# Оформление страниц в HTML

Создание Web-страниц и сайтов требуют знания нескольких языков. В первую очередь, это язык гипертекстовой разметки – HTML, [HyperText Markup Language](https://ru.wikipedia.org/wiki/HTML). Как у любого языка, у HTML есть *грамматика и словарь*. Слова этого языка записываются в **<угловых>** скобках и называются элементами, или тегами.

Теги используются для указания начала и конца элемента разметки, то есть, обычно **теги используются парами**. К конечному элементу пары тегов добавляют «/». Так отмечают начало и конец параграфа, или абзаца: **<p>**Текст параграфа**</p>**. Парный тег – это **контейнер**. Если начало и конец элемента совпадают, используют **одиночные теги**. Примеры одиночных тегов: <br> (принудительный разрыв строки), <img> (вставка картинки).

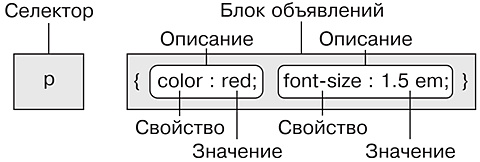
Подробнее о языке HTML можно прочитать на странице <http://gvard.github.io/web/html/>

Но HTML создан для разметки и предоставляет только самые базовые возможности по оформлению материалов.

Например, при помощи тега **<b>** можно сделать начертание шрифта полужирным, а при помощи тега **<i>** – курсивным.

# CSS: Каскадные таблицы стилей

В современной версии стандарта HTML5 практически все оформление страницы делается при помощи **каскадных листов стилей**, [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). Команда CSS состоит из селектора и соответствующих ему свойств:



* Все относящиеся к селектору свойства записываются в фигурных скобках
* Описания свойств разделяются точкой с запятой. **Не поставил «;» между свойствами – получил ошибку**
* Свойства и их значения разделяются двоеточием
* Вокруг селекторов, свойств, их значений можно ставить любое количество пробелов, знаков табуляции и переносов строки.

## Способы использования CSS в HTML

1. внешние стили, подключаются тегом <link> в голове документа, т.е. **внутри** тега <head>:

**<link rel="stylesheet" href="style.css">**

1. внутренние стили, внутри тега **<style>** (a <style> – внутри тега <head> документа html)
2. встроенные стили, как значение атрибута style:

**<p style="display:none">**

# Наследование

У элементов есть связи внутри страницы.

* Тег, внутри которого находятся другие теги, для них **предок**, а они – **потомки**.
* Тег, который находится непосредственно в предке, без промежуточных елементов – **дочерний элемент**, а предок для него – **родитель**.

Свойства предков могут наследоваться потомками.

# Разбор страницы

Разберем все вышенаписанное на примерах. У нас есть статья с заголовком, параграфами и картинкой:



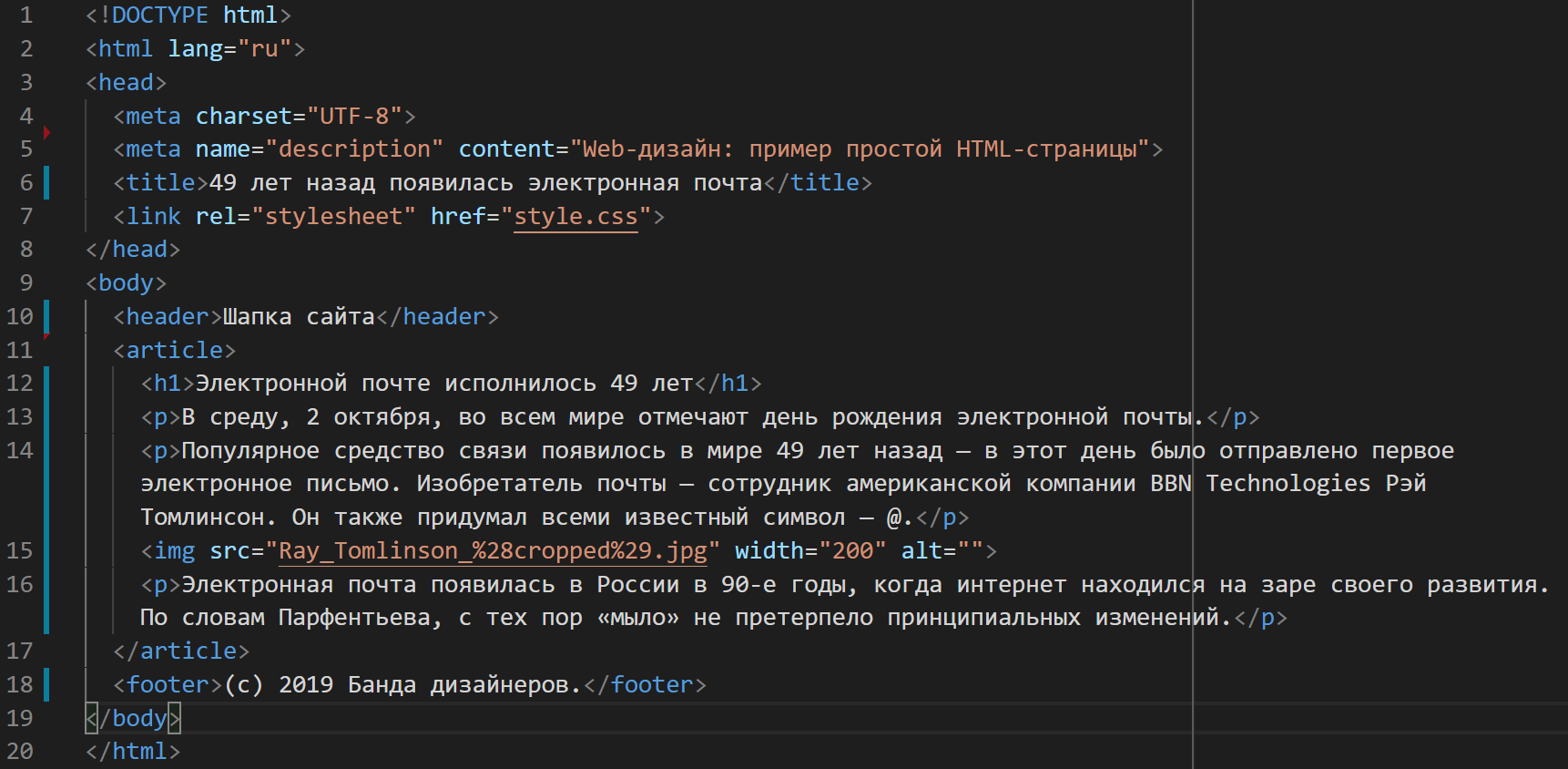
Оформление этой страницы минималистично:

1. Заголовок – полужирным начертанием и увеличенным размером шрифта (кеглем)
2. Вокруг заголовка, картинки и параграфов установлены поля (пустое пространство вокруг этих элементов)

Мы хотим, чтобы:

1. контент (содержимое страницы) не «прилипал» к краям окна браузера,
2. вокруг статьи был установлен цвет фона, отличный от фона статьи.

Вот упрощенный код HTML:



Вложенность тегов заметна благодаря **отступам в два пробела**.

Для компактности мы убрали отступ у тегов **<html>.**

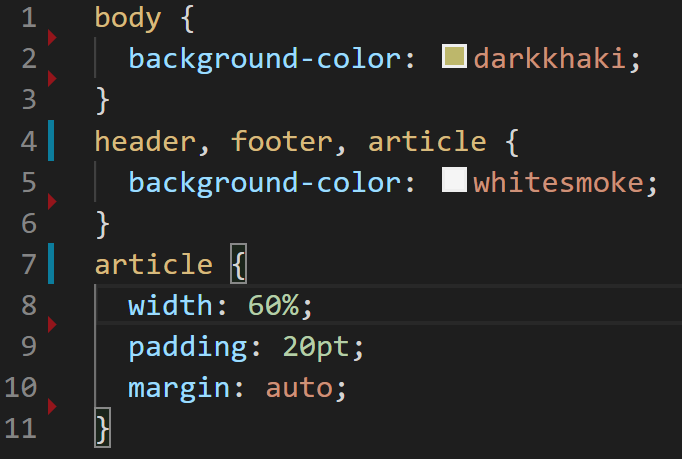
Теги **<meta>, <title>, <link>** – потомки тега **<head>**. Это – служебные теги.

**<header>, <article>, <footer>** – потомки тега **<body>.** Это теги, в которых находится содержание страницы, контент.

# CSS: Цвет и фон

Мы воспользовались первым способом подключения каскадных листов стилей и в 7-й строке написали тег **<link>**.

Наши правила CSS будут находиться в файле **style.css**:



Селектор **body** определяет правила для всего содержимого страницы. То есть, фон цвета «темный хаки» будет применен ко всей странице. Так работает **наследование**.

Но следующей строкой мы переопределяем свойство **background-color** у трех элементов: **header, footer, article**. Их фон теперь – светло-серый.

Вопрос: у какого элемента остался фон цвета хаки?

Ответ: ни у какого! Больше потомков у body нет.

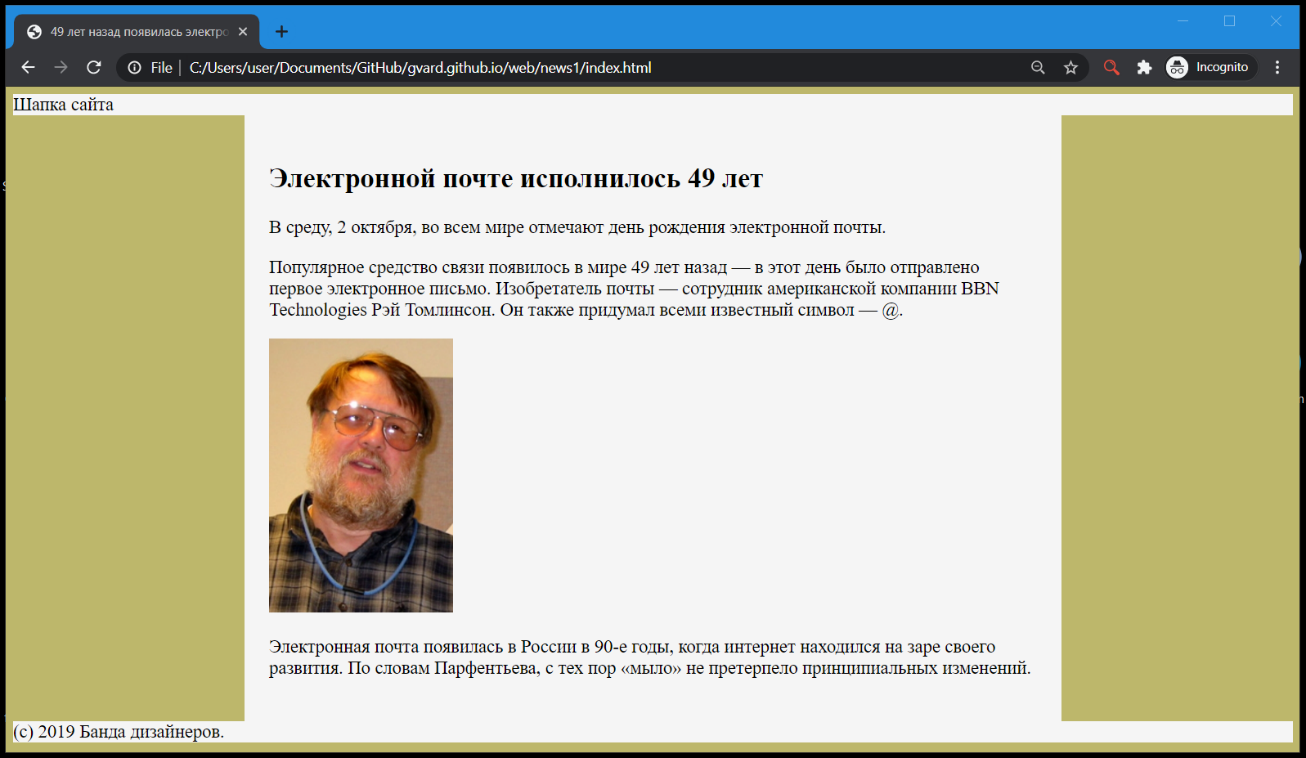
Тогда зачем мы так сделали?

Ответ – в следующей строке. Для элемента article мы определяем ширину – 60% от ширины окна,

Внутренние поля – 20 пунктов,

Внешние поля будут установлены автоматически. То есть, станут равными слева и справа.

Результат:



Так вот зачем мы установили фон всей страницы!

Такие поля вокруг содержания страницы часто называют **паспарту**. Ударение – на последний слог, ведь слово родом из французского языка.

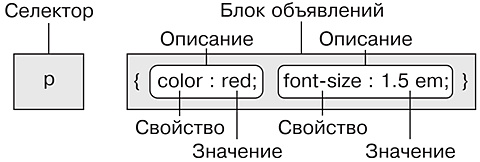
Замечательно видна разница между внешними и внутренними полями. Внутренние – цвета фона самого элемента, внешние – цвета фона предка.

Учебная страница доступна по адресу <http://gvard.github.io/web/news1/>

Мы еще к ней вернемся!

С внутренними и внешними полями будем работать позже, а пока запомним несколько правил по цвету и фону.

Вернемся к первой иллюстрации:



**body {color: red}** установит цвет текста – красный **у всей страницы**.

**p {color: red}** установит цвет текста – красный **у всех параграфов**.

**body {background-color: green}** установит цвет фона **всей страницы** в зеленый.

Если мы хотим поменять цвет фона у параграфов, мы напишем:

**p {background-color: green}**

Чтобы установить на фон всей страницы картинку, напишем:

**body {background-image: url(“bg.jpg”)}**

**p { background-image: url(“bg.jpg”)}** установит фон только параграфам.

Обратите внимание, в примерах выше фон устанавливается всем одноименным элементам. Если параграфы, то все сразу!

Чтобы обращаться к элементам индивидуально, существует множество способов. Все они описаны на странице <http://gvard.github.io/web/css/>

Мы их обязательно разберем позже.

# Шрифт, текст

Текст – это любые печатные символы на странице. Шрифт – это графический рисунок начертаний букв и знаков. **Абсолютно любой текст на веб-странице написан определенным шрифтом, даже если мы его не задавали в стилях!**

На картинке выше было также указано правило, устанавливающее размер шрифта.

**body {font-size: 1.5em}** установит размер шрифта в 1.5 em **всему тексту на странице**. em – это относительная единица измерения, равная текущему указанному размеру шрифта.

Подробнее о шрифтах, в частности, о единицах измерения – на странице <http://gvard.github.io/web/fonts/>

# Шрифты: основные характеристики

* **начертание**: прямой, *курсивный* (свойство **font-style**) Пример: **body {font-** **style: italic}**
* **насыщенность**: светлый, **полужирный**, **жирный** (свойство **font-weight**) Пример: **body {font-** **style: bold}**
* **ширина**: нормальный, узкий, широкий (свойство **font-stretch**), шрифт фиксированной ширины (моноширинный)
* **размер** (кегль) – высота литер шрифта (свойства **font-size**)
* **наличие (serif) или отсутствие (sans-serif)**[**засечек**](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D0%B5%D1%87%D0%BA%D0%B8) на концах линий. Это группы шрифтов [Антиква](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D1%82%D0%B8%D0%BA%D0%B2%D0%B0), с засечками и [Гротеск](https://en.wikipedia.org/wiki/Sans-serif), без засечек:



Позже мы будем подключать внешние шрифты к веб-страницам, а пока посмотрим на семейства шрифтов:

# Семейства шрифтов: свойство font-family

Шрифты с засечками, или антиква (**font-family: se**rif):

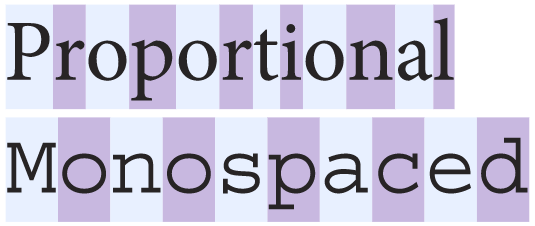
* Times Georgia (шрифт Georgia)

Шрифты без засечек, рубленые шрифты, или гротески (**font-family: sans-serif**)

* Arial, Helvetica

Моноширинные шрифты, когда все знаки имеют одинаковую ширину (**font-family: monospace**)

* Courier New, Courier



# Самостоятельная работа

* Цель каждого кружковца – работа над собственными проектами. Проект – это веб-страница или веб-сайт, то есть, набор веб-страниц.
* Задача на сегодня – научиться применять каскадные листы стилей. Что следует сделать:

1. Создать веб-страницу. Можно скопировать код HTML из примера выше.
2. Подключить к странице каскадные листы стилей (CSS). Можно любым из трех способов, описанных выше.
3. Установить при помощи CSS цвет и фон любых элементов. Можно – всей страницы.
4. По желанию: установить размер шрифта, начертание шрифта, семейство шрифта или любое другое свойство шрифта.