



ВВЕДЕНИЕ В ВЕРСТКУ



АЛЕКСАНДР БЕСПОЯСОВ



АЛЕКСАНДР БЕСПОЯСОВ

Front-end разработчик в «Медиа-Шторм»



bespoyasov@me.com



[webtwozero](https://github.com/webtwozero)



[facebook](https://facebook.com)

ПЛАН ЗАНЯТИЯ

1. Введение в HTML

- Структура страницы
- !DOCTYPE html
- html
- head
- title
- meta
- link
- body

2. Введение в CSS

- Селектор
- font-family
- font-style
- font-weight
- font-size
- color
- text-decoration



ЧТО ЭТО ВООБЩЕ ТАКОЕ?

Принято считать, что верстка сайтов - это процесс создания веб-страницы на основе дизайна.

Проще говоря, дизайнер отдает макеты с дизайном, а верстальщик перегоняет его в веб-страницы. Вроде все просто. И не должно быть никаких проблем. Но...



ВВЕДЕНИЕ В HTML

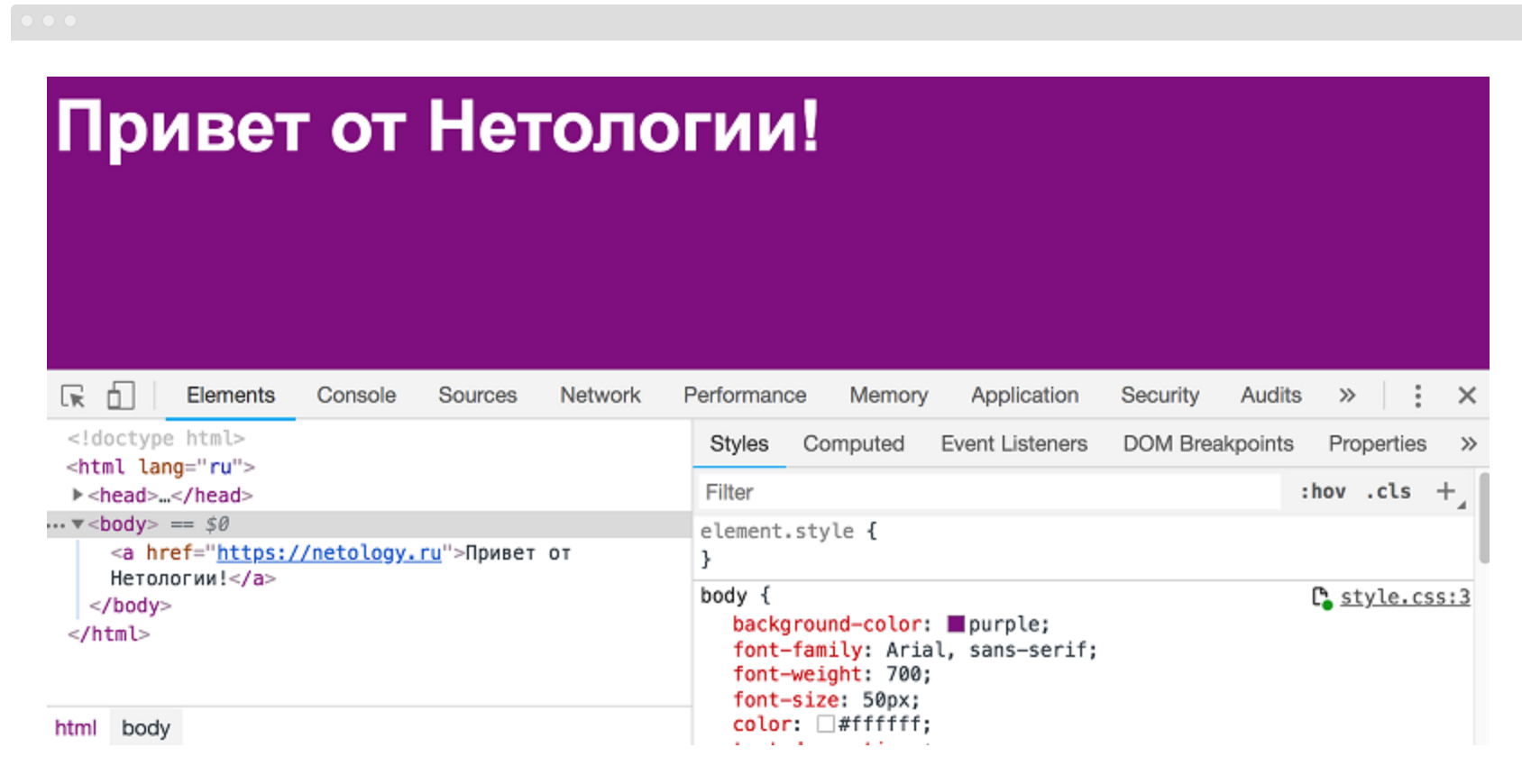
РАССМОТРИМ ПРИМЕР



Привет от Нетологии!

Один из разработчиков Нетологии сделал нам страницу с приветствием.

DEVTOOLS



Посмотрим на страницу используя инспектор. Для этого нужно выбрать «Проинспектировать элемент» из меню после щелчка правой кнопкой мыши по странице или нажмем F12 (⌘ + option + J для mac).

HTML И ТЕГИ



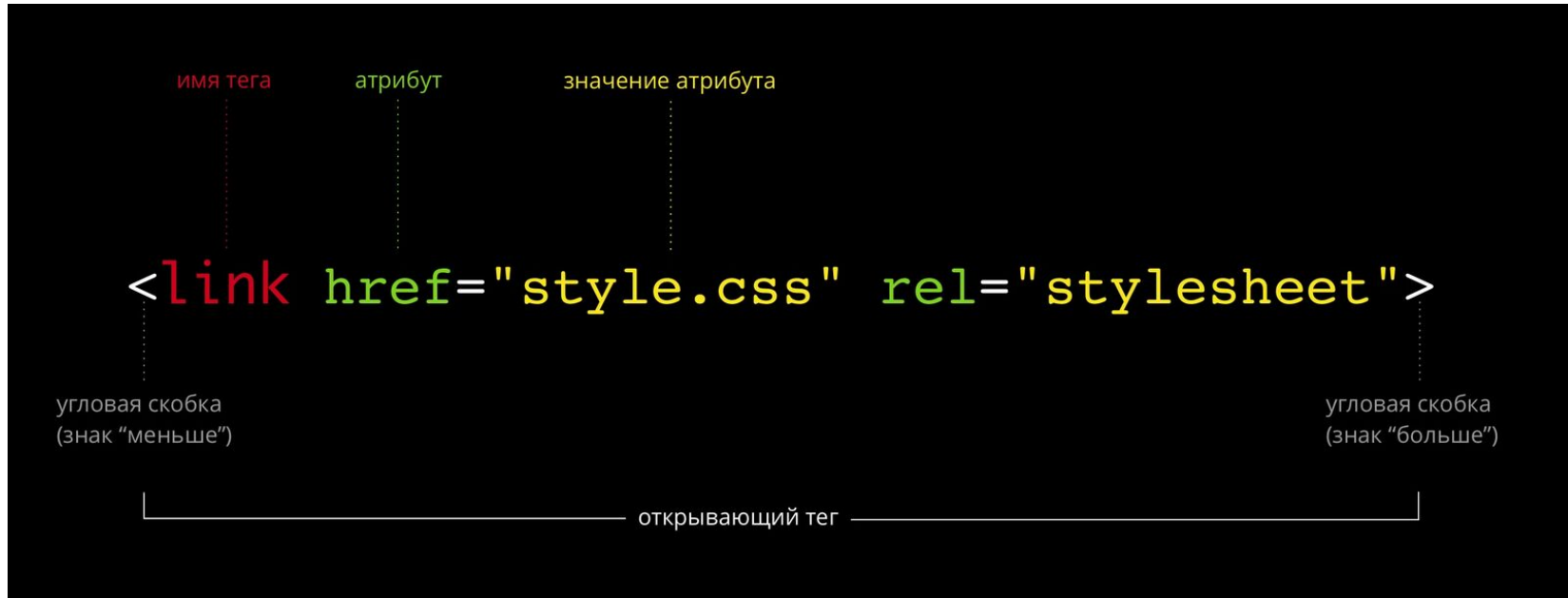
HTML — это язык описания структуры веб-страницы.

Если провести аналогию с человеком, то используя HTML, разработчики создают скелет веб-страницы. Делают они это с помощью специальных слов, которые понимает браузер. Эти слова принято называть **тегами**.



*HTML **теги** — это специальные слова, позволяющие браузеру отобразить страницу.*

НЕПАРНЫЕ ТЕГИ



Абсолютно любой тег начинается с символа `<` (меньше). Далее указывается название тега. Например, `link`. А после него добавляется символ `>` (больше). В итоге получается слово — `<link>`.

АТТРИБУТЫ

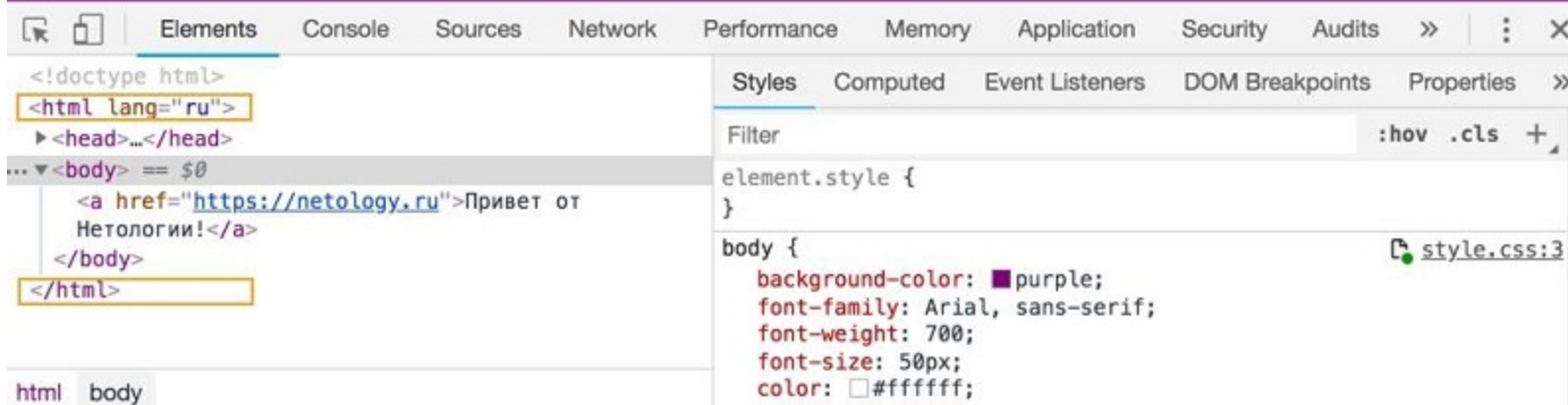


```
href="style.css"
```

После названия тега ставится пробел и указывается название атрибута, а его значение указывается в кавычках. Между названием и значением добавляется символ `=`.

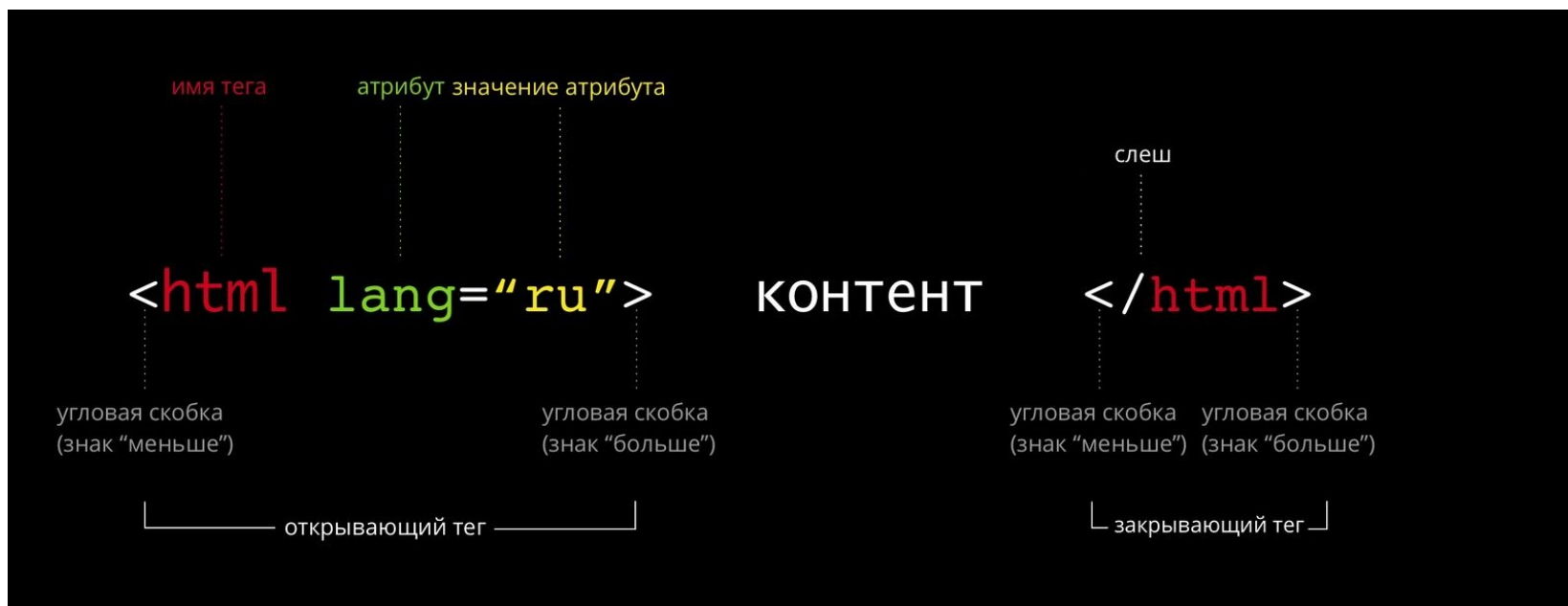
ПАРНЫЕ ТЕГИ

Привет от Нетологии!



КОНСТРУКЦИЯ ПАРНОГО ТЕГА

Главное отличие парного тега от непарного заключается в том, что в парный тег мы можем добавить контент, т.е. еще элементы или текст.



Для создания закрывающего тега нужно написать открывающий тег, но после символа `<` добавить обратный слеш `/`.



СТРУКТУРА СТРАНИЦЫ

НАПИШЕМ СЛЕДУЮЩИЙ КОД:

```
1  <!DOCTYPE html>
2  <html lang="ru">
3  <head>
4      <title>Привет от Нетологии!</title>
5      <meta charset="utf-8">
6      <link href="css/style.css" rel="stylesheet">
7  </head>
8
9  <body>
10     Привет от Нетологии!
11 </body>
12
13 </html>
```



РЕЗУЛЬТАТ



Привет от Нетологии!

ТЕГ `<!DOCTYPE html>`

Тег `<!DOCTYPE html>` показывает браузеру, по каким правилам ему нужно отобразить HTML страницу.

Если тег `doctype` отсутствует, браузеры будут использовать устаревшие правила отображения и при этом многие современные возможности работать не будут.

Поэтому всегда нужно добавлять этот тег в начало HTML страницы, и такая запись: `<!DOCTYPE html>` сообщит браузеру, что на странице нужна поддержка всех, в том числе и новых, заявленных в стандарте возможностей.

ТЕГ `<html>`

Тег `<html>` является корневым элементом страницы.

Свое название он получил, потому что именно с него начинается создание самой страницы, и соответственно все остальные элементы будут вкладываться в него. Внутри элемента всегда находятся два основных элемента - `<head>` и `<body>`.

Также у тега `html` есть обязательный атрибут `lang`. С помощью этого атрибута браузер может понять, на каком языке написан текст на странице. Это необходимо для корректного отображения символов языка. В нашем примере атрибут имеет значение `«ru»`, что означает русский язык.

ТЕГ `<head>`

Тег `head` — это невидимый для пользователей раздел, в котором хранятся элементы, целью которых является указание информации о странице.

В теге `head` обязательно должен находиться тег `title`.

ТЕГ `<title>`

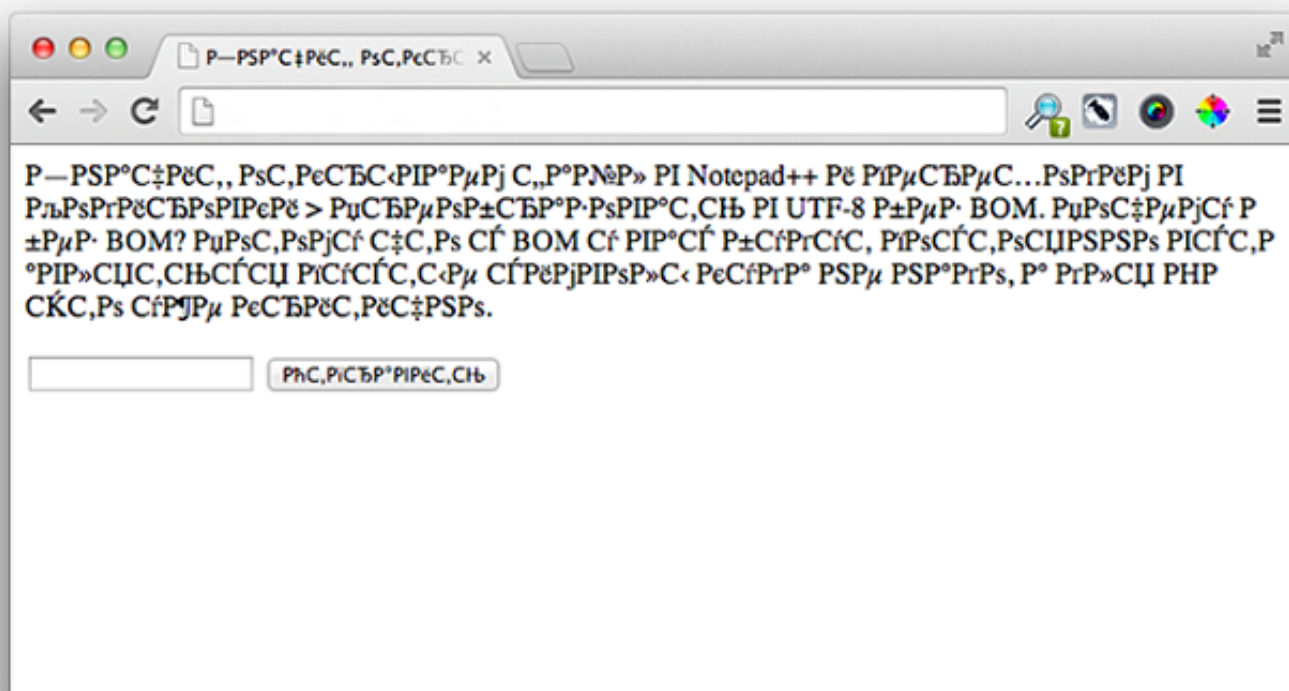
Определяет заголовок документа, который отображается во вкладке браузера. Допускается использовать только один тег `title`.



ТЕГ <meta>

В нашем примере, используя `meta` тег, мы указали кодировку страницы. Для этого мы использовали атрибут `charset` со значением `utf-8`.

Когда браузер не понимает символ, то он выводит что-то вроде этого:



Тег `<link>`

Тег `link` помогает браузеру загрузить внешний ресурс на страницу. Элемент имеет обязательный атрибут `href`, в котором указывается адрес откуда браузеру нужно загрузить файл.

```
<link href="css/style.css" rel="stylesheet">
```

Атрибут `rel` нужен, чтобы сообщить браузеру, какой тип файла мы хотим подключить. Так как нам нужен CSS-файл, мы записали `stylesheet` в качестве значения этого атрибута.

АБСОЛЮТНЫЕ И ОТНОСИТЕЛЬНЫЕ ПУТИ

Существует два основных типа адреса - это абсолютный и относительный. Абсолютные адреса должны начинаться с указания протокола (обычно http://) и содержать имя сайта. Относительные ссылки ведут отсчет от текущего документа.

абсолютный путь:

`https://yastatic.net/s3/home/logos/citylogos/bak2018/ru.png`

относительный путь:

`/logos/citylogos/bak2018/ru.png`

ТЕГ `<body>`

Мы рассмотрели основные теги, встречающиеся в теге `head`, а теперь перейдем к тегу `body`.

Тег `body` содержит основной контент HTML документа или, иными словами — видимую для пользователя секцию HTML документа.



ВВЕДЕНИЕ В CSS



РАЗМЕТКА БЕЗ ПОДКЛЮЧЕНИЯ СТИЛЕЙ

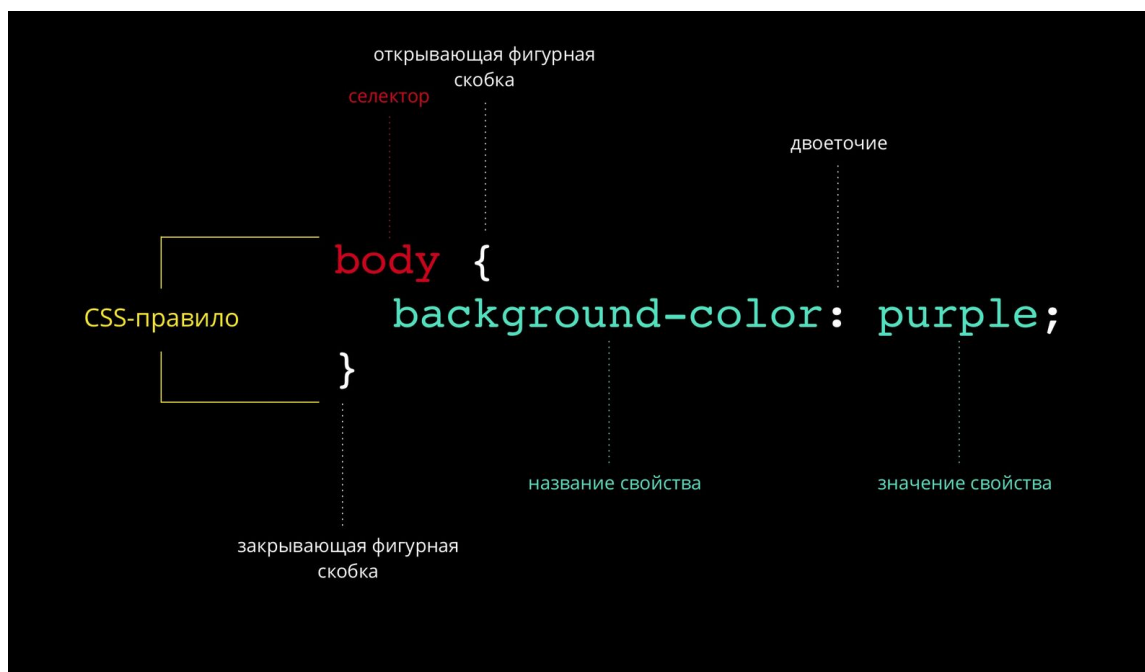


Привет от Нетологии!

CASCADING STYLE SHEETS



CSS (Cascading Style Sheets, каскадные таблицы стилей) — это язык для описания внешнего вида веб-страницы.



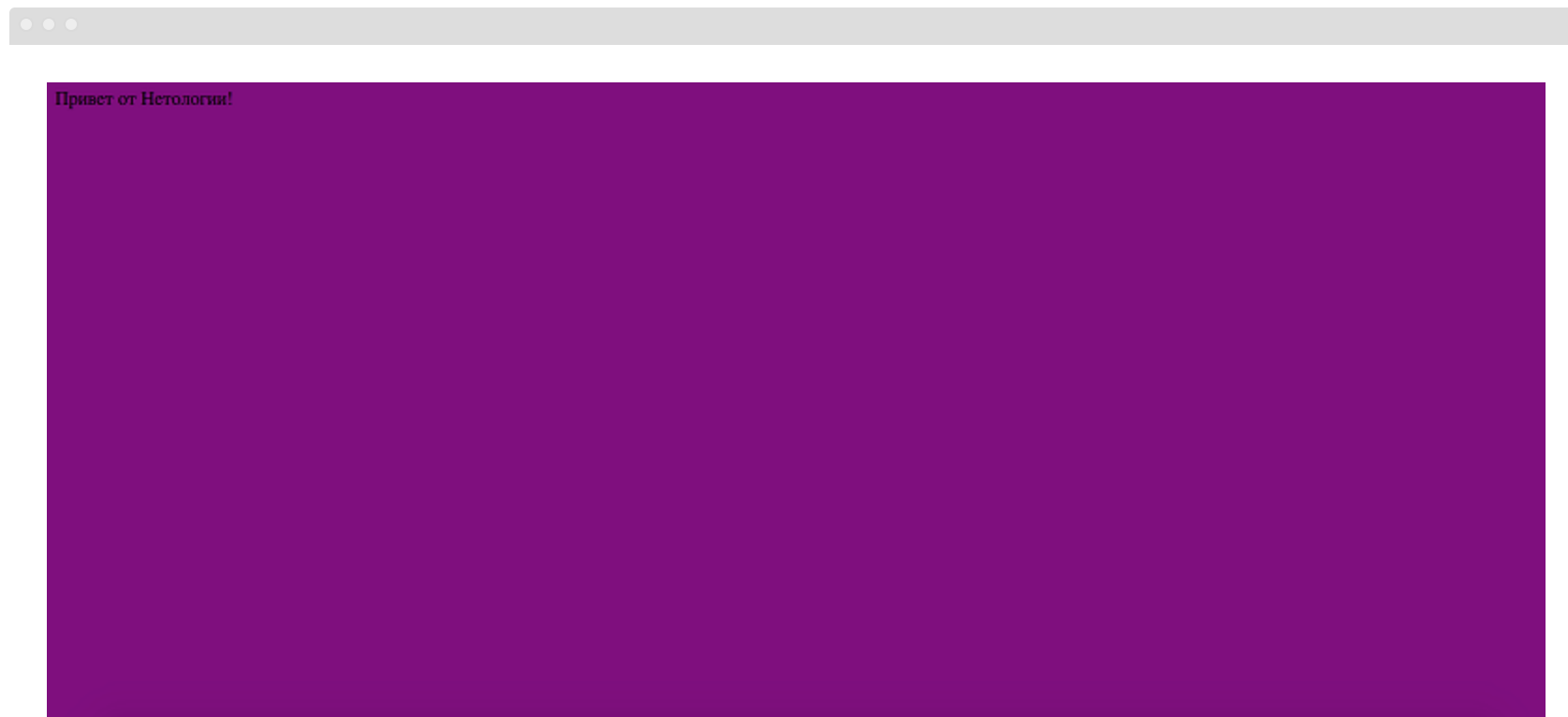
СЕЛЕКТОР

Селектор - это выражение, с помощью которого браузер находит HTML-элементы и применяет к ним CSS.

В нашем примере мы использовали селектор по имени тега. Когда браузер прочитал селектор `body`, то он будет искать все элементы с таким названием. Например, в следующем примере браузер изменит цвет фона для элемента `body`.

```
1 | body {  
2 |     background-color: purple;  
3 | }
```

РЕЗУЛЬТАТ



И действительно, фон страницы стал фиолетовым!

CSS СВОЙСТВА ДЛЯ СТИЛИЗАЦИИ ТЕКСТА

Текст на странице должен выглядеть следующим образом:



Привет от Нетологии!

В CSS существует свойства с помощью, которых мы можем стилизовать текста, и соответственно решить нашу задачу.

font-family

Свойство с помощью, которого можно указать название шрифта. Для этого нужно написать его названия в качестве значения свойства.

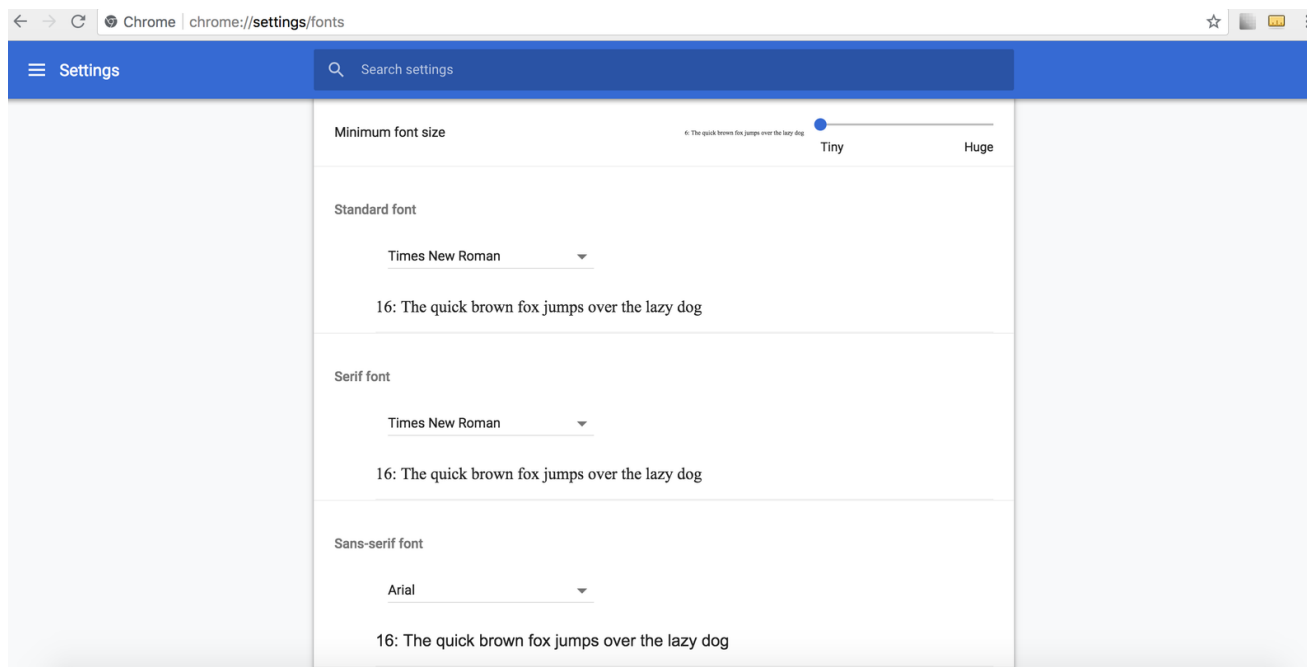
```
1 body {  
2   font-family: Arial;  
3 }
```

Свойство `font-family` позволяет указывать несколько шрифтов через запятую:

```
1 body {  
2   font-family: Arial, Tahoma;  
3 }
```

font-family

И когда первый шрифт не может быть загружен, то браузер пытается отобразить следующим. Например, Tahoma. А если и его нет, то отображает шрифт, который установлен в настройках браузера. В большинстве случаев это Times New Roman, но может быть и другой.



СЕМЕЙСТВА ШРИФТОВ

- **serif** — шрифты с засечками (например, Times_);
- **sans-serif** — рубленые шрифты (например Arial);
- **cursive** — курсивные шрифты;
- **fantasy** — декоративные шрифты;
- **monospace** — моноширинные шрифты(например Courier).

```
1  body {  
2    font-family: Arial, sans-serif;  
3  }
```

В этом примере кода, если по какой-то причине шрифт Arial не загрузится, то браузер запросит альтернативный шрифт, который также относится к семейству sans-serif.

font-style

Свойство определяет начертание шрифта, а именно обычное, курсивное или наклонное. Для этого существуют следующие значения:

- **normal** — обычное начертание текста;
- **italic** — курсивное начертание — это специальный шрифт имитирующий рукописный;
- **oblique** — наклонное начертание — образуется путем наклона обычных знаков вправо.

font-weight

Это CSS свойство позволяет управлять толщиной шрифта. Существует несколько способов задать значение свойства. Первый использовать следующие значения:

- **100** — тонкое начертание;
- **200** — сверхсветлое;
- **300** — светлое;
- **400** — нормальное;
- **500** — среднее;
- **600** — полужирное;
- **700** — жирное;
- **800** — сверхжирное;
- **900** — тяжёлое.

font-weight

Второй, это использовать ключевые слова:

- **normal** — нормальное начертание — то же, что и **400**;
- **bold** — полужирное начертание — то же, что и **700**;
- **lighter** — изменяет начертание относительно насыщенности родительского элемента (сверхтонкое начертание);
- **bolder** — изменяет начертание относительно насыщенности родителя элемента (сверхжирное начертание).

font-size

Свойство позволяет задать размер шрифта. Есть несколько допустимых единиц для размера шрифта, например, можно указать его в пикселях.

Пиксель — это специальная единица, придуманная для CSS. Под пикселем понимается одна визуально различимая точка экрана. Обозначается он как `px`. Например, чтобы задать размер шрифта в 50 пикселей, нужно записать:

```
font-size: 50px;
```



color

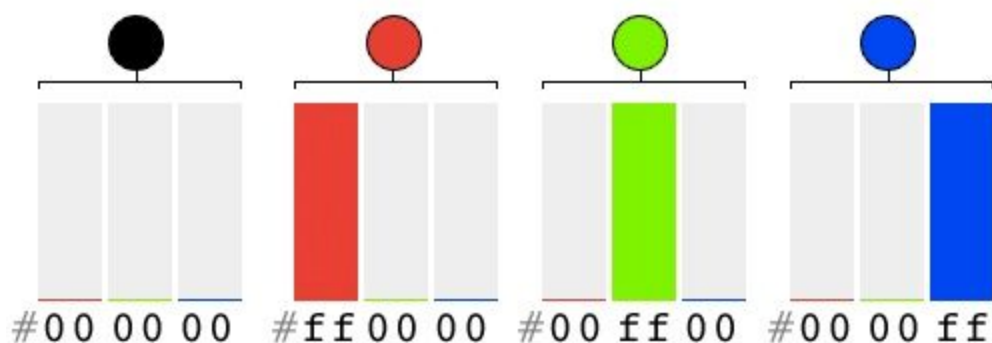
Свойство позволяет установить цвет текста. Значение для свойства можно задать несколькими способами. Первый, это использовать ключевые слова:

- **white**
- **gray**
- **black**
- **и т.д.**

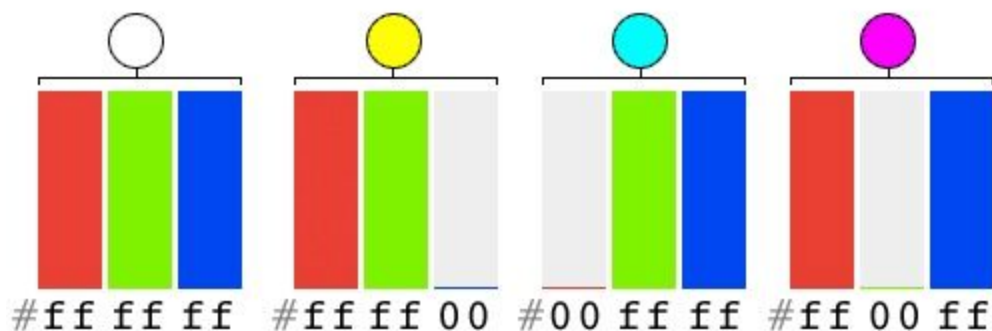
Этот способ используется очень редко.

HEX

Чаще всего используют шестнадцатеричный формат и он выглядит следующим образом:



Нулевые значения по какому-то из трех каналов означают отсутствие цвета



Наибольшие значения по какому-то из трех каналов означают максимум яркости этого цвета

ПИШЕМ КОД

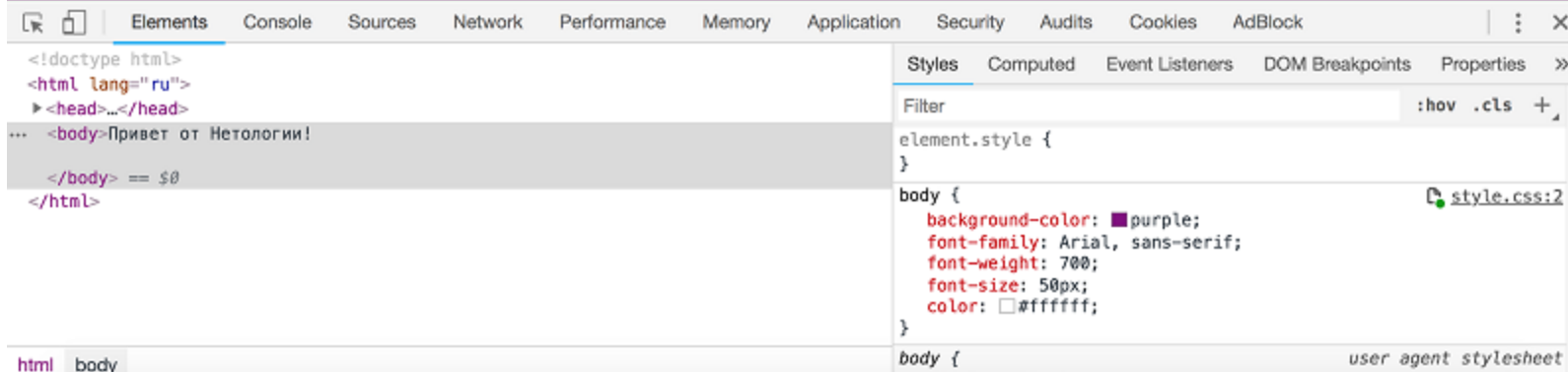
Мы рассмотрели все необходимые свойства для решения нашей задачи и теперь собственное решим ее. Нам нужно подключить шрифт `Arial`, сделать текст жирным, размером `50px` и белого цвета.

```
1  body {  
2      font-family: Arial, sans-serif;  
3      font-weight: 700;  
4      font-size: 50px;  
5      color: #ffffff;  
6  }
```

РЕЗУЛЬТАТ

Верстка выглядит так, как нам нужно! Но полагаться на свой глаз не стоит. Проверим в инспекторе, применились ли стили. Для этого существует вкладка «Styles»

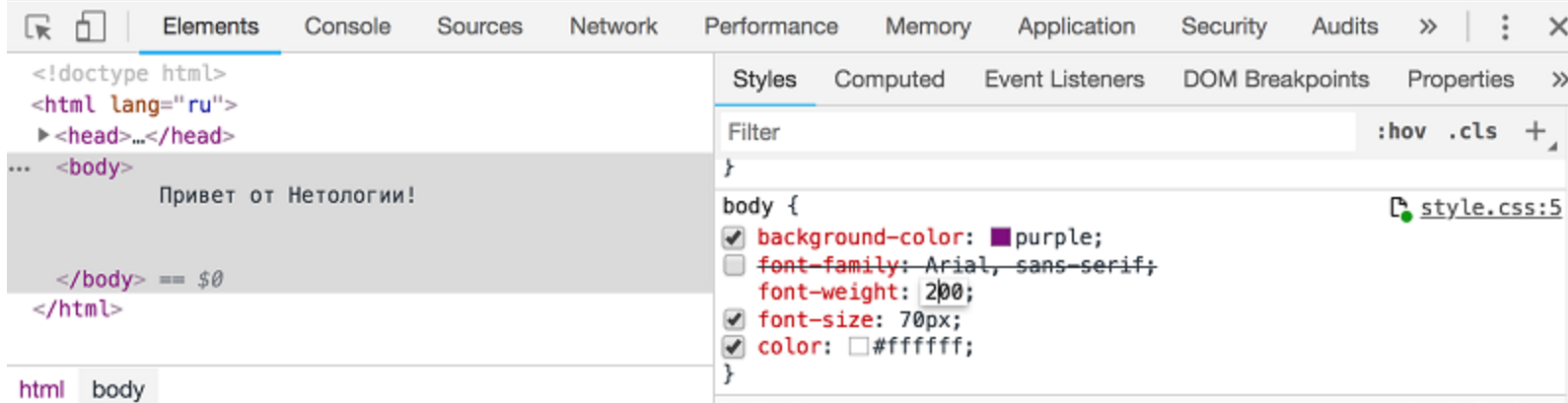
Привет от Нетологии!



РЕЗУЛЬТАТ

Также в инспекторе мы можем добавлять свойства к элементу или отключить их.

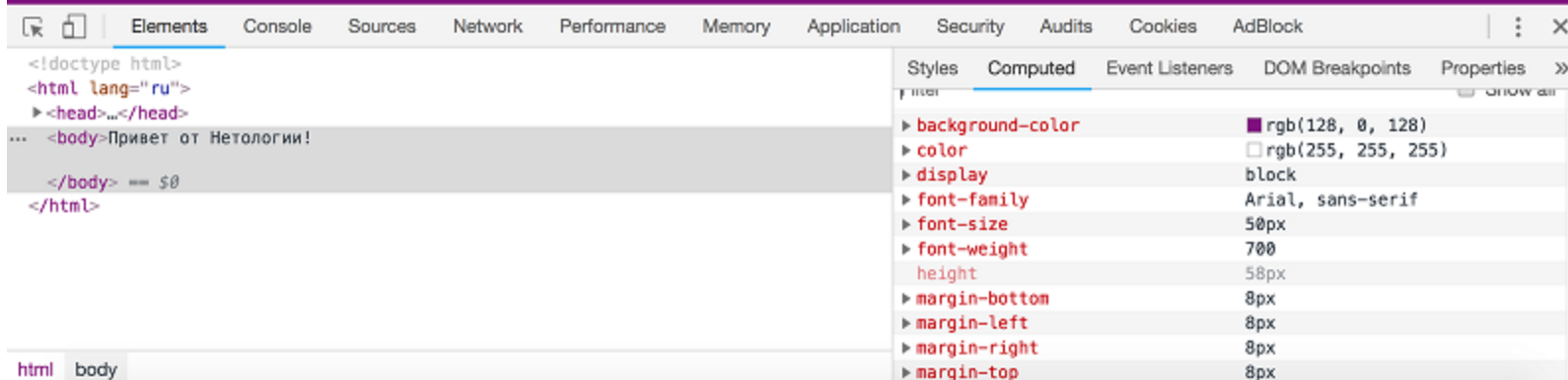
Привет от Нетологии!



COMPUTED

Если мы выберем вкладку Computed, мы сможем увидеть не только стили, которые написали мы, а вообще все примененные к элементу стили:

Привет от Нетологии!



ТЕГ `<a>`

Элемент `a` предназначен для создания ссылок.

Если мы хотим, чтобы при клике пользователя по элементу происходил переход на другую страницу, то обязательно нужно добавить атрибут `href`. И указать адрес страницы куда осуществится переход в качестве значения атрибута.

```
<a href="https://netology.ru">Перейти на сайт Нетологии</a>
```

Можно создать переход и внутри страницы. Для этого нужно в атрибуте `href` написать `#`, а после символа написать значение атрибута `id` у элемента, к которому нужно будет перейти.

```
<a href="#contacts">перейти к блоку с контактами</a>  
<div id="contacts">контакты</div>
```

ПИШЕМ КОД

Мы рассмотрели все что нужно, чтобы сделать текст ссылкой.

```
1 <body>
2   <a href="https://netology.ru">Привет от Нетологии!</a>
3 </body>
```



Привет от Нетологии!

text-decoration

Свойство позволяет декорировать текст, добавляя к нему различные и для него возможны следующие значения:

- **none** — не подчеркнутый текст;
- **underline** — добавляет нижнее подчеркивание;
- **line-through** — добавляет зачеркивание;
- **overline** — добавляет надчеркивание;

I wandered lonely as a cloud

underline

I wandered lonely as a cloud

overline

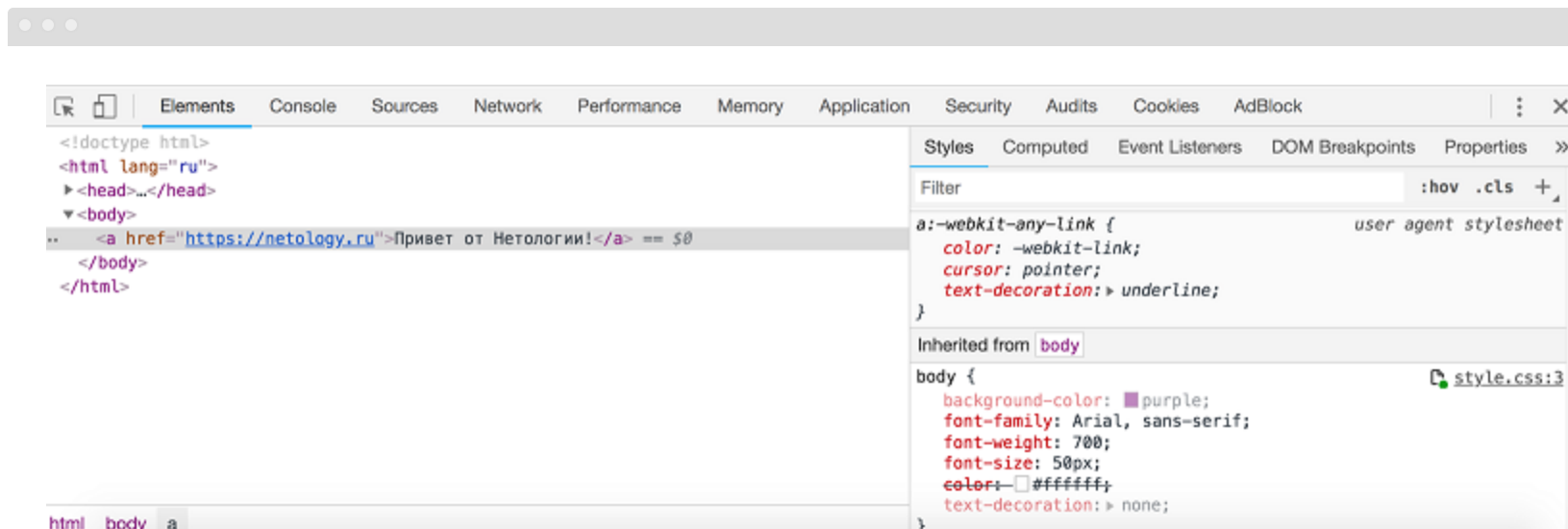
I wandered lonely as a cloud

line-through

УБИРАЕМ ПОДЧЕРКИВАНИЕ

И теперь осталось задать белый цвет для текста и убрать подчеркивание.

```
1 body {  
2   color: #ffffff;  
3   text-decoration: none;  
4 }
```



ЗАДАЕМ СВОЙСТВА ДЛЯ ССЫЛКИ

Мы видим, что свойство `color` зачеркнуто, а свойство `text-decoration` выделено блекло. Таким способом инспектор показывает, что эти свойства не применились к элементу. Это случилось, потому что в браузере уже есть стили для ссылки.

Поэтому нам нужно задать свойства `color` и `text-decoration` для элемента `a`.

```
1  a {  
2    color: #ffffff;  
3    text-decoration: none;  
4  }
```



ИТОГИ

HTML

- Тег `html` является корневым элементом страницы. У тега есть обязательный атрибут `lang`;
- Тег `title` определяет заголовок документа, который отображается во вкладке браузера;
- Элемент `meta` определяет данные, которые используются для хранения информации, предназначенной для браузеров и поисковых систем;
- Тег `link` помогает браузеру загрузить внешний ресурс на страницу. Элемент имеет обязательный атрибут `href`, в котором указывается адрес откуда браузеру нужно загрузить файл;
- Тег `body` содержит основной контент HTML документа.
- Тег `a` предназначен для создания ссылок.

CSS

- Селектор - это выражение, с помощью которого браузер находит HTML-элементы и применяет к ним CSS свойства;
- Свойство `font-family` указывает название шрифта;
- Свойство `font-style` определяет начертание шрифта;
- свойство `font-weight` позволяет управлять толщиной шрифта;
- Свойство `font-size` позволяет задать размер шрифта;
- Свойство `color` позволяет установить цвет текста;
- Свойство `text-decoration` позволяет декорировать текст.



Задавайте вопросы и напишите отзыв о лекции!

АЛЕКСАНДР БЕСПОЯСОВ

 bespoyasov@me.com

 [webtwozero](https://www.soundcloud.com/webtwozero)

 [facebook](https://www.facebook.com/webtwozero)