|  |
| --- |
| Школа itProger |
| Изучение CSS для новичков |
| Переложение видео уроков |

|  |
| --- |
| Aliadm  Bish 2024 |

Оглавление

[Изучение CSS для новичков / Урок #1 – Введение в CSS. Что это и как с ним работать? 2](#_Toc171689851)

[Введение в Cascading Style Sheets (CSS) 2](#_Toc171689852)

[Установка текстового редактора Atom 3](#_Toc171689853)

[Изучение CSS для новичков / Урок #2 – Форматы подключения стилей 3](#_Toc171689854)

[Создание нового файла и добавление стилей 3](#_Toc171689855)

[Способы подключения стилей 3](#_Toc171689856)

[Использование тега style 3](#_Toc171689857)

[Подключение стилей к HTML-странице 4](#_Toc171689858)

[Создание и подключение CSS-файла 4](#_Toc171689859)

[Варианты подключения CSS 4](#_Toc171689860)

[Изучение CSS для новичков / Урок #3 – Селекторы для выборки элементов 4](#_Toc171689861)

[Селекторы в CSS 4](#_Toc171689862)

[Классы и идентификаторы 5](#_Toc171689863)

[Селекторы выбора 5](#_Toc171689864)

[Вложенные селекторы 5](#_Toc171689865)

[Группировка селекторов 5](#_Toc171689866)

[Изучение CSS для новичков / Урок #4 – Псевдоклассы и псевдоэлементы 6](#_Toc171689867)

[Псевдоклассы и псевдоэлементы 6](#_Toc171689868)

[Примеры использования псевдоклассов 6](#_Toc171689869)

[Псевдокласс “active” 7](#_Toc171689870)

[Псевдоклассы и их использование 7](#_Toc171689871)

[Псевдоэлементы и их использование 8](#_Toc171689872)

[Работа с псевдоклассами 8](#_Toc171689873)

[Изучение CSS для новичков / Урок #5 – Работа с фоновыми картинками 9](#_Toc171689874)

[Стили для работы с фоновыми картинками 9](#_Toc171689875)

[Установка заднего фона 9](#_Toc171689876)

[Повторение картинки и позиция 9](#_Toc171689877)

[Работа с задним фоном 9](#_Toc171689878)

[Использование свойства “background-size” 10](#_Toc171689879)

[Использование свойства “background-attachment” 10](#_Toc171689880)

[Изучение CSS для новичков / Урок #6 – Стили для текста 10](#_Toc171689881)

[Работа с текстом 10](#_Toc171689882)

[Дополнительные свойства 11](#_Toc171689883)

[Работа с шрифтами 11](#_Toc171689884)

[Свойства текста 11](#_Toc171689885)

[Изучение CSS для новичков / Урок #7 – Стили для блоков 12](#_Toc171689886)

[Создание блока с помощью CSS 12](#_Toc171689887)

[Установка высоты и ширины блока 12](#_Toc171689888)

[Удаление внутренних и внешних отступов 13](#_Toc171689889)

[Работа с блочными и строчными объектами 13](#_Toc171689890)

[Изучение CSS для новичков / Урок #8 – Позиционирование блоков 14](#_Toc171689891)

[Позиционирование объектов с помощью CSS 14](#_Toc171689892)

[Использование свойства “float” 14](#_Toc171689893)

[Создание веб-сайта 14](#_Toc171689894)

[Стилизация шапки 15](#_Toc171689895)

[Стилизация футера 15](#_Toc171689896)

[Стилизация основной части и боковой панели 15](#_Toc171689897)

[Позиционирование объектов 16](#_Toc171689898)

[Позиционирование с помощью свойства “position:” 16](#_Toc171689899)

[Абсолютное позиционирование с помощью свойства “position: absolute;” 16](#_Toc171689900)

[Фиксация объекта с помощью свойства “fixed” 17](#_Toc171689901)

[Изучение CSS для новичков / Урок #9 – Работа со списками 17](#_Toc171689902)

[Работа со списками через CSS 17](#_Toc171689903)

# Изучение CSS для новичков / Урок #1 – Введение в CSS. Что это и как с ним работать?

[Ссылка на видео](https://youtu.be/hft4XYApT44?list=PLDyJYA6aTY1meZ3d08sRILB46OJ-wojF2)

## Введение в Cascading Style Sheets (CSS)

**00:00**

* Видео начинается с приветствия и обсуждения плана курса по изучению языка стилей CSS.
* CSS - это каскадные таблицы стилей, которые позволяют описывать стили для любых объектов в HTML.
* Без CSS невозможно создать красивый веб-сайт, так как он позволяет добавить стили к элементам.

## Установка текстового редактора Atom

**02:55**

* В этом уроке автор устанавливает текстовый редактор Atom, который будет использоваться для написания кода.
* Установка Atom происходит без подводных камней, и после запуска открывается окно с настройками.
* Автор показывает, как изменить тему и формат подсветки кода в настройках Atom.

# Изучение CSS для новичков / Урок #2 – Форматы подключения стилей

[Ссылка на видео](https://youtu.be/DX9ITfNVM7I?list=PLDyJYA6aTY1meZ3d08sRILB46OJ-wojF2)

## Создание нового файла и добавление стилей

**00:00**

* Создание новой папки для хранения веб-сайта и открытие ее в текстовом редакторе Atom.
* Создание файла index.html с основным HTML-синтаксисом и добавление текста-заглушки.

## Способы подключения стилей

**02:46**

* Три формата подключения стилей: атрибут style, тег style и тег style внутри тега head.
* Атрибут style подходит для добавления одного или двух стилей, тег style - для нескольких стилей.

## Использование тега style

**06:39**

* Тег style внутри тега head для подключения стилей к определенным элементам.
* Пример: выбор всех тегов p и добавление стиля color: red.
* <style>  
   p {  
   color: red;  
   }  
  </style>

## Подключение стилей к HTML-странице

**07:36**

* В видео обсуждается три формата подключения стилей к HTML-странице: через тег style, через атрибут style и через отдельный файл CSS.
* Первый формат, через тег style, добавляет стиль ко всем тегам, указанным в этом теге (см. выше).
* Второй формат, через атрибут style, добавляет стиль к конкретному элементу, указанному в этом атрибуте.
* <p style="color: red;"></p>
* Третий формат, через отдельный файл CSS, является наиболее правильным и удобным, так как позволяет отделить CSS от HTML-кода.
* <link rel="stylesheet" href="css/main.css">

## Создание и подключение CSS-файла

**09:33**

* В видео показано, как создать отдельный файл CSS и подключить его к HTML-странице с помощью тега link.
* В файле CSS можно прописать любые свойства и селекторы, которые будут применены к HTML-коду.
* body {  
   color: red;  
  }

## Варианты подключения CSS

**11:25**

* В видео обсуждаются три варианта подключения CSS к HTML-странице: через тег style, через атрибут style и через отдельный файл CSS.
* Третий вариант, через отдельный файл CSS, является наиболее правильным и удобным, так как позволяет отделить CSS от HTML-кода.

# Изучение CSS для новичков / Урок #3 – Селекторы для выборки элементов

[Ссылка на видео](https://youtu.be/LQxx5Z9XEdM?list=PLDyJYA6aTY1meZ3d08sRILB46OJ-wojF2)

## Селекторы в CSS

**00:00**

* В этом уроке изучаются различные селекторы, которые позволяют выбирать нужные HTML-объекты для добавления стилей.
* В качестве селектора можно использовать название тега, к которому нужно добавить стили.
* Также можно использовать классы и идентификаторы для выбора объектов.

## Классы и идентификаторы

**01:51**

* Классы позволяют быстро выбрать нужные объекты и добавить к ним свойства.
* Идентификаторы должны быть уникальными на странице и используются для одиночных объектов, таких как шапка или футер сайта.
* В обоих случаях для обращения к классу или идентификатору нужно указать его название и знак решетки.

## Селекторы выбора

**07:31**

* В видео обсуждаются три формата выбора различных HTML-объектов: по названию тэга, по классу и по идентификатору.
* Обсуждаются вложенные селекторы выбора, которые позволяют выбирать объекты, находящиеся внутри других объектов.

## Вложенные селекторы

**10:24**

* В видео объясняется, как использовать вложенные селекторы для добавления стилей только к объектам, находящимся внутри других объектов.
* Приводится пример использования вложенных селекторов для добавления стилей только к тэгам b, находящимся внутри тэга div.

## Группировка селекторов

**12:11**

* В видео объясняется, как группировать селекторы для добавления стилей к нескольким объектам одновременно.
* Приводится пример использования запятой для указания дополнительных селекторов, к которым будут добавлены стили.

Пример:

**Файл index.html:**

```html

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Reprehenderit quas nobis laborum nam sequi expedita officia distinctio aperiam, voluptate minus, asperiores dolores officiis perferendis! Quibusdam debitis veritatis neque excepturi at.

Lorem ipsum dolor sit, amet <b>consectetur</b> adipisicing elit. Ducimus dolorum dignissimos sint, <i>accusantium</i> nulla a quos explicabo maiores cum earum, blanditiis ad repellendus at eligendi odit delectus laboriosam, reiciendis qui!  
</div>

**Файл main.css:**

.main-text { color: red; }

#text-in-p { color: blue; }

#bold-text-in-p { color: green; }

b { color: black; }

div b, div i { color: brown; }

/*div, .class-name, #id, div p { color: red }*/

# Изучение CSS для новичков / Урок #4 – Псевдоклассы и псевдоэлементы

[Ссылка на видео](https://youtu.be/nZHrCDJEnw4?list=PLDyJYA6aTY1meZ3d08sRILB46OJ-wojF2)

## Псевдоклассы и псевдоэлементы

**00:00**

* В языке CSS существуют понятия псевдоклассов и псевдоэлементов.
* Псевдоклассы позволяют управлять свойствами CSS для различных состояний объекта.
* Например, можно изменить цвет ссылки при наведении на нее.
* Псевдоэлементы позволяют добавлять стили к определенным частям элементов, таким как текст или подчеркивание.

## Примеры использования псевдоклассов

**01:57**

* В примере с ссылкой можно изменить цвет ссылки и подчеркивание при наведении.
* Также можно указать, что текст будет подчеркиваться при наведении.
* Псевдокласс hover позволяет добавить стили, которые будут применяться к объекту при наведении на него.

**Файл index.html:**

```html Ссылка

**Файл main.css:** ```css .main\_link { text-decoration: none; }

.main\_link:hover { color: blue; text-decoration: underline; text-decoration-color: red; cursor: pointer; }

## Псевдокласс “active”

**08:31**

* Псевдокласс active срабатывает каждый раз, когда пользователь нажимает на объект.
* В примере с текстом можно установить подчеркивание при нажатии.
* Псевдокласс visited добавляет стили, которые будут применяться к посещенным ссылкам.

**Файл index.html:**

```html

Lorem ipsum dolor sit amet consectetur adipisicing elit. Eum autem eveniet placeat doloremque quod vero in aspernatur, sit molestias veritatis fuga vel quaerat beatae! Quam vel doloribus accusantium repellat pariatur.

**Файл main.css:**

```css .main\_link:active { text-decoration: line-through; }

.main\_link:visited { color: grey; }

## Псевдоклассы и их использование

**09:29**

* В видео обсуждается использование псевдоклассов для установки стилей для ссылок, которые уже были посещены.
* Внутри псевдокласса visited обычно прописывается изменение цвета текста.
* Также обсуждаются другие псевдоклассы, такие как focus, disable и enable, которые позволяют выбирать только определенные элементы.

**Файл index.html:**

```html

Smile

**Файл main.css:**

```css input { color: blue; }

input:focus { color: red; }

p.text:not(:empty) { color: grey; }

## Псевдоэлементы и их использование

**17:06**

* В видео объясняется использование псевдоэлементов для работы с определенной частью текста внутри HTML-объекта.
* Обсуждается псевдоэлемент after, который позволяет обратиться к невидимой части текста за объектом.
* Приводится пример использования этого псевдоэлемента для вывода текста после объекта.

**19:00 Использование псевдоэлементов**

* В видео объясняется, как использовать псевдоэлемент after для добавления текста после объекта.
* Также можно использовать псевдоэлемент before для добавления текста перед объектом.
* Псевдоэлемент first-letter позволяет обратиться к первому символу внутри абзаца, а first-line - к первой строке.

**Файл index.html:**

```html

Some text

**Файл main.css:** ```css #some\_id::after { content: “123”; color: red; text-decoration: line-through; }

#some\_id::before { content: “123”; color: red; text-decoration: line-through; }

p.main\_link::first-letter { color: green; }

p.main\_link::first-line { color: magenta; }

## Работа с псевдоклассами

**24:39**

* В видео объясняется, как использовать псевдокласс first-of-type для добавления стиля только к первому элементу в списке.
* Псевдокласс last-of-type позволяет добавить стиль только к последнему элементу в списке.
* Эти псевдоклассы и псевдоэлементы могут быть использованы для работы с различными частями объектов и их состояниями.

**Файл index.html:**

```html

Smile

**Файл main.css:** ```css p:first-of-type { text-decoration: line-through; }

p:last-of-type { text-decoration: line-through; }

# Изучение CSS для новичков / Урок #5 – Работа с фоновыми картинками

[Ссылка на видео](https://youtu.be/k2XTkIW9Y0I?list=PLDyJYA6aTY1meZ3d08sRILB46OJ-wojF2)

## Стили для работы с фоновыми картинками

**00:00**

* В языке CSS существуют различные стили для работы с фоновыми картинками на сайте.
* В этом уроке мы изучим некоторые из них.

## Установка заднего фона

**01:52**

* Для установки заднего фона можно использовать свойство background-color.
* Можно также указать изображение, которое будет добавлено на страницу, используя свойство background-image.

## Повторение картинки и позиция

**04:32**

* Можно указать, каким образом картинка будет повторяться на странице, используя свойство background-repeat.
* Также можно указать, где именно будет находиться картинка, используя свойство background-position.

**Файл main.css:**

```css body { background-color: #7f9ab3; color: blue; background-image: url(‘https://i.artfile.ru/3000x2001\_1534615\_[www.ArtFile.ru].jpg’); background-repeat: no-repeat; background-position: center; }

## Работа с задним фоном

**06:23**

* Создание функции для отображения картинки на заднем фоне.
* Установка свойства no-repeat для предотвращения повторения картинки.
* Установка свойства center для центрирования картинки по вертикали.
* Установка свойства height для указания высоты картинки в процентах от высоты экрана.

## Использование свойства “background-size”

**08:19**

* Установка свойства background-size для указания размера картинки.
* Использование значения cover для растяжения картинки на 100% ширины заднего фона.

## Использование свойства “background-attachment”

**10:15**

* Установка свойства background-attachment для фиксации картинки на заднем фоне.
* Создание двух объектов с разными высотами для демонстрации эффекта.
* Использование свойства background-attachment: fixed для фиксации картинки на заднем фоне.

**Файл index.html:**

```html

**Файл main.css:**

```css #first { /*background-color: #7f9ab3; color: blue; background-image: url(‘https://i.artfile.ru/3000x2001\_1534615\_[www.ArtFile.ru].jpg’); background-repeat: no-repeat; background-position: center;*/

color: blue;  
background: #7f9ab3 url('https://i.artfile.ru/3000x2001\_1534615\_[www.ArtFile.ru].jpg') no-repeat center top;  
height: 500vh;  
background-size: cover;  
background-attachment: fixed;

}

#second { height: 2000px; }

# Изучение CSS для новичков / Урок #6 – Стили для текста

[Ссылка на видео](https://youtu.be/B4x380VocnQ?list=PLDyJYA6aTY1meZ3d08sRILB46OJ-wojF2)

## Работа с текстом

**00:00**

* Создание параграфа и добавление текста.

**Файл index.html:**

```html

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Repellendus necessitatibus quidem quia quibusdam accusantium veritatis dolore ducimus voluptates nulla rem distinctio obcaecati tenetur, quos et corrupti magnam nemo harum adipisci.

* Использование свойства color для установки цвета текста.
* Установка свойства text-align для определения расположения текста.
* Работа с размером шрифта с помощью свойства font-size.
* Установка размера шрифта в относительных единицах с помощью свойства font-size.

## Дополнительные свойства

**01:56**

* Использование свойства font-style для установки наклона текста.
* Использование свойства font-weight для установки жирности текста.
* Использование свойства font для установки шрифта.
* Использование сервиса [Google Fonts](https://fonts.google.com/selection) для установки нестандартных шрифтов.

## Работа с шрифтами

**06:41**

* Вместо скачивания шрифта, можно указать ссылку на ресурс с ним.
* Указывается формат жирности шрифта, например, “bold”.
* Импорт шрифта в проект и указание его в качестве шрифта.

## Свойства текста

**08:38**

* text-transform позволяет указать регистр символов.
* word-spacing - пространство между словами.
* letter-spacing - пространство между символами.
* line-height - расстояние между строками.
* Применение свойств text-decoration и cursor к тексту.

**Файл index.html:**

```html

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Repellendus necessitatibus quidem quia quibusdam accusantium veritatis dolore ducimus voluptates nulla rem distinctio obcaecati tenetur, quos et corrupti magnam nemo harum adipisci.

**Файл main.css:**

```css .main\_text { color: #000; text-align: center; font-size: 1.2em; font-style: normal; font-weight: 300; font-family: ‘Sansita Swashed’, cursive; text-transform: lowercase; word-spacing: 20px; letter-spacing: 2px; line-height: 50px; text-decoration: line-through; cursor: copy; }

# Изучение CSS для новичков / Урок #7 – Стили для блоков

[Ссылка на видео](https://youtu.be/EjCTUncWviU?list=PLDyJYA6aTY1meZ3d08sRILB46OJ-wojF2)

## Создание блока с помощью CSS

**00:00**

* Создание объекта “блок” с идентификатором block.

**Файл index.html:**

```html

Simple

## Установка высоты и ширины блока

* Установка высоты и ширины блока в 200 пикселей.
* Добавление обводки к блоку с помощью свойства border.
* Установка ширины обводки в 2 пикселя, типа обводки - solid, цвета - silver.

**02:55 Добавление стилей к обводке**

* Установка различных стилей обводки: dotted, silver.
* Работа с различными свойствами обводки: border, border-bottom, border-top, border-right.
* Установка ширины обводки с помощью свойства width, style.

**07:38 Добавление прозрачности и внешних отступов**

* Добавление прозрачности к блоку с помощью свойства opacity.
* Установка значения прозрачности от 0 до 1.
* Добавление внешних отступов с помощью свойства margin.
* Установка значения внешних отступов в 20 пикселей.
* Работа с внешними отступами для каждой стороны блока.

**10:31 Добавление внешних и внутренних отступов**

* В видео объясняется, как добавить внешние и внутренние отступы к объектам в CSS.
* Внешние отступы добавляются с помощью свойства margin, а внутренние - с помощью padding.
* Если указать только одно значение для свойства, оно будет добавлено ко всем сторонам объекта.
* Если указать два значения, первое будет добавлено к верхней и нижней сторонам, а второе - к боковым.
* Если указать три значения, первое будет добавлено к верхней и нижней сторонам, второе - к боковым, а третье - к одной из боковых сторон.
* Если указать четыре значения, каждое из них будет добавлено к соответствующей стороне объекта.

**Файл main.css:**

```css #block { background: red; height: 200px; width: 300px; border: 7px dotted silver; border-bottom: 0; border-top-style: solid; border-style: double; border-radius: 15px 50% 25px 50px; opacity: 0.9; margin: 20px 50px; padding: 20px 50px; }

## Удаление внутренних и внешних отступов

**14:09**

* В видео объясняется, как удалить внутренние и внешние отступы, которые добавляются по умолчанию в браузерах.
* Для этого нужно обратиться к тэгу body, html или прописать знак звездочки, чтобы отключить все внутренние и внешние отступы.

**Файл main.css:**

```css \* { margin: 0; padding: 0; }

## Работа с блочными и строчными объектами

**15:09**

* В видео объясняется, что строчные объекты, такие как span, em, b, a, не могут иметь свойства width, height, margin, padding и другие.
* Чтобы решить эту проблему, можно использовать свойство display, чтобы указать, что объект будет блочным или строчным.
* Например, если объект является строчным, можно использовать display: inline, чтобы он вел себя как обычный строчный объект.
* Если объект является блочным, можно использовать display: block, чтобы он вел себя как блок.

**Файл index.html:**

```html

Simple

Simple

**Файл main.css:**

```css .block { background: red; height: 200px; width: 300px; border: 7px dotted silver; border-bottom: 0; border-top-style: solid; border-style: double; border-radius: 15px 50% 25px 50px; opacity: 0.9; margin: 20px 50px; padding: 20px 50px; display: block; }

# Изучение CSS для новичков / Урок #8 – Позиционирование блоков

[Ссылка на видео](https://youtu.be/cVeFNl3Jz8I?list=PLDyJYA6aTY1meZ3d08sRILB46OJ-wojF2)

## Позиционирование объектов с помощью CSS

**00:00**

* Создание нескольких объектов с идентификаторами first и second и классом block.

**Файл index.html:**

```html

* Удаление внешних и внутренних отступов для всех объектов.

**Файл main.css:**

```css \* { margin: 0; padding: 0; }

* Установка ширины и высоты для обоих объектов (200px).
* Установка margin для всех сторон (20px).
* Установка background для обоих объектов (yellow или red).

.block { width: 200px; height: 200px; margin: 20px; background: red; }

## Использование свойства “float”

**01:52**

* Обращение к объекту first и указание свойства float со значением right.
* Результат: объект first прижимается к правой стороне, другие блоки обтекают его слева.
* Добавление свойства float со значением left для объекта second.
* Результат: оба объекта находятся на одной линии, не залазят друг на друга.
* Использование значений left и right для свойства float.
* Результат: объекты могут находиться на разных сторонах, в зависимости от значения float.

**Файл main.css:**

```css #first { float: right; }

#second { float: left; background: blue; }

## Создание веб-сайта

**04:37**

* Создание шапки сайта с помощью HTML и CSS.
* Использование идентификаторов вместо классов для секций сайта (потому что по одному на странице).

**Файл index.html:**

```html

Шапка

Основная часть

Боковая панель

Футер

## Стилизация шапки

**06:37**

* Установка высоты, текста, бордера и фона.
* Использование лайфхака из CSS для идеального расположения текста по вертикали. Лайфхак состоит в том, что чтобы текст находился идеально по вертикали по центру блока, нужно чтобы параметры height (высота блока) и line-height (расстояние между строк) - были одинаковыми.

**Файл main.css:**

```css #header, #footer { width: 100%; height: 70px; text-align: center; border-bottom: 4px solid silver; background: #333; color: #fff; line-height: 70px; margin-bottom: 50px; float: left; }

## Стилизация футера

**09:07**

* Установка стилей для отступа, бордера и фона.

**Файл main.css:**

```css #footer { margin-bottom: 0; margin-top: 50px; border-bottom: 0; border-top: 4px solid silver; float: left; }

## Стилизация основной части и боковой панели

**10:00**

* Установка ширины, отступа, фона и бордера для основной части.
* Установка ширины, отступа, фона и бордера для боковой панели.

**Файл main.css:**

```css #main { width: 50%; margin-left: 10%; background: #fafafa; border: 1px solid silver; padding: 20px 2.5%; float: left; }

#aside { width: 20%; padding: 20px 2.5%; background: #fafafa; border: 1px solid silver; float: left; margin-left: 5%; }

## Позиционирование объектов

**12:00**

* Автор объясняет, что для корректного позиционирования объектов на одном уровне, они должны иметь свойство float.
* Он добавляет float: left; к тегу header и footer, чтобы объекты располагались в одну линию.
* Однако, если объекты имеют обводку, они могут не отображаться на одну линию.
* Автор предлагает удалить обводку или добавить margin-left: для корректного позиционирования.

## Позиционирование с помощью свойства “position:”

**15:00**

* Автор объясняет, что position: можно применять к любому блоку, и предлагает использовать его для блока aside.
* Он демонстрирует, как можно позиционировать объект относительно его текущего положения (position: relative;), вверх, вниз или по бокам.
* Автор также объясняет, как использовать position: для смещения объекта по левой или правой стороне.

**Файл main.css:**

```css #aside { width: 20%; padding: 20px 2.5%; background: #fafafa; border: 1px solid silver; float: left; margin-left: 5%; position: relative; bottom: -10px; left: -2px; }

## Абсолютное позиционирование с помощью свойства “position: absolute;”

**17:45**

* Автор демонстрирует, как использовать position: absolute; для абсолютного позиционирования объекта относительно самого экрана.

**Файл index.html:**

```html

+

* Он добавляет красный объект с белым текстом, центрированным по вертикали и горизонтали, и бордер радиусом 50%.
* Автор объясняет, что position: absolute; позволяет абсолютно позиционировать объект относительно размеров экрана.

**Файл main.css:**

```css #some\_block { background: red; color: #fff; text-align: center; width: 50px; height: 50px; line-height: 50px; border-radius: 50%; position: absolute; left: 20px; top: 50px; }

## Фиксация объекта с помощью свойства “fixed”

**19:41**

* Автор объясняет, что свойство fixed позволяет зафиксировать объект на определенной позиции.
* Он демонстрирует, как использовать fixed для создания зафиксированного блока на странице. При скролле страницы (т.к. высота у нее 2000 пикселей), блок остается на месте.
* Автор подчеркивает, что fixed является простым и эффективным способом создания зафиксированных блоков на веб-страницах.

**Файл main.css:**

```css body { height: 2000px; }

#some\_block { background: red; color: #fff; text-align: center; width: 50px; height: 50px; line-height: 50px; border-radius: 50%; position: fixed; left: 20px; top: 50px; }

# Изучение CSS для новичков / Урок #9 – Работа со списками

[Ссылка на видео](https://youtu.be/zy4yzUjpnvA?list=PLDyJYA6aTY1meZ3d08sRILB46OJ-wojF2)

## Работа со списками через CSS

**00:00**

* В этом уроке мы научимся работать со списками через CSS.
* Сначала рассмотрим свойство z-index:, которое позволяет указать, какой объект будет более главным в иерархии слоев.

**Файл main.css:**

```css #main { width: 50%; margin-left: 10%; background: #fafafa; border: 1px solid silver; padding: 20px 2.5%; float: left; position: relative; z-index: 1; }

#aside { width: 20%; padding: 20px 2.5%; background: #fafafa; border: 1px solid silver; float: left; margin-left: 5%; position: relative; bottom: -10px; z-index: 0; left: -200px; }

* Затем мы создадим новый список ul и добавим к нему айди nav (navigation).

**Файл index.html:**

```html

Шапка

Главная

О Нас

Контакты

Основная часть

* Укажем свойство “list-style:” со значением “none;”, чтобы убрать все стили из списка.
* Затем укажем, что все объекты будут блочными или строчными.
* Далее мы работаем с навигацией, указывая, что это будет блочный объект с отступом слева и шириной 50%.
* Затем мы указываем стили для каждого элемента списка (#nav li), включая бэкграунд, бордер, пединг и мерджин.
* В конце мы добавляем стили для наведения, включая изменение цвета текста, бордер колор и бордер радиус.

**03:35 Создание менюшки**

* Мы создаем менюшку, используя список и добавляя стили.
* Сначала мы указываем, что это будет дисплей блок, флот лефт и ширина 50%.
* Затем мы указываем текст элайн с центром и нижний отступ.
* Для каждого элемента списка мы указываем стили, включая бэкграунд, бордер, пединг, мерджин и мерджин райт.
* В конце мы добавляем стили для наведения, включая изменение цвета текста и бордер колор.
* В итоге, мы создали полноценную менюшку, которая выглядит довольно здорово.

**Файл main.css:**

```css #nav { list-style: none; display: block; float: left; width: 50%; nav-left: 25%; text-align: center; margin-bottom: 50px; }

#nav li { display: inline; background: #fafafa; border: 1px solid silver; padding: 0 20px; cursor: pointer; }

#nav li:hover { color: #fff; background: #333; border-radius: 5px; border-color: #333; }

#nav li:not(:last-of-type) { margin-right: 20px; }