1. Какую программу для создания кликабельных прототипов вы знаете?

2. Что такое HTML?

3. Из чего состоит форматированный текст HTML?

4. Что такое HTML-тег?

5. Парными являются все теги?

6. Какие основные виды списков применяют при создании веб-страницы?

7. Зачем использовать списки в HTML?

8. Можем ли мы писать комментарии в HTML-коде? Как?

9. Почему некоторые символы на веб-странице иногда отображаются некорректно?

10. Можно ли присваивать значения не всем атрибутам тега?

11. Как разместить знак copyright на веб-странице, ведь его нет на клавиатуре?

12. Как создать ссылки на разные фрагменты (разделы) одной и той же веб-страницы?

13. Можем ли мы выравнивать элементы списка в HTML-файле?

14. Адрес какой веб-страницы обычно считается адресом сайта?

15. Зачем мы используем альтернативный текст (атрибут alt тега ) для изображений?

16. Могут ли файлы HTML хорошо работать в каком-нибудь редком или ультрасовременном браузере?

17. Как вы думаете гиперссылка может быть только текстовой?

18. Что из себя представляют атрибуты тега <li> — элемента списка?

19. Для чего нужны таблицы стилей (CSS)?

20. Какие типы нумерации в списках вы знаете?

21. Как задать разные цвета для фрагментов текста веб-страницы?

22. Как указать цвета для границ таблицы?

23. Можем ли мы создать ссылку, которая ведет на другую веб-страницу?

24. Могут ли таблицы стилей помочь выровнять изображение и задать способ позиционирования текста относительно него?

25. Может ли одна гиперссылка вести на разные страницы?

26. Как изменить цвет маркера?

27. Можно ли ввести какие-то ограничения на размер текстовых полей в HTML?

28. Что делает тег <br>?

29. Есть ли другой способ разделить текст без использования <br>?

30. Есть у таблиц стилей какая-либо иерархия правил?

31. Какие браузеры поддерживают HTML 5? Какие новые теги в нем появились и для чего?

32. Где в html-документе рекомендуется размесить <link> для подключения CSS?

33. Объясните три основных способа применения стилей CSS к веб-странице.

34. Что такое DOM (объектная модель документа) и как она связана с CSS?

35. В чем разница между классами и идентификаторами в CSS?

36. Опишите floats и как они работают?

37. Какое по умолчанию значение свойства display у <div>, <p>?

38. Что такое CSS препроцессор? Какие мета-языки вы знаете?

39. Что такое селекторы CSS? Назовите сновные.

40. Где и для чего используются mixin и extend?

41. Приведите примеры работы специфичности?

42. Что такое грид-система в CSS? Как и для чего ее можно создать?

43. Какие свойства flex и их значения вы знаете?

44. Что такое SVG?

45. Каким способом можно создать анимацию в веб?

46. Расскажите о блочной модели CSS.

47. Что такое Bootstrap и как ее подключить?

48. Какие методы для манипуляции с DOM вы используете?

49. В чем разница между null и undefined?

50. Какое значение имеет this?

51. Что делает функция feth?

52. Какие методы для обращения к узлам DOM вам известны?

53. Что такое функциональное программирование и какие особенности JS позволяют говорить о нем как о функциональном языке программирования?

54. Что такое функции высшего порядка (Higher Order Functions)?

55. Что делает функция map?

56. В чем разница между ключевыми словами «var», «let» и «const»?

57. Что такое стрелочные функции (Arrow Functions)? В чем ее отличие от обычной функции? Приведите пример.

58. Что такое классы (Classes)? Приведите пример.

59. Какие вы знаете недостатки JS?

60. Какие способы создания событий для компонента формы вы знаете?

61. Какие у JS есть достоинства по сравнению с другими языками программирования?

62. Что такое объект в JS? Приведите примеры использования.

63. Для чего используется ключевое слово «new»?

64. Может ли быть свойством объекта функция?

65. Функция Array.prototype.map какого порядка?

66. Для чего нужен атрибут TYPE в элементе формы INPUT?

1. Примеры программ для создания кликабельных прототипов включают Figma, Sketch, Adobe XD, InVision, Axure RP и Balsamiq.

2. HTML (HyperText Markup Language) - это язык разметки, используемый для создания структуры и представления содержимого веб-страницы.

3. Форматированный текст HTML состоит из HTML-тегов, атрибутов и текстового контента.

4. HTML-тег - это элемент языка HTML, который используется для обозначения и структурирования различных частей содержимого веб-страницы.

5. Нет, не все теги являются парными. Некоторые теги, такие как `<br>` или `<img>`, являются одиночными и не имеют закрывающего тега.

6. Основные виды списков включают маркированные списки (`<ul>`), нумерованные списки (`<ol>`) и списки определений (`<dl>`).

7. Списки в HTML используются для организации и представления информации в упорядоченной или неупорядоченной форме, что облегчает восприятие и навигацию по контенту веб-страницы.

8. Да, можно писать комментарии в HTML-коде. Комментарии начинаются с `<!--` и заканчиваются на `-->`. Все, что находится между этими символами, будет считаться комментарием и будет проигнорировано браузером.

9. Некорректное отображение символов на веб-странице может быть вызвано различными факторами, такими как неподдерживаемая кодировка символов, отсутствие соответствующего шрифта на компьютере пользователя или ошибки в кодировке и передаче данных.

10. Да, можно присваивать значения только определенным атрибутам тега, а другие атрибуты можно опустить.

11. Знак copyright (©) можно разместить на веб-странице с помощью символической ссылки &copy; или кодового значения &#169;.

12. Чтобы создать ссылку на разные фрагменты одной веб-страницы, нужно присвоить идентификаторы (`id`) нужным элементам и затем создать ссылки, указывая `href` с символом решетки (`#`) и идентификатором элемента.

13. Да, элементы списка в HTML можно выравнивать с помощью CSS-свойства `list-style-position`. Значение `inside` выравнивает маркеры списка внутри содержимого, а значение `outside` помещает

маркеры снаружи содержимого.

14. Адресом сайта обычно считается адрес его домашней страницы, который часто записывается в формате "http://www.example.com".

15. Альтернативный текст (атрибут `alt`) для изображений используется в HTML для предоставления текстовой замены, если изображение не может быть отображено или доступно для пользователя. Он также помогает поисковым системам и людям с ограниченными возможностями понять содержание изображения.

16. HTML-файлы должны хорошо работать в большинстве современных браузеров. Однако редкие или ультрасовременные браузеры могут не поддерживать некоторые новые функции HTML.

17. Гиперссылка может быть не только текстовой, но также может быть представлена изображением или другими элементами, которые можно нажимать или активировать.

18. Элемент `<li>` представляет отдельный пункт списка и может содержать текст или другие элементы.

19. Таблицы стилей (CSS) используются для стилизации и оформления веб-страниц. Они позволяют разделять содержимое и представление, управлять внешним видом элементов и обеспечивать более гибкую и согласованную визуальную обработку веб-сайта.

20. Типы нумерации в списках включают нумерацию арабскими цифрами (`type="1"`), римскими цифрами (`type="i"`, `type="I"`), буквами (`type="a"`, `type="A"`) и другими.

21. Цвета для фрагментов текста веб-страницы можно задавать с помощью CSS-свойства `color` или использовать специальные теги, такие как `<span>` или `<font>` с атрибутом `color`.

22. Цвета для границ таблицы можно задавать с помощью CSS-свойства `border-color` или использовать специальные классы стилей.

23. Да, можно создать ссылку, которая ведет на другую веб-страницу, указав путь к файлу в атрибуте `href` элемента `<a>`.

24. Таблицы стилей могут помочь выровнять изображение и задать способ позиционирования текста относительно него с помощью свойств `float`, `position` и других CSS-свойств.

25. Да, одна гиперссылка может вести на разные страницы. Для этого в атрибуте `href` нужно указать разные адреса страниц.

26. Ц

вет маркера списка можно изменить с помощью CSS-свойства `list-style-color`.

27. Да, размер текстовых полей в HTML можно ограничивать с помощью атрибута `maxlength`, указывающего максимальное количество символов, которые можно ввести.

28. Тег `<br>` используется для создания перевода строки внутри текста или между элементами.

29. Другим способом разделить текст без использования `<br>` является использование блочных элементов, таких как `<p>` или создание отдельных блоков с помощью CSS.

30. В таблицах стилей (CSS) существует иерархия правил, где более специфичные (точные) правила имеют приоритет над менее специфичными. Это позволяет контролировать, какие стили будут применяться к конкретным элементам.

31. Большинство современных браузеров поддерживают HTML 5. Некоторые из новых тегов HTML 5 включают `<header>`, `<footer>`, `<nav>`, `<video>`, `<canvas>` и другие, предназначенные для более семантической разметки и поддержки новых функций, таких как видео и графика.

32. Рекомендуется размещать элемент `<link>` для подключения CSS в секции `<head>` HTML-документа.

33. Три основных способа применения стилей CSS к веб-странице: внешние таблицы стилей (внешний CSS-файл), встроенные стили (с помощью элемента `<style>` внутри HTML-документа) и встроенные стили непосредственно в теги HTML (с помощью атрибута `style`).

34. DOM (Document Object Model) - это представление HTML-документа в виде дерева объектов, которое позволяет JavaScript взаимодействовать с элементами и изменять содержимое и структуру документа. DOM связан с CSS, так как CSS использует селекторы для выбора элементов DOM и применения стилей к ним.

35. В CSS классы и идентификаторы используются для выбора и стилизации элементов. Основная разница состоит в том, что идентификатор (атрибут `id`) должен быть уникальным для каждого элемента на странице, тогда как класс (атрибут `class`) может быть применен к нескольким элементам.

36. Floats в CSS используются для обтекания элементов другими элементами. Они изменяют позиционирование элемента вдоль горизонтальной оси

. Float-свойство может использоваться для создания макетов с несколькими колонками или выравнивания элементов.

37. У тегов `<div>` и `<p>` значение свойства display по умолчанию равно "block".

38. CSS-препроцессоры - это инструменты, которые расширяют возможности CSS и предоставляют дополнительные функции, такие как переменные, миксины и вложенные стили. Некоторые популярные CSS-препроцессоры: Sass, Less и Stylus.

39. Селекторы CSS - это паттерны, которые используются для выбора определенных элементов на веб-странице. Некоторые основные селекторы включают элементные селекторы (`element`), классовые селекторы (`.class`), идентификаторные селекторы (`#id`), псевдоклассы (`:hover`, `:nth-child()`) и псевдоэлементы (`::before`, `::after`).

40. Mixin и extend - это функциональности некоторых CSS-препроцессоров. Mixin позволяет группировать наборы стилей для повторного использования, а extend позволяет наследовать стили одного селектора от другого.

41. Специфичность в CSS определяет, какой стиль будет применяться, когда есть несколько правил, которые применяются к одному элементу. Она основана на комбинации селекторов и их весе. Примеры работы специфичности: если есть два правила с одинаковым селектором, то важнее будет то, которое идет позже в коде или имеет большую специфичность.

42. Грид-система в CSS представляет собой способ организации элементов в виде сетки. С помощью грид-системы можно создавать регулярные сетки с фиксированными или адаптивными размерами ячеек для легкого расположения и выравнивания содержимого.

43. Некоторые свойства flex в CSS включают `flex-direction`, `flex-wrap`, `flex-grow`, `flex-shrink`, `flex-basis`, `align-items` и `justify-content`. Они используются для создания гибких и адаптивных макетов с использованием Flexbox.

44. SVG (Scalable Vector Graphics) - это формат векторной графики, который позволяет создавать и отображать графические элементы и иллюстрации в веб-страницах с использованием XML-синтаксиса.

45. Анимацию

в веб можно создать с помощью CSS анимаций, JavaScript или с использованием специальных библиотек и фреймворков, таких как CSS-анимации, jQuery или библиотеки для анимации, такие как GreenSock Animation Platform (GSAP).

46. Блочная модель CSS описывает способ форматирования и расположения элементов веб-страницы. Она состоит из содержимого элемента, отступов (padding), границ (border) и полей (margin) вокруг элемента.

47. Bootstrap - это популярный фреймворк для разработки веб-сайтов, который предоставляет готовые компоненты и стили, чтобы упростить процесс создания адаптивных и кросс-браузерных веб-интерфейсов. Он подключается путем добавления ссылок на соответствующие файлы CSS и JavaScript в HTML-документ.

48. Для манипуляции с DOM в JavaScript используются методы и свойства объектов, таких как `getElementById()`, `querySelector()`, `appendChild()`, `removeChild()` и другие.

49. `null` - это значение, которое указывает на отсутствие значения или пустоту. `undefined` - это значение, которое указывает, что переменная была объявлена, но ей не было присвоено значение.

50. Значение `this` в JavaScript зависит от контекста выполнения функции. В глобальной области видимости `this` ссылается на глобальный объект (например, `window` в браузере), в методе объекта `this` ссылается на сам объект, а при вызове функции с использованием `new` `this` ссылается на новый экземпляр созданного объекта.

51. Функция `fetch` в JavaScript используется для отправки HTTP-запросов и получения ответов от сервера. Она возвращает промис, который позволяет обрабатывать полученные данные.

52. Некоторые методы для обращения к узлам DOM включают `getElementById()`, `querySelector()`, `getElementsByTagName()`, `getElementsByClassName()`, `parentNode`, `childNodes` и другие.

53. Функциональное программирование - это парадигма программирования, которая ориентирована на использование функций в качестве основного строительного блока программы. JavaScript обладает особенностями, такими как возможность передачи функций в качестве аргументов, использование функций высшего порядка и замыкания, что делает его функциональным языком программирования.

54. Функции высшего порядка (Higher Order Functions) - это функции, которые могут принимать другие функции в качестве аргументов и/и

ли возвращать функции в качестве результата. Они позволяют абстрагировать поведение и создавать более гибкие и модульные программы.

55. Функция `map` в JavaScript используется для преобразования каждого элемента массива с помощью переданной функции и возвращает новый массив с результатами преобразования.

56. `var`, `let` и `const` - это ключевые слова для объявления переменных в JavaScript. `var` имеет функциональную область видимости, `let` имеет блочную область видимости и может быть переназначена, а `const` также имеет блочную область видимости, но не может быть переназначена после инициализации.

57. Стрелочные функции (Arrow Functions) - это новый синтаксис для объявления функций в JavaScript. Они имеют более краткую запись и особое поведение относительно контекста `this`. Например: `const sum = (a, b) => a + b;`.

58. Классы (Classes) в JavaScript позволяют создавать объекты на основе шаблона, который определяет свойства и методы объекта. Они предоставляют синтаксический сахар для более удобного и понятного создания и наследования объектов. Пример:

```

class Person {

constructor(name) {

this.name = name;

}

sayHello() {

console.log(`Hello, my name is ${this.name}.`);

}

}

const person = new Person("John");

person.sayHello(); // Output: Hello, my name is John.

```

59. Некоторые недостатки JavaScript включают слабую типизацию, проблемы с областью видимости переменных (`var`), глобальное пространство имен, неявное преобразование типов данных, проблемы безопасности (например, XSS) и некоторые особенности в работе с асинхронным кодом.

60. Некоторые способы создания событий для компонента формы включают использование JavaScript для прослушивания событий, таких как `onclick`, `onsubmit` и `onchange`, или использование библиотеки, такой как jQuery, для упрощения работы с событиями.

61. Достоинства JavaScript включают его широкую поддержку веб-браузерами, возможность создания динамических и интерактивных веб-страниц, широкое сообщество разработчиков, обширный выбор библиотек и фреймворков, а также возможность использования на сервере (Node.js) для разработки полноценных веб-приложений.

62. Объект в JavaScript - это сущ

ность, которая содержит свойства и методы. Он может быть создан с помощью литерала объекта или с помощью конструктора объекта. Примеры:

```

// Литерал объекта

const person = {

name: "John",

age: 30,

sayHello: function() {

console.log(`Hello, my name is ${this.name}.`);

}

};

person.sayHello(); // Output: Hello, my name is John.

// Конструктор объекта

function Person(name, age) {

this.name = name;

this.age = age;

this.sayHello = function() {

console.log(`Hello, my name is ${this.name}.`);

}

}

const person = new Person("John", 30);

person.sayHello(); // Output: Hello, my name is John.

```

63. Ключевое слово "new" используется для создания нового экземпляра объекта на основе конструктора объекта.

64. Да, функция может быть свойством объекта в JavaScript.

65. Функция `Array.prototype.map` является функцией первого порядка (first-order function).

66. Атрибут `type` в элементе формы `input` используется для указания типа вводимых данных, например, `type="text"` для текстового поля, `type="number"` для числового поля, `type="email"` для поля электронной почты и т. д.