Контрольные вопросы

1 Семантические нюансы и архитектурные принципы:

<!DOCTYPE html>: Декларация, провозглашающая версию HTML5 и инициализирующая стандартизированный парсинг.

<html>: Фундаментальный контейнер, определяющий язык документа для интернационализации.

<head>: Репозиторий метаданных, включающий кодировку, адаптивность и ссылки на внешние ресурсы.

<body>: Арена визуального представления, где разворачивается пользовательский интерфейс.

Семантические дескрипторы (<header>, <main>, <section>, <footer>): Инструменты для структурирования контента, повышающие когнитивную доступность и ранжирование в поисковых системах.

2 HTML5 ввёл множество семантических элементов, которые улучшают структуру веб-страниц, делая их более понятными для браузеров, поисковых систем и вспомогательных технологий (например, скрин-ридеров). Эти элементы заменяют универсальные <div> и <span>, добавляя смысл к контенту. Вот ключевые новые элементы (с краткими описаниями):

**<article>**: Представляет независимый контент, такой как статья, пост в блоге или новость. Может содержать заголовок, текст и медиа.

**<aside>**: Определяет боковую панель или дополнительную информацию, не являющуюся основным контентом (например, реклама, ссылки или цитаты).

**<details>**: Создаёт интерактивный элемент с раскрывающимся содержимым (аналог аккордеона). Используется с <summary> для заголовка.

**<figcaption>**: Подпись к изображению или медиа-элементу, обычно внутри <figure>.

**<figure>**: Группирует медиа-контент (изображения, видео) с подписью (<figcaption>), отделяя его от основного текста.

**<footer>**: Нижний колонтитул страницы, раздела или статьи (автор, копирайт, ссылки).

**<header>**: Верхний колонтитул страницы, раздела или статьи (логотип, навигация, заголовки).

**<main>**: Основной контент страницы (должен быть уникальным, без повторяющихся элементов вроде навигации).

**<mark>**: Выделяет текст для акцента (например, подсветка поиска), не меняя его смысл.

**<nav>**: Навигационные ссылки (меню, оглавление).

**<section>**: Группирует тематически связанный контент (разделы страницы, главы).

**<summary>**: Заголовок для <details>, который можно кликнуть для раскрытия.

**<time>**: Представляет дату/время в машиночитаемом формате (например, <time datetime="2023-10-01">1 октября 2023</time>).

3 **<mark>**: Выделяет текст, который имеет особую релевантность в контексте (например, результаты поиска или важные фразы). Браузеры часто подсвечивают его жёлтым фоном.  
Пример: <p>В HTML5 появился элемент <mark>mark</mark> для выделения.</p>

**<time>**: Представляет дату, время или продолжительность в машиночитаемом формате. Помогает поисковикам и календарям понимать временные данные.  
Пример: <p>Событие состоится <time datetime="2023-12-25">25 декабря 2023</time>.</p>

**<bdi>**: Изолирует bidirectional текст (для языков с разным направлением письма, как арабский или иврит внутри латиницы), предотвращая неправильное выравнивание.  
Пример: <p>Привет, <bdi>الاسم</bdi>! (Имя на арабском не сломается).</p>

**<bdo>**: Переопределяет направление текста (LTR или RTL), полезно для демонстрации в многоязычном контенте.  
Пример: <p><bdo dir="rtl">Это текст справа налево</bdo>.</p>

**<wbr>**: Указывает возможную точку разрыва слова (word break opportunity), чтобы текст корректно переносился в узких контейнерах, без добавления пробелов.  
Пример: <p>Супердлинноеслово<wbr>длятестирования.</p> (может сломаться после "слово").

**<ins>**: Обозначает вставленный текст (например, в редактируемом документе), подразумевая добавление. Часто стилизуется подчёркиванием.  
Пример: <p>Оригинал: <del>старый</del><ins>новый</ins> текст.</p>

**<del>**: Обозначает удалённый или устаревший текст (например, в версиях документов). Часто стилизуется зачёркиванием.  
Пример: <p>Оригинал: <del>старый</del><ins>новый</ins> текст.</p>

**<ruby>, <rt>, <rp>**: Набор для ruby-аннотаций (используется в восточноазиатских языках для добавления произношения или переводов над/под текстом). <ruby> — контейнер, <rt> — аннотация, <rp> — скобки для браузеров без поддержки.  
Пример: <ruby>漢<rt>かん</rt><rp>(</rp>じ<rp>)</rp></ruby> (для японского: "漢" с чтением "kan").

4 **<audio>**: Встраивает аудиофайлы (MP3, OGG, WAV и др.) с возможностью воспроизведения. Поддерживает fallback-контент для старых браузеров.  
Пример: <audio controls><source src="song.mp3" type="audio/mpeg"><source src="song.ogg" type="audio/ogg">Ваш браузер не поддерживает аудио.</audio>  
(Добавляет элементы <source> для множественных форматов.)

**<video>**: Встраивает видеофайлы (MP4, WebM, OGG и др.) с элементами управления. Аналогично <audio>, с атрибутами для ширины/высоты.  
Пример: <video width="320" height="240" controls><source src="movie.mp4" type="video/mp4"><source src="movie.webm" type="video/webm">Ваш браузер не поддерживает видео.</video>  
(Использует <source> для альтернативных источников.)

**<source>**: Вспомогательный элемент внутри <audio> или <video> для указания медиа-источников (файлы, URL) с типами MIME. Позволяет браузерам выбирать подходящий формат.  
Пример: См. выше в <audio> и <video>.

**<embed>**: Встраивает внешний контент, такой как плагины (Flash, PDF, SVG) или медиа. Более простой, чем <object>, но менее семантический.  
Пример: <embed src="plugin.swf" type="application/x-shockwave-flash" width="400" height="300"> (Встраивает Flash-плагин.)

**<object>**: Встраивает объекты (плагины, апплеты, документы) с fallback-контентом. Более гибкий, чем <embed>, поддерживает параметры через <param>.  
Пример: <object data="plugin.swf" type="application/x-shockwave-flash"><param name="quality" value="high">Fallback: <a href="plugin.swf">Скачать плагин</a></object>

5 **Static (статическое)**: Это значение по умолчанию для всех элементов. Элемент размещается в обычном потоке документа (normal flow). Никакие отступы не применяются автоматически; элемент ведет себя как обычный блок или инлайновый элемент. Не влияет на другие элементы.

**Relative (относительное)**: Элемент позиционируется относительно его обычного положения. Можно использовать свойства top, right, bottom, left для смещения. Внешний вид потока страницы не нарушается — пространство для элемента остается на своем месте, но сам элемент сдвигается. Полезно для тонкой настройки позиции без влияния на другие элементы.

**Absolute (абсолютное)**: Элемент удаляется из обычного потока и позиционируется относительно ближайшего родителя с позиционированием (не static). Если такого нет, то относительно viewport. Используются top, right, bottom, left для точного размещения. Элемент может накладываться на другие, игнорируя их пространство.

**Fixed (фиксированное)**: Элемент фиксируется относительно области просмотра (viewport). Когда пользователь прокручивает страницу, элемент остается на месте. Часто используется для навигационных меню или sticky-баров. Analogично абсолютному, но привязка к окну браузера.

**Sticky (липкое)**: Комбинация relative и fixed. Элемент ведет себя как relative, пока не erreicht границы контейнера, после чего становится fixed. Полезно для заголовков, которые "прилипают" при прокрутке. Поддержка браузерами появилась позже (с 2017 года), но теперь распространена.

**6** Использование CSS-свойства position

**Relative для родителя + absolute для дочернего**: Родитель устанавливается как относительная точка отсчета, а дочерний элемент сдвигается относительно него. Это позволяет точно размещать элементы, игнорируя обычный поток.

**Flexbox (гибкие контейнеры)**

Родитель превращается в гибкий контейнер, где дочерние элементы выстраиваются вдоль оси. Можно задавать выравнивание по горизонтали (например, по центру) и вертикали, а также менять порядок без изменения HTML.

**CSS Grid (сетки)**

Родитель становится сеткой с строками и колонками. Дочерние элементы автоматически заполняют ячейки, и можно точно указывать их позиции в сетке для создания сложных макетов.

**Обычный поток документа**

Элементы следуют естественному порядку: блоки друг за другом, inline — в строке. Для управления используются отступы (margins) и внутренние поля (padding) родителя для создания пространства внутри.

**Комбинация с дополнительными свойствами**

Z-index управляет наложением: элементы с большим значением перекрывают другие.

Box-sizing включает внутренние поля и границы в расчеты размера контейнера, упрощая позиционирование.

7 **<form> (форма)**: Корневой контейнер, который группирует все элементы формы. Определяет action (куда отправлять данные) и method (как отправлять — GET или POST).

**<input> (поле ввода)**: Основной элемент для ввода текста, паролей, чисел и т.д. Имеет типы: text (текст), password (пароль), email, number, checkbox (флажок), radio (переключатель) и другие.

**<textarea> (текстовая область)**: Многострочное поле для ввода текста, например, комментариев. Размер можно регулировать атрибутами rows и cols.

**<button> (кнопка)**: Элемент для запуска действий, таких как отправка формы (type="submit") или сброс (type="reset"). Может содержать текст или изображения.

**<select> (выпадающий список)**: Позволяет выбрать один или несколько вариантов из списка. Внутри размещаются элементы <option>.

**<option> (вариант выбора)**: Определяет отдельный элемент в <select>. Атрибут selected делает его выбранным по умолчанию.

**<label> (метка)**: Текстовое описание элемента формы, связанное с ним через атрибут for. Обеспечивает доступность и улучшает UX.

**<fieldset> (группа полей)**: Группирует связанные элементы формы для лучшей организации. Часто используется с <legend> для заголовка группы.

**<legend> (заголовок группы)**: Описательный текст для <fieldset>, который визуально выделяет группу элементов.

**<datalist> (список данных)**: Предоставляет список предопределенных значений для autocomplete в <input>. Атрибут list связывает их.

**<output> (вывод результата)**: Отображает результаты вычислений или других операций в форме, обновляемые динамически.

8 accesskey: Определяет клавишу-горячую для фокуса на элементе (например, для навигации с клавиатуры).

autocapitalize: Управляет авто-капитализацией текста в полях ввода (слова, предложения и т.д.).

autofocus: Элемент автоматически получает фокус при загрузке страницы (только для одного элемента).

class: Присваивает CSS-классы для стилизации.

contenteditable: Разрешает редактирование содержимого элемента пользователем (true/false).

data-\*: Пользовательские атрибуты для хранения данных (например, data-custom="value"), доступные через JavaScript.

dir: Определяет направление текста (ltr — слева направо, rtl — справа налево).

draggable: Указывает, можно ли перетаскивать элемент (true/false).

hidden: Прячет элемент (аналогично display: none в CSS).

id: Уникальный идентификатор элемента для ссылок и скриптов.

lang: Определяет язык содержимого элемента.

spellcheck: Включает/отключает проверку орфографии (true/false).

style: Встраивает встроенные CSS-стили.

tabindex: Определяет порядок фокусировки при навигации клавишей Tab (0 — по умолчанию, отрицательные — исключают).

title: Отображает всплывающую подсказку при наведении курсора.

translate: Описывает, переводить ли содержимое при локализации (yes/no).