**Проверочная работа.**

**Языки: HTML, CSS, JS.**

**Темы: Семантика и семантические теги. Форматирование и выделение текста. Шрифты. Списки. Ссылки. Разметка: Grid и Flexbox. Позиционирование. Таблицы. HTML-формы. CSS Animation. JS Animation. Выравнивание; justify- и align-: content, items, self. Адаптивная вёрстка. Drag and Drop. Массивы JS.**  
Создайте сайт, соблюдая нижеперечисленные критерии. Использование старых работ, методических указаний и gpt разрешены. Всё, что собрано тут, уже частично реализовано в предыдущих работах.

1. **Структура.**  
   Помимо div и span, используйте семантические теги, такие как header, footer, main, aside, nav, section, article. Используйте их правильно. Внутри каждого из тегов подпишите, для чего он используется.
2. **Работа с текстом.**  
   Создайте блок. В блоке опишите каждый существующий тег для работы с текстом и форматирования текста, используя этот тег там, где о нём рассказывается. Теги представлены ниже:

**Структура**

[h1](https://htmlbase.ru/html/h1), [h2](https://htmlbase.ru/html/h2), [h3](https://htmlbase.ru/html/h3), [h4](https://htmlbase.ru/html/h4), [h5](https://htmlbase.ru/html/h5), [h6](https://htmlbase.ru/html/h6) - блочные теги предназначены для указания заголовков. Заголовков в html есть 6 уровней, от h1 (первый уровень) до h6 (последний уровень), каждый визуально выделен жирным шрифтом и его соответствующим размером. Заголовки должны описывать содержание, это положительно влияет на оптимизацию сайта. Хорошей практикой считается указывать только один заголовок первого уровня h1, остальных может быть несколько. Заголовки должны указываться по иерархии, на первом уровне заголовок h1 дальше подзаголовок h2 и так далее

[p](https://htmlbase.ru/html/p) - абзац. Блочный тег, разделяет текст на отдельные абзацы, в конце добавляется отступ

Примеры: <https://codepen.io/Yury_P/pen/WNGLgdW>

**Форматирование**

[b](https://htmlbase.ru/html/b) - визуально выделяет текст жирным начертанием

[strong](https://htmlbase.ru/html/strong) - отображает текст жирным начертанием, логически указывает на важность текста. Может быть использован для выделения ключевых слов для оптимизации поиска

[i](https://htmlbase.ru/html/i) - задает курсивное начертание текста

[em](https://htmlbase.ru/html/em) - задает курсивное начертание текста, логически указывает на важность текста

[u](https://htmlbase.ru/html/u) - подчеркивает текст снизу

[pre](https://htmlbase.ru/html/pre) - сохраняет форматирование текста (не удаляет пробелы и переносы строк)

[sup](https://htmlbase.ru/html/sup) - текст отображается как верхний индекс

[sub](https://htmlbase.ru/html/sub) - текст отображается как нижний индекс

[small](https://htmlbase.ru/html/small) - уменьшает текст на условную единицу

[address](https://htmlbase.ru/html/address) - внутри указывается адрес (физический адрес, телефон, почта), обычно отображается курсивным шрифтом

[mark](https://htmlbase.ru/html/mark) - отображает текст как выделенный

[abbr](https://htmlbase.ru/html/abbr) - оборачивает аббревиатуру (сокращение), в атрибуте title указывается ее расшифровка, которая показывается при наведении

[kbd](https://htmlbase.ru/html/kbd) - отображает название клавиш или текст набранный пользователем, обычно отображается моноширинным шрифтом

[dfn](https://htmlbase.ru/html/dfn) - внутри пишется название нового термина, далее идет его пояснение, обычно отображается курсивным шрифтом

[ins](https://htmlbase.ru/html/ins) - выделяет новый текст, который был добавлен при обновлении содержания, обычно отображается как подчеркнутый текст

[del](https://htmlbase.ru/html/del) - выделяет старый текст, который был удален при обновлении содержания, обычно отображается как зачеркнутый текст

[s](https://htmlbase.ru/html/s) - зачеркивает текст

Примеры: <https://codepen.io/dmitryvalak/pen/zYKXjaV>

**Цитаты**

[q](https://htmlbase.ru/html/q) - отображает текст как цитату, обрамляет его кавычками

[blockquote](https://htmlbase.ru/html/blockquote) - отображает текст как цитату. По сравнению с тегом используется для длинных цитат, оформляется отступами

[cite](https://htmlbase.ru/html/cite) - выделяет цитату на материал из другого источника, обычно отображается курсивным шрифтом

Примеры: <https://codepen.io/dmitryvalak/pen/yLarjoq>

**Списки**

[ul](https://htmlbase.ru/html/ul) - маркированный список. Каждый элемент списка оборачивается тегами [li](https://htmlbase.ru/html/li)

[ol](https://htmlbase.ru/html/ol) - нумерованный список. Каждый элемент списка оборачивается тегами [li](https://htmlbase.ru/html/li)

[dl](https://htmlbase.ru/html/dl) - список терминов. Внутри используются теги: [dt](https://htmlbase.ru/html/dt) - название термина, [dd](https://htmlbase.ru/html/dd) - описание термина

Примеры: <https://codepen.io/Yury_P/pen/mdrgZQb>

**Отображение кода**

[code](https://htmlbase.ru/html/code) - оборачивает программный код, обычно отображается моноширинным шрифтом, не учитывает переносы строк и дополнительные пробелы

[var](https://htmlbase.ru/html/var) - оборачивает переменную из программного кода, обычно отображается моноширинным шрифтом

[samp](https://htmlbase.ru/html/samp) - оборачивает текст, которые является результатом вывода скрипта или программы, обычно отображается моноширинным шрифтом

Примеры: <https://codepen.io/Yury_P/pen/yLaGxPB>

**Другие теги**

[span](https://htmlbase.ru/html/span) - строчный тег. Сам по себе не имеет никакого логического или визуального смысла. Часто используется внутри абзаца (

) для выделения части текста. Если часть текста нужно стилизовать по другому, ее можно обернуть в span, указать класс и в css добавить нужные стили

[br](https://htmlbase.ru/html/br) - переносит текст на новую строку

[wbr](https://htmlbase.ru/html/wbr) - указывает где можно сделать перенос слова

[hr](https://htmlbase.ru/html/hr) - вставляет горизонтальную полосу

bdo - задает направление текста, в атрибуте dir указывается направление: rtl - справа на лево, ltr - слева направо

bdi - запрещает изменение направления текста, по аналогии с bdo направление указывается в атрибуте dir, по умолчанию значение auto

Примеры: <https://codepen.io/Yury_P/pen/rNMoZrL>

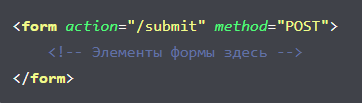
1. **Блок ссылок.**   
   Создайте блок ссылок.   
   1) Первая ссылка будет отправлять на вторую страницу вашего сайта. Для этого необходимо создать еще одну страницу .html и указать её в специальном атрибуте ссылки.  
   2) Вторая ссылка будет открывать сторонний сайт в новой вкладке. Для этого необходимо указать специальный атрибут ссылки.  
   3) Третья ссылка будет при нажатии на неё переносить к указанному блоку на странице (Ссылка якорь). Для этого необходимо через «#» указать идентификатор нужного вам блока.
2. **Необходимо сбросить дефолтные стили CSS.**
3. **Разметка: Grid и Flexbox.**  
   Необходимо сделать основную разметку страницы с помощью CSS Grid (display: grid;), а содержимое блоков поменьше позиционировать с помощью Flexbox (display: flex;).
4. **Flex-flow.**  
   Содержимое блоков поменьше из задания выше, должно визуально отображать свойства flexbox, такие как: flex-direction и flex-wrap (можно использовать flex-flow, чтобы объединить 2 этих свойства).
5. **Position.**  
   Необходимо сделать фиксированный блок (fixed) в правом нижнем углу экрана, а также необходимо баннер, который ведет себя сначала как обычный элемент, потом на какое-то время фиксируется (sticky), потом пропадает (display: none;) (на выбор, либо пропадает сам, либо по нажатию на закрывающую кнопку на нём).
6. **Таблицы и объединения ячеек.**  
   В один из блоков добавьте таблицу и объединением ячеек и столбцов повторите следующий рисунок. После этого стилизуйте таблицу с помощью CSS.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| **2** |  |  |  |  |  |  |  |  |  |  |
| **3** |  |  |  |  |  | |  |  |  |  |
| **4** |  |  |  | |  | |  |  |
| **5** |  |  |  |  |  |  |  |  |
| **6** |  |  |  |  |  |  |
| **7** |  |  |  | |  | |  |  |
| **8** |  |  |  |  |  |  |  |  |
| **9** |  |  |  |  |  |  |  |  |  |  |

1. **CSS Animation**  
   Необходимо пошагово (@keyframes) анимировать какую-либо фигуру, любой тип анимации.
2. **HTML-формы.**  
   Реализуйте форму, которая будет содержать все поля, которые перечислены ниже. После этого, реализуйте функционал кнопки submit, по нажатию на которую будет появляться всплывающее окно со всеми данными из заполненной формы.  
     
   HTML-формы используются для сбора данных от пользователей. Они могут содержать различные элементы, такие как текстовые поля, кнопки, флажки и выпадающие списки.  
     
   Основная структура формы:

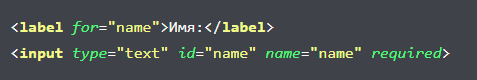
Форма создается с помощью тега <form>. Основные атрибуты:

1. action: URL, куда будут отправлены данные формы.
2. method: метод отправки данных (GET или POST).

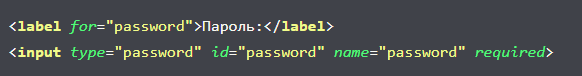


**Текстовые поля**

Используются для ввода текста.

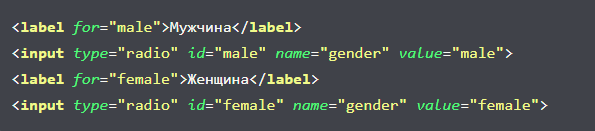


Для ввода пароля используйте следующее:



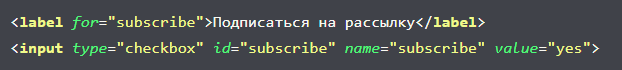
**Радио-кнопки**

Для выбора одного варианта из нескольких.



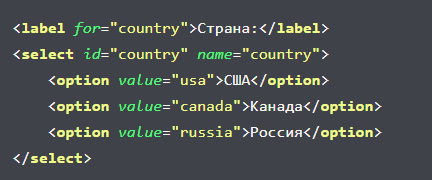
**Чекбоксы**

Для выбора нескольких вариантов.



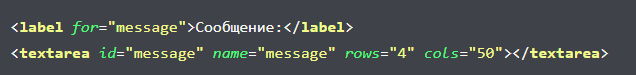
**Выпадающие списки**

Для выбора одного варианта из списка.



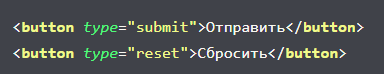
**Многострочное текстовое поле**

Для ввода большого объема текста.



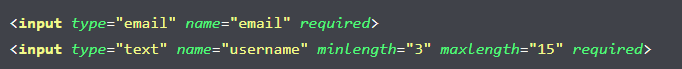
**Кнопки**

Для отправки формы или выполнения других действий.



**Валидация форм**

HTML5 поддерживает встроенную валидацию форм. Используйте атрибуты, такие как required, minlength, maxlength, pattern и другие.



**JS Animation.**  
Необходимо добавить любую js анимацию, и в коде комментариями подписать каждое действие.

1. **Шрифты.**  
   Подключите любой шрифт к сайту, и пусть весь текст на странице будет с этим шрифтом.
2. **Выравнивание; justify- и align-: content, items, self.**Поработайте с группами justify- и align-, и наглядно продемонстрируйте в одном из блоков, что с помощью них можно делать. Также подпишите каждое из свойств и дайте им определение (justify- и align-: content, items, self).
3. **Адаптивная вёрстка (@media)**
4. **Drag and Drop.**  
   Реализуйте drag and drop функцию, которая будет выполнять следующую задачу: при переносе изображения в drag&drop поле, какому-либо из блоков web-страницы задавалось свойство background-image с этим изображением.
5. **Массивы JS.**  
   Необходимо добавить на web-страницу кнопку, по нажатию на которую должен создаваться массив элементов (любых символов), после чего этот массив должен плавно вращаться по кругу, останавливаясь через 3-5 секунд. В alert (всплывающее уведомление) должно выводиться самое верхнее значение (находящееся сверху круга).