**ЛР 2 CSS**

В разработанный в рамках ЛР1 проект добавить стили для оформления страниц.

**Цель**: закрепить изученные возможности CSS на практике.

Не подключать в проект библиотеки для оформления стилей.

Отключить <https://www.bootstrapcdn.com>, однако можно использовать для вдохновения

1. <https://www.bootstrapcdn.com>,
2. [www.csszengarden.com](http://www.csszengarden.com/),
3. <https://css-tricks.com/snippets/>,

чтобы знакомиться с различными стилями.

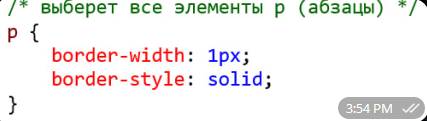
<https://mybrandnewlogo.com/ru/generator-nazvanii-kompanii> - генерация логотипов, названий

Теория:

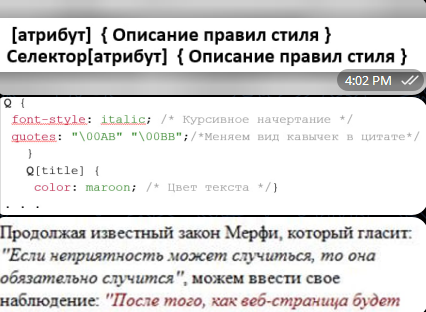
Обязательны для использования в проекте (при желании можно самим выбрать страницы, где нижеуказанное применить):

**селекторы:**

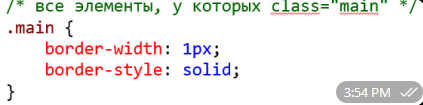
* элементов,



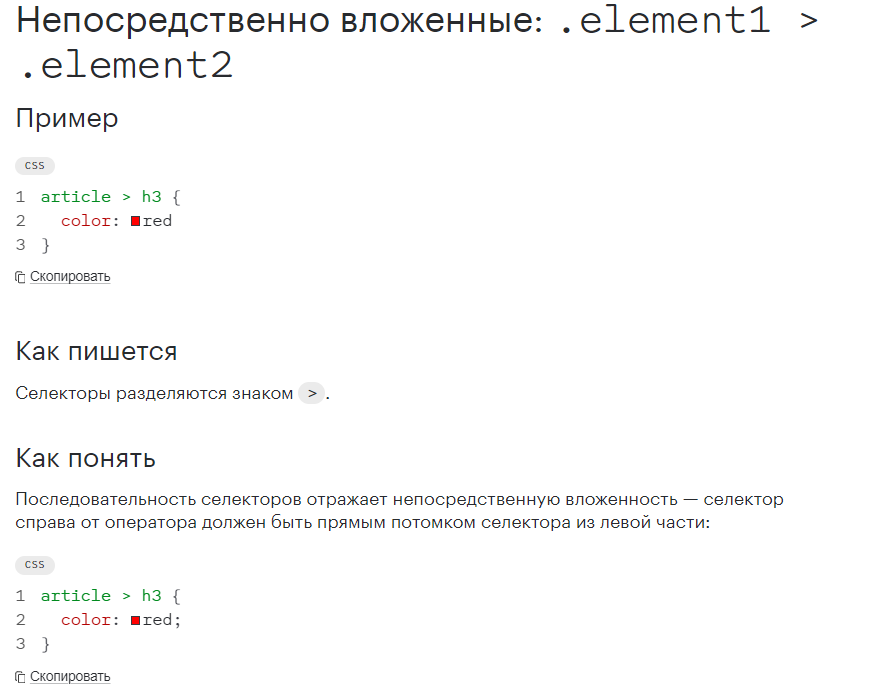
* атрибутов (начинаются с подстроки, заканчиваются подстрокой, атрибут с определенным значением),



* классов,

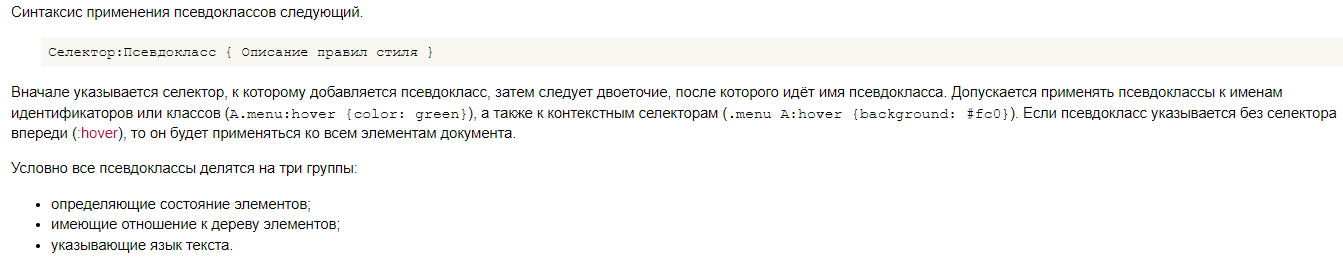


* комбинации селекторов (объединение – AND, селекторы потомков, дочерних элементов)



**псевдоклассы**:

* динамическое состояние для ссылок – различные стили для посещенных, не посещенных, при наведении курсора, нажатых; для элементов в фокусе



1. Опред состояние:  
   active

Link

Focus

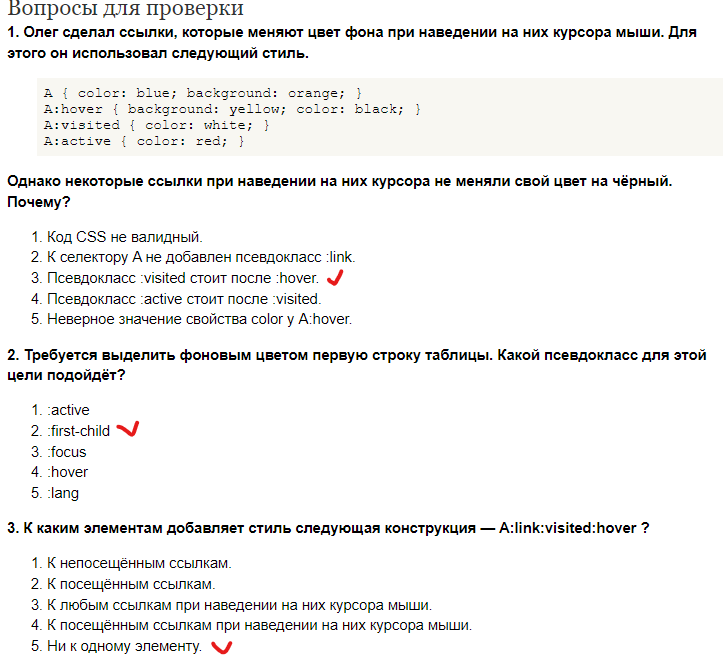
Hover

Visited

1. Имеющие отношение к дереву элементов:

First\_child

1. Указывающие язык текста:  
   lang:



* различные стили для элементов в списке и таблице в зависимости от положения элементов – первый, последний, n-й
* выделить стилями **доступные, обязательные и заблокированные** элементы формы, элементы со **значениями вне диапазона**, включенные **флажки и переключатели**

**/\* Стили для доступных элементов \*/**

**:enabled {**

**/\* Ваш стиль для доступных элементов \*/**

**}** **/\* Стили для обязательных элементов \*/**

**:required {**

**/\* Ваш стиль для обязательных элементов \*/**

**}** **/\* Стили для заблокированных элементов \*/**

**:disabled {**

**/\* Ваш стиль для заблокированных элементов \*/**

**}** **/\* Стили для элементов с недопустимыми значениями \*/**

**:out-of-range {**

**/\* Ваш стиль для элементов с недопустимыми значениями \*/**

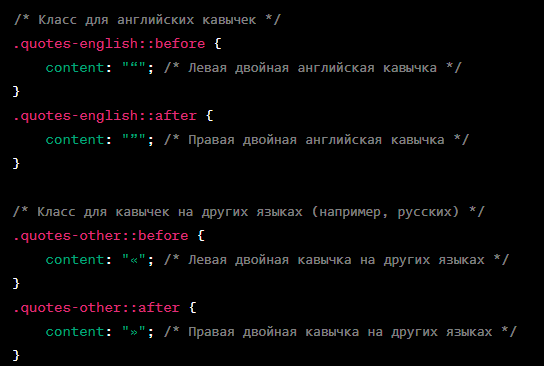
**}** **/\* Стили для включенных флажков и переключателей \*/**

**:checked {**

**/\* Ваш стиль для включенных флажков и переключателей \*/**

**}**

* отображение кавычек в зависимости от **языка** цитаты



**псевдоэлементы**:

* выделить первую букву (или строку) текста абзаца в статье

p::first-letter {

font-size: 150%; /\* Изменить размер первой буквы \*/

font-weight: bold; /\* Сделать первую букву жирной \*/

/\* Другие стили для первой буквы \*/

} p::first-line {

font-size: 120%; /\* Изменить размер первой строки \*/

font-weight: bold; /\* Сделать первую строку жирной \*/

/\* Другие стили для первой строки \*/

}

* добавить контекст в начале и конце фрагмента текста

p::before {

content: "Начало контекста. ";

background-color: yellow; /\* Цвет фона для начала контекста \*/

font-weight: bold; /\* Жирный шрифт для начала контекста \*/

padding: 5px; /\* Поля для начала контекста \*/

}

p::after {

content: " Конец контекста.";

background-color: yellow; /\* Цвет фона для конца контекста \*/

font-weight: bold; /\* Жирный шрифт для конца контекста \*/

padding: 5px; /\* Поля для конца контекста \*/

}

* один из списков оформить маркерами – символами юникода

<ul class="unicode-list">

<li>&#9658; Пункт 1</li>

<li>&#9658; Пункт 2</li>

<li>&#9658; Пункт 3</li>

</ul>

.unicode-list {

list-style: none; /\* Скрыть стандартные маркеры списка \*/

padding: 0; /\* Убрать внутренние отступы списка \*/

}

.unicode-list li {

margin: 0 0 5px; /\* Установить отступ между пунктами списка \*/

}

* изменять стиль при выделении текста

::selection {

background-color: yellow; /\* Цвет фона при выделении \*/

color: black; /\* Цвет текста при выделении \*/

font-weight: bold; /\* Жирный шрифт при выделении \*/

}

подключить **шрифты**:

* основной (из службы шрифтов),

<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=ИМЯ\_ШРИФТА">

body {

font-family: 'ИМЯ\_ШРИФТА', sans-serif;

}

* альтернативный,

body {

font-family: 'ИМЯ\_ШРИФТА', sans-serif;

}

* семейство шрифтов

body {

font-family: 'ИМЯ\_ШРИФТА', 'ИМЯ\_АЛЬТЕРНАТИВНОГО\_ШРИФТА', sans-serif;

}

**медиа-запросы** для отображения страниц в зависимости от ширины и высоты области отображения

@media screen and (max-width: 600px) {

/\* Ваши стили для устройств с шириной экрана менее 600 пикселей \*/

}

@media screen and (min-height: 800px) {

/\* Ваши стили для устройств с высотой экрана более 800 пикселей \*/

}

@media screen and (max-height: 400px) {

/\* Ваши стили для устройств с высотой экрана менее 400 пикселей \*/

}

@media screen and (max-width: 600px) and (min-height: 800px) {

/\* Ваши стили для устройств с шириной менее 600 пикселей и высотой более 800 пикселей \*/

}

@media screen and (orientation: portrait) {

/\* Ваши стили для устройств в портретной ориентации \*/

}

Изменять **вид курсора** при наведении на кнопки

.my-button {

cursor: pointer;

}

.my-button {

cursor: grab;

}

.my-button {

cursor: grabbing;

}

.my-button {

cursor: grab pointer;

}

Для отображения любого **текстового фрагмента** настроить:

* отступы и поля,
* **margin (внешние отступы)**: Определяет отступы вокруг элемента, которые создают пространство между элементом и другими элементами вокруг него.
* **padding (внутренние отступы)**: Определяет отступы внутри элемента, которые создают пространство между содержимым элемента и его границей.
* **border (граница)**: Определяет стиль, ширину и цвет границы элемента.

.text-fragment {

margin: 20px; /\* Внешние отступы \*/

padding: 10px; /\* Внутренние отступы \*/

border: 1px solid #000; /\* Граница \*/

}

* заголовки (или любой другой фрагмент текста) оформить с помощью text-transform

Свойство **text-transform** позволяет изменять преобразование текста, включая его регистр. С помощью этого свойства вы можете оформить заголовки или другие фрагменты текста в разных стилях. Вот примеры использования различных значений **text-transform**: .lowercase {

text-transform: lowercase;

}

* красную строку,

.first-line {

text-indent: -1em;

padding-left: 1em;

border-left: 2px solid red;

}

* заголовки (или любой другой фрагмент текста) оформить с помощью text-transform

.uppercase {

text-transform: uppercase;

}

* межстрочный интервал,

/\* Задать фиксированный межстрочный интервал \*/

.custom-line-height {

line-height: 1.5; /\* Это число устанавливает отношение межстрочного интервала к размеру шрифта. Например, 1.5 означает, что интервал равен полтора размера шрифта. \*/

}

/\* Задать межстрочный интервал в пикселях \*/

.line-height-pixels {

line-height: 24px; /\* Устанавливает фиксированный интервал в пикселях. \*/

}

/\* Задать межстрочный интервал в процентах от размера шрифта \*/

.line-height-percent {

line-height: 150%; /\* Это устанавливает интервал в 150% от размера шрифта. \*/

}

* любой фрагмент оформить капителью (или использовать расширенные возможности font-variant )

.capitalized-text {

text-transform: capitalize;

}

* интервал между словами и символами

.word-spacing {

word-spacing: 5px; /\* Задает интервал между словами в пикселях \*/

}

.letter-spacing {

letter-spacing: 2px; /\* Задает интервал между символами в пикселях \*/

}

.line-spacing {

line-height: 1.5; /\* Задает интервал между строками как множитель текущего размера шрифта \*/

}

* перенос и разрыв слов

.word-break {

word-break: break-all; /\* Позволяет словам переноситься на следующую строку при необходимости \*/

word-wrap: break-word; /\* Позволяет длинным словам переноситься на следующую строку, если они не умещаются в контейнер \*/

white-space: pre-line; /\* Сохраняет переносы строк в тексте (для сохранения переносов вручную в тексте) \*/

}

.hyphens {

hyphens: auto; /\* Разрешает автоматический перенос слов в тексте с учетом правил разделения слов (доступен не во всех браузерах) \*/

overflow-wrap: break-word; /\* Позволяет длинным словам разбиваться на несколько строк, чтобы предотвратить горизонтальный скроллинг \*/

}

* шрифты,

.custom-font {

font-family: "Your Custom Font", sans-serif; /\* Замените "Your Custom Font" на название вашего шрифта \*/

font-size: 16px; /\* Размер шрифта \*/

font-weight: bold; /\* Толщина шрифта \*/

}

* выравнивание текста,

.text-align-center {

text-align: center; /\* Выравнивание текста по центру \*/

}

.text-align-right {

text-align: right; /\* Выравнивание текста вправо \*/

}

.text-align-justify {

text-align: justify; /\* Выравнивание текста по ширине \*/

}

* позиционирование блоков для расположения на странице

.position-relative {

position: relative; /\* Относительное позиционирование \*/

top: 10px; /\* Смещение блока вверх на 10 пикселей \*/

left: 20px; /\* Смещение блока влево на 20 пикселей \*/

}

.position-absolute {

position: absolute; /\* Абсолютное позиционирование \*/

top: 50%; /\* Смещение блока на 50% высоты родительского контейнера \*/

left: 0; /\* Смещение блока влево до левого края родительского контейнера \*/

}

.position-fixed {

position: fixed; /\* Фиксированное позиционирование (относительно видимой области окна браузера) \*/

top: 0;

right: 0;

}

Настроить задний фон сайта:

body {

background-image: url('your-background-image.jpg'); /\* Укажите путь к изображению фона \*/

background-size: cover; /\* Растянуть фон на всю ширину и высоту \*/

background-repeat: no-repeat; /\* Запретить повторение фона \*/

background-attachment: fixed; /\* Зафиксировать фон, чтобы он не прокручивался вместе со страницей \*/

background-color: #f0f0f0; /\* Цвет фона, если изображение не загрузится или не покроет всю страницу \*/

}

Для отображения каталога товаров/услуг (инфо из ЛР4) использовать **CSS Grid Layout:**

<div class="catalog">

<div class="item">Товар 1</div>

<div class="item">Товар 2</div>

<div class="item">Товар 3</div>

<!-- Добавьте больше элементов каталога -->

</div>

.catalog {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(250px, 1fr)); /\* Сетка с автоматически подстраивающимися колонками \*/

grid-gap: 20px; /\* Расстояние между элементами \*/

}

.item {

background-color: #f0f0f0;

padding: 20px;

border: 1px solid #ccc;

}

**На Главной странице:**

* Для списка компаний партнеров использовать блоки круглой формы, настроить для них ширину, цвет, тип и тень.

```css

/\* Стили для списка компаний партнеров \*/

.partner-list {

/\* ... \*/

}

.partner {

width: 150px;

background-color: #007bff;

color: #fff;

text-align: center;

padding: 20px;

border-radius: 50%; /\* Круглая форма \*/

box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.3);

}

```

* Настроить **шрифт** и **кернинг** для наименования компании

```css

/\* Стили для наименования компании \*/

.company-name {

font-family: Arial, sans-serif; /\* Настройка шрифта \*/

font-size: 18px;

letter-spacing: 1px; /\* Кернинг \*/

}

* Использовать **анимацию** для объектов на баннере (комбинация **animation** и **transform** обязательна)

(комбинация animation и transform обязательна).\*\*

```css

/\* Анимация для объектов на баннере \*/

@keyframes moveObject {

0% {

transform: translateX(0);

}

50% {

transform: translateX(100px);

}

100% {

transform: translateX(0);

}

}

.banner-object {

animation: moveObject 4s linear infinite;

}

```

* Для отображения инфо о последней статье использовать **позиционирование** (**position**)

```css

/\* Стили для отображения инфо о последней статье \*/

.latest-article {

position: relative;

}

```

* Область навигации по сайту (наименование страниц) должна фиксироваться при прокрутке остальной страницы.

/\* Фиксированная навигация по сайту \*/

nav {

position: fixed;

top: 0;

background-color: #fff;

box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);

z-index: 999;

}

**О компании**:

* Использовать трансформацию для отображения логотипа (функция на выбор)
  + .company-logo {  
     max-width: 150px; /\* Увеличьте максимальную ширину для увеличения размера логотипа \*/  
     width: 100%; /\* Занимать всю доступную ширину \*/  
     display: block;  
     margin: 10px auto;  
     border: 5px solid #fff; /\* Добавление белой рамки \*/  
     border-radius: 50%; /\* Создание круглой рамки \*/  
     transform-style: preserve-3d; /\* Включение 3D-пространства \*/  
     transform: rotateX(170deg) rotateY(145deg) rotateZ(145deg); /\* Вращение вокруг всех трех осей \*/  
    }

1. `translate()`: Перемещает элемент по оси X и Y.

Пример: `transform: translate(20px, 30px);`

2. `rotate()`: Поворачивает элемент на указанный угол.

Пример: `transform: rotate(45deg);`

3. `scale()`: Масштабирует элемент по осям X и Y.

Пример: `transform: scale(1.5, 2);`

4. `skew()`: Наклоняет элемент по осям X и Y.

Пример: `transform: skew(30deg, 45deg);`

5. `matrix()`: Применяет 2D-трансформацию с помощью матрицы.

Пример: `transform: matrix(a, b, c, d, e, f);`

6. `perspective()`: Устанавливает точку перспективы для 3D-трансформаций.

Пример: `transform: perspective(100px) translate3d(0, 0, 50px);`

7. `rotateX()`, `rotateY()`, `rotateZ()`: Поворачивает элемент вокруг осей X, Y или Z.

Пример: `transform: rotateX(45deg);`

8. `scaleX()`, `scaleY()`, `scaleZ()`: Масштабирует элемент по отдельным осям.

Пример: `transform: scaleX(1.5);`

9. `translateX()`, `translateY()`, `translateZ()`: Перемещает элемент по отдельным осям.

Пример: `transform: translateX(20px);`

10. `perspective-origin`: Устанавливает точку перспективы для 3D-трансформаций.

Пример: `transform-origin: 50% 50%;`

* Для сертификата использовать изображения для границ из графического файла (не забыть про вендорные префиксы – приставка к свойству css, позволяющая обеспечить некую функцию в котором браузере она не внедрена и тп),
* /\* Стили для ссылки на сертификат \*/  
  .certificate-link {  
   display: block;  
   text-align: center;  
   padding: 10px;  
   text-decoration: none;  
   margin: 20px auto;  
   max-width: 150px;  
   font-weight: bold;  
   border-width: 2px; /\* Ширина границы \*/  
   border-style: solid;  
   border-image-source: url('static/background/mobile\_background.jpg'); /\* Путь к изображению границы \*/  
    
   /\* Добавляем вендорные префиксы \*/  
   -webkit-border-image-source: url('static/background/mobile\_background.jpg'); /\* Для WebKit (Chrome, Safari) \*/  
   -moz-border-image-source: url('static/background/mobile\_background.jpg'); /\* Для Mozilla Firefox \*/  
   -o-border-image-source: url('static/background/mobile\_background.jpg'); /\* Для Opera \*/  
   -ms-border-image-source: url('static/background/mobile\_background.jpg'); /\* Для Internet Explorer и Microsoft Edge \*/  
    
   border-image-slice: 1; /\* Использовать всё изображение для границы \*/  
    
   /\* Добавляем вендорные префиксы \*/  
   -webkit-border-image-slice: 1;  
   -moz-border-image-slice: 1;  
   -o-border-image-slice: 1;  
   -ms-border-image-slice: 1;  
  }
* фоновое изображение,
* .main-container {  
   background-image: url('/static/background/background.jpg');  
   background-size: cover;  
   background-repeat: no-repeat;  
   background-attachment: fixed;  
   opacity: 1; /\* Полностью непрозрачный фон \*/  
   color: #fff; /\* Белый цвет текста \*/  
   font-family: 'Arial', sans-serif; /\* Основной шрифт из службы шрифтов \*/  
   line-height: 1.5; /\* Межстрочный интервал \*/  
   padding: 20px; /\* Отступы вокруг текста \*/  
  }
* градиент,
* background: radial-gradient(ellipse at center, #673ab7 0%, rgba(2,0,36,1) 25%, rgba(79,1,91,1) 100%); /\* Deep Purple \*/

1. \*\*Линейный градиент (Linear Gradient)\*\*:

- Горизонтальный градиент: Идет от левого края к правому.

- Вертикальный градиент: Идет от верхнего края к нижнему.

- Диагональный градиент: Идет по диагонали от одного угла к другому.

- Пользовательский угол: Вы можете настроить угол направления градиента.

```css

linear-gradient(to right, red, yellow);

linear-gradient(to bottom, green, blue);

linear-gradient(45deg, orange, purple);

linear-gradient(135deg, #FF5733, #5733FF);

```

2. \*\*Радиальный градиент (Radial Gradient)\*\*:

- Радиальный градиент начинается из центра элемента и распространяется наружу или наоборот.

- Можно настроить размер и форму градиента.

```css

radial-gradient(circle, red, yellow);

radial-gradient(ellipse, green, blue);

radial-gradient(at top right, orange, purple);

```

3. \*\*Повторяющийся градиент (Repeating Gradient)\*\*:

- Позволяет создавать повторяющиеся градиенты.

```css

repeating-linear-gradient(to right, red, yellow 20px);

repeating-radial-gradient(circle, red, yellow 20px);

```

4. \*\*Указание цветовых остановок (Color Stops)\*\*:

- Можно определить несколько цветовых остановок, чтобы создать более сложные градиенты.

```css

linear-gradient(to right, red, yellow, green);

radial-gradient(circle, red 20%, yellow 40%, green 60%);

```

5. \*\*Инверсный градиент (Inverted Gradient)\*\*:

- Позволяет создавать градиенты, в которых цвета инвертированы, то есть начинаются с фонового цвета и переходят к заднему плану.

```css

linear-gradient(to right, from(background) to text);

```

6. \*\*Анимированные градиенты (Animated Gradients)\*\*:

- Можно создавать анимацию в градиентах с помощью CSS анимации.

```css

@keyframes gradientAnimation {

0% {

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

}

50% {

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

}

100% {

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

}

}

.element {

animation: gradientAnimation 5s infinite;

}

```

* z-index,
* z-index: -1;

концепция трехмерного размещения html элементов по оси z, располож перпенд экрану

`z-index` - это CSS свойство, которое определяет порядок накладывания элементов (слоев) в трехмерном пространстве веб-страницы. Это свойство контролирует, какие элементы находятся "поверх" или "под" другими элементами.

Основные моменты `z-index`:

1. \*\*Значение `z-index`\*\*:

- Положительные целые числа: Элементы с более высокими значениями `z-index` находятся поверх элементов с меньшими значениями.

- Отрицательные целые числа: Элементы с более низкими значениями `z-index` находятся поверх элементов с более высокими значениями.

- Значение `auto`: Элементы с `z-index: auto` располагаются в стандартном порядке относительно других элементов на странице.

2. \*\*Элементы с `position`\*\*:

- `z-index` применяется только к элементам, которые имеют установленное свойство `position`, такие как `position: relative`, `position: absolute`, или `position: fixed`. Элементы с `position: static` не наследуют `z-index`.

3. \*\*Уровень вложенности\*\*:

- `z-index` работает на уровне элемента и его детей. Если у внешнего элемента `z-index` больше, чем у внутреннего, внешний элемент будет находиться поверх внутреннего.

Пример:

```html

<div class="container">

<div class="box" style="z-index: 2;">Box 1</div>

<div class="box" style="z-index: 1;">Box 2</div>

</div>

```

```css

.container {

position: relative;

}

.box {

position: absolute;

width: 100px;

height: 100px;

}

```

В этом примере, `Box 1` будет находиться поверх `Box 2`, потому что у него выше `z-index`. При этом, `container` также может иметь свой собственный `z-index`, который повлияет на его дочерние элементы.

Значение `z-index` полезно для управления порядком отображения сложных компонентов на веб-странице, таких как меню, всплывающие окна и слайдеры.

* настройку шрифтов,
* @font-face {  
   font-family: 'IBM Plex Serif';  
   src: url('https://fonts.google.com/specimen/IBM+Plex+Serif') format('woff2'); /\* Замените ссылку на фактическую \*/  
   font-weight: normal;  
   font-style: normal;  
  }  
    
  .company-description {  
   background: radial-gradient(ellipse at center, #673ab7 0%, rgba(2,0,36,1) 25%, rgba(79,1,91,1) 100%); /\* Deep Purple \*/  
   color: white; /\* Цвет текста \*/  
   padding: 20px;  
   border-radius: 10px;  
   margin: 0 10px;  
   text-indent: 1em; /\* Красная строка \*/  
   font-weight: bold; /\* Жирный шрифт \*/  
   font-family: 'IBM Plex Serif', sans-serif; /\* Применение стороннего шрифта \*/  
  }

1. \*\*Системные шрифты\*\*: Это шрифты, которые уже установлены на устройствах пользователей (например, Arial, Times New Roman). Они не требуют дополнительной загрузки и отображаются быстро, но выбор ограничен системными шрифтами.

Пример:

```css

font-family: Arial, sans-serif;

```

2. \*\*Шрифты Google Fonts\*\*: Google Fonts предоставляет бесплатные шрифты, которые вы можете подключить к вашему веб-сайту, добавив ссылку на шрифт в вашем HTML-коде. Это позволяет использовать разнообразные шрифты без их загрузки на сервер.

Пример:

```html

<link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">

``

И CSS:

```css

font-family: 'Open Sans', sans-serif;

```

3. \*\*Локальные шрифты\*\*: Вы также можете загрузить шрифты на свой сервер и использовать их на веб-сайте. Вам нужно удостовериться, что формат шрифта поддерживается браузерами.

Пример:

```css

@font-face {

font-family: 'MyCustomFont';

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

}

```

И затем:

```css

font-family: 'MyCustomFont', sans-serif;

```

4. \*\*Шрифты из CDN (Content Delivery Network)\*\*: Вы можете использовать CDN для загрузки шрифтов, таких как Font Awesome, в ваш проект. Это обычно выполняется путем добавления ссылки на стилевые таблицы и использования классов или символов для отображения иконок.

Пример (Font Awesome):

```html

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">

```

Использование классов:

```html

<i class="fas fa-camera"></i>

```

* выравнивание текста и т.д.

Выравнивание текста в CSS можно настроить с помощью свойства **text-align**. Это свойство определяет, как текст внутри элемента будет выравниваться горизонтально относительно его контейнера. Доступны следующие варианты для **text-align**:

1. **left**: Текст выравнивается по левому краю контейнера.
2. **right**: Текст выравнивается по правому краю контейнера.
3. **center**: Текст выравнивается по центру контейнера.
4. **justify**: Текст выравнивается по обоим краям контейнера, так что промежутки между словами могут изменяться, чтобы заполнить всю доступную ширину контейнера.

* Настроить оформление остальных элементов страницы – история по годам списком, реквизиты выделить шрифтом, видео/аудио -стиль выбрать самим.
* /\* Стили для списка с маркерами Юникода \*/  
  .ol-marker-list {  
   list-style-type: none; /\* Скрыть маркеры по умолчанию \*/  
  }  
    
  .ol-marker-list li::before {  
   content: '\2022'; /\* Символ Юникода для маркера в виде точки • \*/  
   color: #FF5733; /\* Цвет маркера \*/  
   font-size: 20px; /\* Размер маркера \*/  
   margin-right: 10px; /\* Отступ между маркером и текстом \*/  
  }



**Новости**:

* список статей (7 статей) в соответствии с тематикой сайта с заголовком,
  + добавляется через панель администратора
* кратким содержанием (одно предложение),
  + -//-
* картинкой и кнопкой «Читать далее» при нажатии на которую открывается вся статья – использовать **overflow** для краткой информации о статье и многоточие, когда краткая информация не помещается в выделенную область.
* **многоколоночный макет** для отображения текста статьи

Многоколоночный макет (или многоколоночное форматирование) - это метод форматирования текста и медиа, который позволяет разделить содержимое страницы на несколько вертикальных колонок. Это означает, что текст и изображения могут размещаться в нескольких колонках рядом друг с другом, а не только в одной колонке, как это обычно бывает на веб-страницах.

Преимущества многоколоночных макетов:

1. Увеличение читаемости: Многоколоночные макеты делают длинные тексты более читаемыми, так как читатель может легче скользить глазами по тексту.

2. Экономия места: Этот макет позволяет более эффективно использовать экранное пространство, особенно на больших мониторах.

3. Визуальный интерес: Многоколоночные макеты могут придать странице визуальный интерес, особенно если содержимое размещено в разных колонках.

4. Легкость чтения: Отдельные колонки могут использоваться для разных категорий информации, что делает навигацию более удобной.

5. Поддержка адаптивного дизайна: Многоколоночные макеты могут быть адаптированы для работы на разных устройствах, от мониторов до мобильных устройств.

Для создания многоколоночных макетов веб-разработчики используют CSS-свойство `column-count`, которое определяет количество колонок на странице, и другие свойства для настройки разрывов и отступов между колонками. Это позволяет легко создавать более сложные и интересные макеты для веб-сайтов.

**Словарь терминов и понятий**:

* список часто-задаваемых вопросов с датой добавления на сайт,
* при нажатии на которые открывается развернутый ответ – выбрать стиль самим

1. \*\*`<details>` и `<summary>`:\*\* Эти элементы HTML используются для создания простых выпадающих списков или аккордеонов. `<details>` определяет контейнер с дополнительной информацией, которая может быть скрыта или раскрыта, а `<summary>` - заголовок этого контейнера. Пример:

```html

<details>

<summary>Заголовок</summary>

<p>Скрытый текст</p>

</details>

```

Пользователи могут щелкнуть на заголовок (`<summary>`), чтобы отобразить или скрыть скрытый текст.

2. \*\*`<select>` и `<option>`:\*\* Эти элементы HTML используются для создания выпадающих списков с выбором одного элемента из списка. `<select>` создает выпадающий список, а `<option>` - элементы списка. Пример:

```html

<select>

<option value="option1">Option 1</option>

<option value="option2">Option 2</option>

<option value="option3">Option 3</option>

</select>

```

Пользователи могут выбирать один из элементов из выпадающего списка.

3. \*\*CSS-стилизация:\*\* Вы также можете использовать CSS для стилизации элементов `<select>` и `<option>` и изменения их внешнего вида.

Каждый из этих методов имеет свои особенности и подходит для разных случаев использования. `<details>` и `<summary>` хороши для простых выпадающих блоков информации, в то время как `<select>` и `<option>` используются для создания выпадающих списков выбора. Самый подходящий метод зависит от конкретных потребностей вашего веб-проекта.

**Контакты**:

* Фото сотрудников с описанием выполняемых работ, телефонами, почтой и т.д. - использовать **Flexbox** для размещения инфо о каждом сотруднике

Flexbox (сокращение от "Flexible Box") - это метод компоновки элементов в веб-дизайне и верстке, предназначенный для упрощения размещения и выравнивания элементов на веб-странице в отношении друг друга внутри их контейнера. Этот метод позволяет создавать более сложные макеты и легко управлять распределением пространства и выравниванием элементов, в том числе при изменении размеров экрана и адаптивной верстке.

Основные преимущества Flexbox включают:

1. \*\*Удобное распределение пространства:\*\* Flexbox автоматически распределяет доступное пространство между элементами внутри контейнера. Это особенно полезно для создания макетов с динамическим изменением размеров экрана или контейнера.

2. \*\*Простое выравнивание элементов:\*\* Выравнивание элементов внутри контейнера становится проще, так как Flexbox предоставляет множество свойств для управления выравниванием по горизонтали и вертикали.

3. \*\*Порядок элементов:\*\* Flexbox позволяет изменять порядок элементов без изменения порядка размещения в HTML. Это полезно для создания адаптивных макетов.

4. \*\*Автоматическая коррекция размеров:\*\* Flexbox автоматически корректирует размеры элементов внутри контейнера, чтобы они вписывались в доступное пространство.

Применение Flexbox особенно полезно при создании навигационных меню, карточек, нижних панелей и других компонентов веб-сайта, где важно правильное размещение элементов. Этот метод стал широко используемым в веб-дизайне и предоставляет удобные средства для создания адаптивных и удобных макетов.

**Политика конфиденциальности**:

* в соответствии с тематикой разработанного сайта – свой стиль

**Вакансии**:

* список вакансий с описанием – использовать **плавающие блоки** (**float**, **clear** )

Вам стоит знать, что использование **float** и **clear** для создания макетов становится менее популярным, так как сейчас более современные методы, такие как Flexbox и CSS Grid, предоставляют более простой и надежный способ управления макетами. Однако, если вы хотите создать макет с использованием **float** и **clear**, вот пример CSS для списка вакансий:

/\* Стили для отдельной вакансии \*/  
.vacancy-item {  
 width: 100%; /\* Размер одной вакансии (для двух вакансий в ряду) \*/  
 float: left; /\* Перенос элементов влево \*/  
 clear: both; /\* Предотвращение перекрытия с предыдущими элементами \*/  
 margin-bottom: 20px;  
 padding: 20px;  
 border: 1px solid #ccc;  
 border-radius: 10px; /\* Закругленные углы для карточек \*/  
 text-align: left; /\* Выравнивание текста влево \*/  
}

**Отзывы**:

* Для настройки шрифтов на компонентах формы использовать ключевые слова, соответствующие типу компонента, не забыть о псевдоклассах (см. выше)

Ключевые слова в CSS представляют собой зарезервированные слова, которые используются для определения различных свойств стилей. Эти слова являются предопределенными и выполняют определенные функции в CSS.

Например, некоторые общие ключевые слова в CSS включают:

1. `inherit`: Это ключевое слово используется для наследования свойства от родительского элемента.

2. `initial`: Это ключевое слово возвращает свойство к его начальному значению, как оно определено в спецификации CSS.

3. `unset`: Это ключевое слово сначала пытается наследовать свойство, а если наследование невозможно, оно ведет себя как `initial`.

4. `none`: Это ключевое слово часто используется для отключения какого-либо свойства, такого как `list-style` или `text-decoration`.

5. `auto`: Это ключевое слово используется, чтобы браузер самостоятельно установил значение свойства, основываясь на контексте и содержании элемента.

6. `transparent`: Это ключевое слово используется для задания прозрачного цвета.

7. `currentColor`: Это ключевое слово устанавливает значение цвета в текущее значение `color` для элемента.

**Промокоды и купоны**: список промокодов действующих и в архиве.

**Таблицы:**

* убрать двойные линии между ячейками внутри таблицы

/\* Убираем двойные линии между ячейками в таблице \*/

table {

border-collapse: collapse;

}

* изменить фон в пустых ячейках
* оформить заголовок таблицы над таблицей

/\* Стили для заголовка таблицы \*/

caption {

font-weight: bold;

text-align: center;

font-size: 24px;

}

* применить горизонтальное и вертикальное выравнивание информации в ячейках и настроить расположение таблицы на странице

Эти стили будут зависеть от ваших предпочтений и макета. Обычно используют **text-align** для горизонтального выравнивания и **vertical-align** для вертикального выравнивания. Расположение таблицы на странице можно настраивать с помощью стилей для самой таблицы или контейнера вокруг неё

* стилизовать строки и столбцы

Эти стили будут зависеть от ваших предпочтений и макета. Обычно используют **text-align** для горизонтального выравнивания и **vertical-align** для вертикального выравнивания. Расположение таблицы на странице можно настраивать с помощью стилей для самой таблицы или контейнера вокруг неё

/\* Стили для активных промокодов \*/

.active-promo {

background-color: #9effab; /\* Пример цвета фона \*/

}

/\* Стили для архивных промокодов \*/

.archived-promo {

background-color: #f3f3f3; /\* Пример цвета фона \*/

}

/\* Стили для всех ячеек в первом столбце (Code) \*/

.promo-table td:nth-child(1) {

color: #007BFF; /\* Пример цвета текста \*/

}