрана.

2) Добавьте к изображениям тёмно-серую обводку толщиной 5 пикселей.

3) Обводка первого изображения в каждом ряду - красного цвета.

4) "Подрезаем" изображения так, чтобы они стали круглыми.

5) При наведении на любую картинку обводка должна становится жёлтой, а изображение поворачиваться на угол 45 градусов.

6) При ширинах экрана 900 пикселей и меньше предложите свою версию адаптивной вёрстки страницы с перестраиванием изображений по 2 в ряд, а при ширинах 500 пикселей и меньше - по 1 в ряд.

**JS:**

7) Напишите скрипт, который будет подсчитывать количество картинок на странице и выводить вверху страницы их количество и актуальную дату в формате ДД.ММ.ГГГГ ЧЧ:ММ.

8\*) При клике на каждое изображение в сетке:

- во всплывающем окне поверх всего контента появляется центрированное по-вертикали и горизонтали изображение в полном размере.

- также поверх контента, однако под большим изображением должен быть полупрозрачный затемняющий фон.

- рядом с большим изображением должна быть кнопка, при клике на которую большая картинка скрывается.

*\* в этом задании не рекомендуется использовать плагины всплывающих фотогалерей, явным образом решающие эту задачу.*

**BONUS:**

9) Под каждым изображением в сетке размещаем маленький крестик.

- при клике на него соответствующее изображение удаляется из общего списка.

- когда пользователь удаляет картинки, сохраняем данные о том, что было удалено, и после перезагрузки страницы удалённые ранее картинки должны отстуствовать на странице. Способ хранения данных - на ваше усмотрение.

- внизу под всеми изображениями размещаем кнопку "Восстановить", которая делает все ранее скрытые пользователем картинки видимыми.

**Задание #3**

Небольшое веб-приложение, которое выводит на экран данные, полученные со стороннего API.

Источник данных: <http://universities.hipolabs.com>, это данные по университетам разных стран. По ссылке <https://github.com/Hipo/university-domains-list> описано, как пользоваться этим API.

**Суть приложения в следующем:**

1) Создайте страницу, которая содержит текствое поле, кнопку «Отправить» и кнопку «Сброс».

2) В текстовом поле человек может написать латиницей интересующую его страну (*например: ukraine*) и нажать на кнопку отправки. После этого происходит запрос, и на основе полученных данных, под формой строится таблица с результатами, соответствующая полям, которые есть в json-файле.

3) Помимо полей из полученного json-файла, в таблице первой должна быть колонка с порядковым номером строки.

4) Колонка со ссылкой на сайт университета должна показывать ссылки не текстом, а именно ссылкой.

5) Кнопка «Сброс» очищает текстовое поле и удаляет со страницы таблицу с данными, возвращая страницу в исходный вид.

**BONUS:**

6) В каждом ряду таблицы добавить последней ячейкой чекбокс "Сохранить в мой список". Вверху страницы добавить счётчик, который показывает количество отмеченных чекбоксов.

7) Сохранять состояние страницы, чтобы при перезагрузке сохранялась таблица с результатами поиска, отмеченные чекбоксы и значение счётчика. Способ хранения данных - на ваше усмотрение.

**Дополнительная информация:**

- каждое задание сохраняйте в отдельной папке task-1, task-2, task-3. А все их потом поместите в zip-архив, который назовите латиницей как ваше ***имя\_фамилия.zip***

- фон документа - белый, остальные цвета, оттенки и внешний вид элементов, выбор дополнительных виджетов и стилистики оформления - на ваше усмотрение.

- в заданиях 2 и 3 разрешается использование библиотек и фреймворков широкого назначения (React, Angular, jQuery и др.).

- вместо обычного js-кода допустимо использование стандарта ES6 либо же TypeScript

- структура дополнительно созданных в рамках задания папок и файлов, их количество, а также дополнетельный инструмментарий разработки (сборщики, менеджеры пакетов, компиляторы и др.) - на ваше усмотрение.

- в случае необходимости сборки или компиляции веб-приложения, опишите краткий readme-файл с описанием чётких шагов для его запуска.

**Приветствуется:**

- чистый и хорошо структурированный и докумментированный код.

- рациональное использование html-тегов.

- плавные ховер-эффекты, интересная анимация и др. "фишки", которые по-вашему мнению будет уместно добавить к заданиям.

- кроссбраузерность приведённого решения, включая поддержку мобильных устройств и старые версии браузеров.