**1. Что такое CSS? Какие задачи оно решает?**

CSS (Cascading Style Sheets, "каскадные таблицы стилей") — это язык, который используется для описания внешнего вида документа HTML. CSS отвечает за стилизацию элементов, их расположение на странице, цвета, шрифты и взаимодействие с пользователем.

Задачи, которые решает CSS:

Управление внешним видом — стилизация текста, фонов, границ и прочего.

Расположение элементов — как в потоке документа, так и в сложных макетах (например, с использованием flexbox или grid).

Адаптивность для различных устройств и экранов.

Добавление анимаций и эфектов взаимодействия (hover, focus).

**2. Что такое каскад в CSS? Как он работает?**

Каскад в CSS — это механизм, который определяет, как применяются стили, если они конфликтуют. Основной принцип каскада — определить, какие правила имеют больше приоритета. При расчете используются три фактора:

Специфичность (specificity): чем точнее селектор, тем выше приоритет.

Порядок в коде: стили, определенные позже, переопределяют предыдущие, если их специфичность одинакова.

Важность правила: !important придает правилу наивысший приоритет.

**3. Чем id отличается от class? Какой использовать в работе с CSS?**

id\*\*: указывает уникальный идентификатор элемента. На странице может быть только один элемент с этим id.

class\*\*: задает наименование класса, который может применяться к нескольким элементам.

В CSS:

id имеет более высокую специфичность, чем class.

Использовать id для стилизации не рекомендуется, так как это усложняет переопределение стилей (из-за специфичности). Предпочтительнее использовать классы.

**4. Что такое "блочные" и "строчные" элементы? Как это связано с CSS?**

Блочные элементы\*\* (block): занимают всю ширину родителя (100%), начиная с новой строки.

Строчные элементы\*\* (inline): занимают только пространство, соответствующее их содержимому, и не начинают новую строку.

**5. Объясните CSS-специфичность.**

Специфичность — это метод, с помощью которого определяется, какой селектор имеет больший приоритет. Она рассчитывается по таблице "весов":

Стили из атрибута style на элементе = 1000.

Селектор id = 100.

Селектор класса/псевдокласса (.class или :hover) = 10.

Селектор элемента (div, h1) = 1.

**6. Что делает свойство z-index?**

Свойство z-index определяет порядок наложения (слоев) элементов. Элементы с более высоким значением z-index отображаются поверх элементов с меньшим значением. Работает только с элементами, у которых задано значение position (relative, absolute, fixed).

**7. Какие доступны позиционирования элементов в CSS?**

Значения свойства position:

static (по умолчанию): элемент находится в обычном потоке документа.

relative: элемент смещается относительно самого себя.

absolute: элемент позиционируется относительно ближайшего предка с position: relative, если такой имеется. Если нет, то относительно ``.

fixed: элемент фиксируется на экране, не перемещается при скролле.

sticky: гибрид relative и fixed; элемент "приклеивается" при определенных условиях.

**8. Что такое Flexbox и когда его использовать?**

Flexbox — это CSS-модуль для создания удобного управления позиционированием и выравниванием элементов в одном измерении (по строке или по столбцу). Flexbox удобен для создания адаптивных макетов.

**9. Что такое Grid и как он отличается от Flexbox?**

CSS Grid — это модуль для работы с двухмерными макетами (строки и столбцы). Если Flexbox удобен для работы с линейными макетами, Grid позволяет описывать сложные сетки.

**10. Какие методы центровки есть в CSS?**

Основные способы центровки:

Текст и блочные элементы: text-align: center;

Flexbox:

      display: flex;

   justify-content: center; /\* по горизонтали \*/

   align-items: center; /\* по вертикали \*/

Grid:

      display: grid;

   place-items: center;

Абсолютное позиционирование:

      position: absolute;

   left: 50%;

   top: 50%;

   transform: translate(-50%, -50%);

**11. Как сделать адаптивный дизайн?**

Основные методы:

Использование медиазапросов (@media).

Пропорциональные единицы измерения (например, %, em, rem).

Сетки (Flexbox, Grid).

Универсальные свойства, например, max-width.

**12. Что такое псевдоэлементы?**

Псевдоэлементы позволяют стилизовать части элемента. Основные:

::before — вставляет контент перед элементом.

::after — вставляет контент после элемента.

::Placeholder, ::Selection, ::First-line, ::First-letter

**13. В чем разница между absolute, relative, и fixed позиционированием?**

Подробный ответ дан в вопросе №7.

**14. Как работает свойство opacity?**

opacity задает уровень прозрачности элемента (значение от 0 до 1). При значении opacity: 0; элемент полностью прозрачный.

**15. Как сделать элемент квадратным, независимо от его ширины?**

Ответ:

Используйте padding с привязкой к ширине элемента:

.element {

  width: 50%;

  padding-top: 50%; /\* Высота равна ширине \*/

**16. Что такое псевдоклассы? Чем они отличаются от псевдоэлементов?**

Псевдоклассы используются для стилизации элементов в определенных состояниях или на основе их положения в DOM-структуре. Начинаются с : (например, :hover, :nth-child(), checked, disabled, valid, invalid, visited).

Псевдоэлементы используются для стилизации частей самого элемента и всегда начинаются с :: (например, ::before, ::after).

**17. Какие единицы измерения существуют в CSS? Назовите их особенности.**

CSS использует абсолютные и относительные единицы измерения:

Абсолютные:\*\*

  px — пиксели. Точный размер, не зависит от устройства.

  cm, mm, in — сантиметры, миллиметры, дюймы. Практически не используются.

  pt — пункты (1 pt = 1/72 дюйма).

Относительные:\*\*

  % — процент относительно родительского элемента.

  em — размер шрифта относительно шрифта родителя.

  rem — размер шрифта относительно корневого (html { font-size }).

  vw, vh — проценты от ширины и высоты окна браузера.

  vmin, vmax — минимальное или максимальное значение между vw и vh.

**18. Какие свойства отвечают за отступы и размеры элемента?**

В CSS отступы и размеры состоят из четырех основных составляющих (блочной модели):

Размеры элемента:

   width, height — размер контента элемента.

   max-width, max-height и min-width, min-height ограничивают размеры.

Внутренние отступы (padding):

   padding — отступы от контента до границы элемента.

Внешние отступы (margin):

   margin — расстояние между элементами.

Границы (border):

   border: width style color;, например:

          border: 1px solid black;

**19. Что такое "Блочная модель" (Box Model)?**

В CSS каждый элемент представлен коробкой (box), состоящей из 4 основных частей:

Content — сам контент элемента (текст, изображение и т.д.).

Padding — внутренний отступ (расстояние между контентом и границей).

Border — рамка элемента.

Margin — внешний отступ от соседних элементов.

Итоговая ширина элемента = ширина контента + padding + border + margin.

**20. Что делает свойство box-sizing? Для чего используется?**

Свойство box-sizing изменяет расчет размеров элементов:

content-box (по умолчанию): размеры задаются для \*\*контента, а padding и border добавляются сверху.

border-box\*\*: размеры задаются для всего элемента, включая контент, паддинг и границу.

**21. Как ограничить длину текста в CSS?**

Для ограничения длины текста используйте следующие свойства:

Обрезка строки с многоточием:

      .text {

     width: 200px;

     white-space: nowrap; /\* Не переносить текст \*/

     overflow: hidden; /\* Обрезать выходящий текст \*/

     text-overflow: ellipsis; /\* Добавить "..." \*/

   }

  Перенос слов:

      .text {

     word-wrap: break-word; /\* Перенос слов при необходимости \*/

   }

**22. Для чего служит свойство overflow?**

Свойство overflow управляет тем, что происходит с содержимым, выходящим за пределы блока.

visible (по умолчанию):\*\* содержимое выходит за пределы.

hidden:\*\* обрезает содержимое.

scroll:\*\* добавляет полосы прокрутки.

auto:\*\* добавляет полосы прокрутки, если содержимое больше контейнера.

**23. В чем разница между inline, inline-block, и block?**

inline:\*\* элемент находится в одной строке с другими элементами. Нельзя задавать ширину и высоту.

block:\*\* элемент занимает всю ширину строки. Можно задать размеры.

inline-block:\*\* как inline, но можно задавать размеры (ширину и высоту).

**24. Что такое медиазапросы?**

Медиазапросы (@media) позволяют применять стили в зависимости от характеристик устройства: ширины экрана, ориентации, разрешения.

**25. Что делает свойство float?**

float используется для "обтекания" элемента текстом или другими элементами. Значения:

left — элемент плавает слева.

right — элемент плавает справа.

none — элемент не плавает.

**26. Что делает свойство clear?**

clear используется для прекращения обтекания (float):

left — сбрасывает обтекание слева.

right — сбрасывает обтекание справа.

both — сбрасывает обтекание с обеих сторон.

**27. Какие свойства используются для анимаций в CSS?**

Для создания анимаций в CSS используются:

transition: анимация изменения свойств.

animation: сложные анимации с использованием ключевых кадров (@keyframes).

Пример @keyframes:

@keyframes slide {

  0% { transform: translateX(0); }

  100% { transform: translateX(100px); }

}

div {

  animation: slide 1s ease-in-out;

}

**28. Что такое clip-path и как его использовать?**

clip-path используется для "обрезки" формы элемента. Пример:

div {

  clip-path: circle(50% at center);

}

**29. Как работает свойство position: sticky?**

sticky объединяет поведение relative и fixed. Элемент ведет себя как relative, пока не пересечет заданное значение (например, верхний край). После этого становится фиксированным.

**30. Как сделать фиксированный фон на странице?**

Используйте свойство background-attachment:

body {

  background-image: url('image.jpg');

  background-attachment: fixed; /\* Фон фиксирован \*/

  background-size: cover;

}

**31. Чем отличаются относительные единицы измерения em и rem?**

em: зависит от размера шрифта \*\*родительского элемента и используется для создания относительного масштабирования. Например, 1em равен размеру шрифта родителя, 2em — удвоенному размеру.

rem: зависит от размера шрифта \*\*корневого элемента (html).

Главное отличие: em зависит от ближайшего родителя, а rem всегда от корневого элемента.

**32. Для чего служит свойство visibility и чем оно отличается от display?**

visibility\*\* управляет видимостью элемента.

  Значения: visible (по умолчанию) или hidden.

  Элемент, скрытый через visibility: hidden, остается на своем месте в потоке документа, но он невидим.

display\*\* управляет отображением элемента в потоке:

  Значение display: none удаляет элемент из потока (как будто его нет), в отличие от visibility.

**33. Какие эффекты дает свойство transition? Как его использовать?**

Свойство transition добавляет анимацию к изменениям CSS-свойств.

**34. Что такое CSS-переменные и как они используются?**

CSS-переменные (custom properties) начинаются с -- и задаются в стилях. Они позволяют управлять повторяющимися значениями.

**35. Какие преимущества дает использование свойства inherit?**

Свойство inherit заставляет элемент наследовать значение от родителя. Это полезно для явного указания наследования у свойств, которые по умолчанию не наследуются, например, border или background.

**36. В чем разница между relative, absolute, и sticky позиционированием?**

relative:\*\* элемент остаётся в потоке, но может быть смещён относительно своего положения.

absolute:\*\* элемент удаляется из потока и позиционируется относительно ближайшего предка с relative или absolute. Если таких предков нет, позиционирование будет происходить относительно ``.

sticky:\*\* гибрид. Элемент ведет себя как relative, пока не достигнет заданного значения, а затем фиксируется (прилипает).

**37. Что такое псевдокласс nth-child()? Как он работает?**

nth-child() позволяет выбрать элемент на основе его позиции среди братьев. Позиция задаётся с помощью формулы:

nth-child(1) — первый элемент.

nth-child(2n) — четные элементы.

nth-child(2n+1) — нечетные элементы.

**38. Как работают свойства min-width и max-width?**

min-width:\*\* задаёт минимальную ширину элемента. Элемент не становится уже этого значения.

max-width:\*\* задаёт максимальную ширину элемента. Элемент не становится шире этого значения.

Если контейнер меньше 300px, то элемент остается шириной 300px.

**39. Что делает свойство object-fit?**

object-fit управляет тем, как контент (например, изображение или видео) вписывается в контейнер.

Примеры:

contain: пропорционально вписывает контент.

cover: заполняет контейнер, обрезая лишнее.

none: контент не изменяет размера.

**40. Что делает свойство overflow-x и overflow-y?**

Эти свойства ограничивают прокрутку содержимого по горизонтали (overflow-x) и вертикали (overflow-y).

Значения:

hidden: обрезает содержимое.

scroll: добавляет полосы прокрутки.

auto: добавляет прокрутку, если содержимое больше контейнера.

**41. Какие уровни видимости задаются через свойства z-index?**

Элемент с большим z-index отображается поверх элементов с меньшим значением.

Отрицательные значения допускаются, что приводит к "снижению" уровня элемента.

**42. Что такое hover, focus, и active? Как они работают?**

Это псевдоклассы, используемые для стилизации состояния элемента:

hover:\*\* применяется, когда курсор мыши находится над элементом.

focus:\*\* применяется, когда элемент находится в фокусе (например, ``).

active:\*\* применяется, когда элемент активен (например, при нажатии на кнопку).

**43. Что делает свойство white-space?**

white-space управляет тем, как пробелы и переносы строк обрабатываются в тексте.

normal (по умолчанию): пробелы схлопываются, строки переносятся.

nowrap: пробелы схлопываются, строки не переносятся.

pre: сохраняются пробелы и переносы.

**44. Как работает свойство cursor?**

cursor изменяет вид указателя мыши. Примеры:

pointer: рука (используется для ссылок).

not-allowed: запрет (перечеркнутый круг).

text: указатель для текста (например, на ``).

**45. Какие виды градиентов поддерживает CSS?**

CSS поддерживает:

Линейные градиенты (linear-gradient)\*\*:

    background: linear-gradient(to right, red, blue);

 Радиальные градиенты (radial-gradient)\*\*:

    background: radial-gradient(circle, red, yellow);

 Конусные градиенты (conic-gradient)\*\*:

    background: conic-gradient(from 0deg, red, yellow, blue);

**46. Что такое свойство background-size, и какие значения оно принимает?**

Свойство background-size определяет размер фонового изображения. Возможные значения:

auto\*\* (по умолчанию): изображение отображается в оригинальном размере.

cover\*\*: изображение растягивается, чтобы полностью покрыть контейнер, сохраняя пропорции. Лишние части изображения обрезаются.

contain\*\*: изображение растягивается, чтобы полностью поместиться в контейнер, сохраняя пропорции.

Пользовательские размеры\*\*: например, 100px 200px или 50% 50%.

**47. Что такое viewport, и как связаны с ним единицы vw и vh?**

viewport\*\* — это область браузера, в которой отображается содержимое. Это видимая часть страницы.

vw\*\* (Viewport Width): 1 единица = 1% ширины окна.

vh\*\* (Viewport Height): 1 единица = 1% высоты окна.

**48. Какие существуют способы стилизации шрифтов в CSS?**

font-family — задает семейство шрифтов:

font-size — изменяет размер шрифта (например, 16px, 1.2em, 120%).

font-weight — задает насыщенность шрифта (например, normal, bold, или от 100 до 900).

font-style — стиль шрифта (например, normal, italic, oblique).

line-height — высота строки.

text-transform — преобразует текст (например, uppercase, lowercase, capitalize).

Добавление пользовательских шрифтов через @font-face:

      @font-face {

     font-family: 'MyFont';

     src: url('myfont.woff2') format('woff2');

   }

**49. Чем отличается inline и inline-block, и когда использовать каждый из них?**

inline:\*\* элементы находятся в одной строке, нельзя задавать ширину и высоту.

inline-block:\*\* элементы находятся в одной строке (как inline), но можно задавать высоту, ширину и паддинги.

**50. Что делает свойство filter, и какие фильтры можно применить?**

Свойство filter позволяет применить графические эффекты к элементам, например, размытость или изменения яркости.

Примеры фильтров:

blur(5px) — размытие.

brightness(150%) — увеличение яркости.

contrast(200%) — увеличение контрастности.

grayscale(100%) — черно-белое изображение.

opacity(50%) — прозрачность.

**51. Как работает transition-delay, и как его применить?**

Свойство transition-delay задает задержку перед началом анимации. Значение указывается в секундах (s) или миллисекундах (ms).

**52. Какие значения свойства position вы знаете, и как они работают?**

static (по умолчанию): элемент находится в нормальном потоке документа.

relative: элемент остается в потоке документа, но может быть смещен относительно своего начального положения.

absolute: элемент позиционируется относительно ближайшего предка с relative, absolute или fixed.

fixed: элемент фиксируется относительно окна браузера.

sticky: прилипает к заданной позиции при прокрутке.

53. Что такое z-index, и как оно работает?

z-index определяет порядок наложения элементов (что поверх чего). Чем выше значение, тем выше элемент.

54. Какие существуют варианты задания границ в CSS?

Свойство border определяет границы элемента. Оно состоит из:

Толщина: например, 1px.

Стиль: например, solid, dashed, dotted, none.

Цвет: например, black.

Также можно задавать:

Скругление углов: border-radius.

Стили для каждой стороны: border-top, border-left и т.д.

**55. Что делает свойство outline, и чем оно отличается от border?**

outline\* — отображает рамку \*вне\*\* элемента, не влияет на размеры элемента, не отнимает места.

border\*\* — рамка внутри размеров элемента, влияет на его размеры.

**56. Что такое overflow, и какие значения принимает?**

Свойство overflow управляет тем, что происходит с содержимым, выходящим за пределы размера элемента.

Значения:

visible: содержимое выходит за пределы (по умолчанию).

hidden: содержимое обрезается.

scroll: добавляются полосы прокрутки.

auto: полосы прокрутки появляются, если содержимое превышает размеры.

**57. Какая разница между align-items и align-content в Flexbox?**

align-items: выравнивает элементы \*\*по основной оси внутри Flex-контейнера (если ось — вертикальная).

align-content\*\*: управляет расположением целых строк в многострочном Flex-контейнере.

**58. Как работает свойство clip-path?**

clip-path позволяет обрезать содержимое элемента с помощью заданной фигуры.

**59. Что делать, если шрифт выглядит некрасиво в браузере? Как улучшить?**

Для улучшения отображения шрифтов используйте свойства:

text-rendering\*\*: например, optimizeLegibility для лучших шрифтов.

-webkit-font-smoothing\*\*: сглаживание шрифтов (для Webkit).

font-weight\*\*: регулировать толщину.

**60. Как создать CSS-анимацию с использованием @keyframes?**

@keyframes задаёт ключевые кадры анимации.

2**. Как подключить CSS к веб-странице**

CSS можно подключить к веб-странице тремя способами:

- Внешний CSS: Используя тег `<link>` в разделе `<head>` HTML-документа.

- Внутренний CSS: Определяя стили в теге `<style>` внутри `<head>`.

-Инлайн CSS: Применяя стиль непосредственно к элементу с помощью атрибута `style`.

**3. Объясните модель коробки (box model) в CSS**

Модель коробки описывает, как элементы отображаются на веб-странице. Она состоит из следующих частей:

- Содержимое (content): Самый внутренний элемент, где размещается текст и изображения.

- Отступы (padding): Пространство между содержимым и границей элемента.

- Границы (border): Линия вокруг элемента, которая может иметь стиль, цвет и толщину.

- Внешние отступы (margin): Пространство между элементами.

4**. Что такое селекторы в CSS?**

Селекторы — это шаблоны, которые определяют, к каким HTML-элементам применяются стили. Существует несколько типов селекторов:

- Теговые селекторы: выбирают элементы по имени тега.

- Классовые селекторы: выбирают элементы по классу, используя точку перед именем класса (например, `.classname`).

- Идентификаторы: выбирают элементы по ID, используя решетку перед именем ID (например, `#idname`).

5. Каковы различия между `display: inline`, `display: block` и `display: inline-block`?

- inline: Элементы не начинают новую строку и занимают только необходимую ширину.

- block: Элементы начинают новую строку и занимают всю доступную ширину.

- inline-block\*\*: Элементы не начинают новую строку, но могут иметь ширину и высоту.

Промежуточные вопросы

**6. Что такое медиа-запросы в CSS?**

Медиа-запросы позволяют применять разные стили в зависимости от характеристик устройства, таких как ширина экрана или ориентация. Это ключевой элемент адаптивного дизайна.

**7. Объясните свойство `z-index`**

Свойство `z-index` определяет порядок наложения элементов на странице. Чем выше значение `z-index`, тем выше элемент будет расположен относительно других элементов с меньшими значениями.

**8. Что такое Flexbox?**

Flexbox — это метод компоновки в CSS, который позволяет легко выравнивать элементы внутри контейнера. Он обеспечивает гибкость в распределении пространства между элементами и их выравнивании.

.container {

display: flex;

justify-content: space-between;

}

**9. Каковы преимущества использования CSS-препроцессоров?**

CSS-препроцессоры, такие как Sass или Less, добавляют функциональность к стандартному CSS:

- Поддержка переменных для хранения значений.

- Миксины для повторного использования стилей.

- Вложенные правила для более удобной структуры кода.

**10. Объясните концепцию специфичности в CSS**

Специфичность определяет, какие стили применяются к элементам в случае конфликта правил. Она рассчитывается на основе типа селектора:

- Инлайн стили имеют наивысшую специфичность.

- ID-селекторы имеют более высокую специфичность, чем классы.

- Классы имеют более высокую специфичность, чем теговые селекторы.

### Продвинутые вопросы

**11. Что такое CSS Grid?**

CSS Grid — это мощная система компоновки для создания сложных макетов с использованием строк и столбцов. Она позволяет легко управлять пространством между элементами.

**12. Как работает свойство `box-sizing`?**

Свойство `box-sizing` определяет, как рассчитываются ширина и высота элемента:

- content-box (по умолчанию): ширина/высота не включает отступы и границы.

- border-box: ширина/высота включает отступы и границы.

**13. Что такое псевдоклассы в CSS?**

Псевдоклассы — это специальные состояния элементов, которые позволяют применять стили к элементам в зависимости от их состояния или положения:

- `:hover`: применяется при наведении мыши на элемент.

- `:focus`: применяется при получении фокуса элементом.

**14. Каковы основные свойства для работы с фоном в CSS?**

Основные свойства для работы с фоном включают:

- `background-color`: задает цвет фона.

- `background-image`: задает изображение фона.

- `background-size`: определяет размер фонового изображения.

**15. Как создать анимацию с помощью CSS?**

Анимации создаются с использованием ключевых кадров (`@keyframes`) и свойства `animation`. Например:

@keyframes example {

from {background-color: red;}

to {background-color: yellow;}

}

.box {

animation-name: example;

animation-duration: 4s;

}

### Заключительные вопросы

**16. Как скрыть элемент с помощью CSS?**

Элементы можно скрыть с помощью следующих свойств:

- `display: none;`: полностью удаляет элемент из потока документа.

- `visibility: hidden;`: скрывает элемент, но он все еще занимает место в потоке документа.

**17. Что такое адаптивный дизайн?**

Адаптивный дизайн — это подход к веб-дизайну, который обеспечивает оптимальное отображение сайта на различных устройствах путем использования медиа-запросов и гибких сеток.

**18. Объясните разницу между классами и идентификаторами в CSS**

Классы (.) могут быть применены к нескольким элементам, тогда как идентификаторы (#) должны быть уникальными для каждого элемента на странице.

**19. Как использовать шрифты Google в проекте?**

Шрифты Google можно подключить через `<link>` в разделе `<head>` HTML-документа или импортировав их через CSS:

@import url('https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap');

**20. Что такое кастомные свойства (переменные) в CSS?**

Кастомные свойства позволяют создавать переменные для хранения значений стилей:

--main-color: blue;

.element {

color: var(--main-color);

}

### Дополнительные вопросы

**21. Как использовать псевдоэлемент `::before`?**

Псевдоэлемент `::before` позволяет добавлять контент перед содержимым элемента:

.element::before {

content: "Prefix ";

}

**22. Как задать фиксированное положение элемента?**

Для задания фиксированного положения элемента используется свойство `position` со значением `fixed`:

.fixed-element {

position: fixed;

top: 0;

right: 0;

}

**23. Что такое "cascading" в CSS?**

"Cascading" означает порядок применения стилей на основе их специфичности и источника (инлайн стили имеют высший приоритет над внешними стилями).

**24. Каковы основные принципы семантического HTML/CSS?**

Семантический HTML/CSS подразумевает использование правильных тегов для обозначения структуры документа и применение стилей таким образом, чтобы они были понятны как пользователям, так и поисковым системам.

**25. Как создать градиентный фон с помощью CSS?**

Градиентный фон создается с использованием свойства `background-image`:

background-image: linear-gradient(to right, red, yellow);

**26. Что такое "overflow" в CSS?**

Свойство `overflow` управляет тем, что происходит с содержимым блока при переполнении его области видимости:

- `visible`: содержимое выходит за пределы блока.

- `hidden`: содержимое обрезается.

- `scroll`: добавляются полосы прокрутки.

**28. Объясните концепцию "mobile-first" дизайна.**

"Mobile-first" дизайн подразумевает создание веб-сайтов с учетом мобильных устройств сначала, а затем адаптацию для более крупных экранов через медиа-запросы.

**29. Как создать сетку с использованием Flexbox?**

Flexbox позволяет легко создавать сетки путем установки свойства `display` на значение `flex` у родительского контейнера:

.container {

display: flex;

flex-wrap: wrap;

}

**30. Что такое "transition" в CSS?**

Свойство `transition` позволяет плавно изменять значения свойств элементов при изменении состояния:

## Дополнительные вопросы и ответы по CSS для собеседования на позицию джуниор фронтенд разработчика

**\*\*\*\*1. Что такое CSS Reset?\*\*\*\***

CSS Reset — это набор стилей, который сбрасывает стандартные стили браузеров, чтобы обеспечить единообразный вид элементов на разных платформах. Это помогает избежать неожиданных стилей по умолчанию.

**\*\*\*\*2. Как работает свойство `float`?\*\*\*\***

Свойство `float` позволяет элементам "плавать" влево или вправо, позволяя тексту и другим элементам обтекать их. Элементы, которые плавают, выходят из нормального потока документа.

**\*\*\*\*3. Что такое `clear` и как его использовать?\*\*\*\***

Свойство `clear` используется для предотвращения обтекания элементов. Например, `clear: both;` заставляет элемент начинать с новой строки после всех плавающих элементов.

**\*\*\*\*4. Как задать фон с помощью изображения?\*\*\*\***

Фоновое изображение задается с помощью свойства `background-image`. Например:

.element {

background-image: url('image.jpg');

}

**\*\*\*\*7. Объясните разницу между абсолютным и фиксированным позиционированием.\*\*\*\***

- \*\*absolute\*\*: Элемент позиционируется относительно ближайшего родителя с ненормальным позиционированием (например, relative). Если такого нет, то относительно окна браузера.

- \*\*fixed\*\*: Элемент фиксируется относительно окна браузера и не перемещается при прокрутке страницы.

**\*\*\*\*8. Как создать адаптивный макет с помощью процентов?\*\*\*\***

Использование процентов для задания ширины элементов позволяет им адаптироваться к размеру родительского контейнера:

**\*\*\*\*9. Что такое CSS Sprites?\*\*\*\***

CSS Sprites — это техника оптимизации загрузки изображений, при которой несколько изображений объединяются в одно. Это уменьшает количество HTTP-запросов к серверу.

**\*\*\*\*12. Что такое CSS Grid Template Areas?\*\*\*\***

CSS Grid Template Areas позволяют задавать макет сетки с использованием именованных областей, что делает код более читаемым:

.container {

display: grid;

grid-template-areas:

'header header'

'sidebar content';

}

**\*\*\*\*14. Что такое `calc()` в CSS?\*\*\*\***

Функция `calc()` позволяет выполнять математические операции для задания значений свойств:

.element {

width: calc(100% - 20px);

}

**\*\*\*\*16. Как управлять порядком отображения элементов с помощью Flexbox?\*\*\*\***

Порядок отображения элементов можно управлять с помощью свойства `order`. Элементы с меньшим значением будут отображаться первыми: