1. **Определение DOM**: DOM (Document Object Model) — это программный интерфейс для HTML и XML документов, который представляет структуру документа в виде дерева объектов. Каждый элемент, атрибут и текст в документе являются узлами этого дерева, что позволяет программно изменять содержимое, структуру и стиль документа.
2. **Способы доступа к элементам HTML документа**:
   * getElementById(id): доступ к элементу по его уникальному идентификатору.
   * getElementsByClassName(className): доступ к элементам по имени класса (возвращает коллекцию).
   * getElementsByTagName(tagName): доступ к элементам по имени тега (возвращает коллекцию).
   * querySelector(selector): доступ к первому элементу, соответствующему CSS-селектору.
   * querySelectorAll(selector): доступ ко всем элементам, соответствующим CSS-селектору (возвращает коллекцию).
   * parentNode: доступ к родительскому элементу.
   * children: доступ к дочерним элементам.
3. **Свойство childNodes**: Это свойство возвращает коллекцию всех дочерних узлов элемента, включая элементы, текстовые узлы и комментарии. Оно позволяет получить доступ ко всем узлам, находящимся внутри данного элемента.
4. **Событие**: Событие — это действие или изменение состояния, которое происходит в документе, например, нажатие кнопки, перемещение мыши, загрузка страницы и т.д. События могут быть вызваны пользователем или происходить автоматически.
5. **Метод addEventListener()**: Этот метод используется для добавления обработчика события к элементу. Он позволяет указать, какое событие нужно отслеживать и какую функцию выполнять при его возникновении.
6. **Отличия addEventListener() от атрибута обработчика события**:
   * addEventListener() позволяет добавлять несколько обработчиков для одного события, тогда как атрибут обработчика события (например, onclick) позволяет установить только один обработчик.
   * addEventListener() поддерживает возможность указания фазы захвата и всплытия, а атрибуты обработчиков событий этого не делают.
   * addEventListener() не перезаписывает существующие обработчики, в отличие от атрибутов.
7. **Forms**: Формы (forms) в HTML — это элементы, которые позволяют пользователям вводить данные и отправлять их на сервер. Формы могут содержать различные элементы управления, такие как текстовые поля, кнопки, флажки и выпадающие списки.
8. **Суть elements**: Свойство elements у формы возвращает коллекцию всех элементов управления, содержащихся в форме. Это позволяет легко получать доступ к полям ввода, кнопкам и другим элементам формы.
9. **Свойство innerHTML**: Это свойство позволяет получить или установить HTML-содержимое элемента. Оно используется для динамического изменения содержимого элемента, например, добавления новых элементов или изменения текста.
10. **Доступ к элементам <select>**: Для доступа к элементам <select> можно использовать свойство options, которое возвращает коллекцию всех <option> внутри <select>. Также можно использовать value для получения или установки выбранного значения.
11. **Циклы**: Циклы используются для повторения блока кода несколько раз. Они позволяют выполнять одно и то же действие для каждого элемента в коллекции или до тех пор, пока не будет выполнено определенное условие.
12. **Метод checkValidity()**: Этот метод проверяет, соответствует ли элемент формы заданным условиям валидации. Он возвращает true, если элемент валиден, и false, если нет. Это полезно для проверки правильности введенных данных перед отправкой формы.
13. **События**: Некоторые распространенные события включают:
    * click: событие клика мыши.
    * mouseover: событие наведения курсора на элемент.
    * mouseout: событие ухода курсора с элемента.
    * keydown: событие нажатия клавиши.
    * keyup: событие отпускания клавиши.
    * submit: событие отправки формы.
    * load: событие загрузки страницы или ресурса.
    * change: событие изменения значения элемента (например, в полях ввода или выпадающих списках).