## Структура HTML-документа

|  |
| --- |
| [С чего начинается HTML](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/1) |
| [Простейшая HTML-страница](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/2) |
| [Заголовок HTML-страницы](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/3) |
| [Кодировка HTML-страницы](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/4) |
| [Ключевые слова](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/5) |
| [Описание содержания страницы](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/6) |
| [HTML-комментарии](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/7) |
| [Подключение стилей](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/8) |
| [Тайна CSS-редактора](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/9) |
| [Подключение внешних стилей](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/10) |
| [Подключение скриптов](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/11) |
| [Подключение внешних скриптов](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/12) |
| [Первое итоговое задание](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/13) |
| [Второе итоговое задание](https://npoed.htmlacademy.ru/htmlcss110/course/2/run/14) |

# С чего начинается HTML [1/14]

Каждый HTML-документ должен начинаться с декларации типа документа или «доктайпа». Тип документа нужен, чтобы браузер мог определить версию HTML и правильно отобразить страницу.

Например, для старой версии HTML 4.01 доктайп выглядит так:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"

"http://www.w3.org/TR/html4/strict.dtd">

А для последней версии HTML уже намного проще:

<!DOCTYPE html>

Последнюю версию HTML ещё называют HTML 5. Но так как эта версия уже принята как стандарт и распространена почти везде, мы будем называть её просто HTML.

# Простейшая HTML-страница [2/14]

Простейшая HTML-страница состоит как минимум из трёх тегов.

Тег <html> — это контейнер, в котором находится всё содержимое страницы, включая теги <head> и <body>. Как правило, тег <html> идёт в документе вторым после доктайпа.

Тег <head> предназначен для хранения других элементов, цель которых — помочь браузеру в работе с данными. Содержимое этого тега не отображается напрямую.

Тег <body> предназначен для хранения содержания веб-страницы (контента), отображаемого в окне браузера.

# Заголовок HTML-страницы [3/14]

Заголовок страницы — это тот текст, который отображается в левом верхнем углу браузера, а также во вкладках.

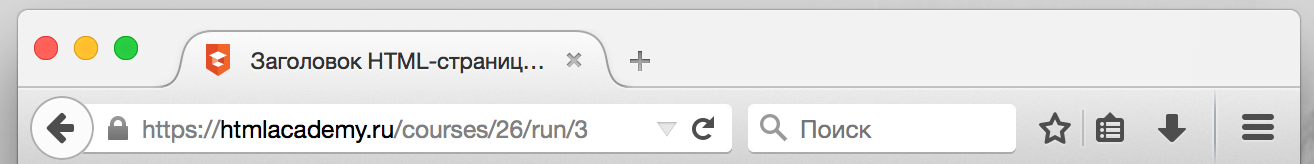
Чтобы задать заголовок страницы, нужно использовать тег <title> внутри тега <head>. Например, вот так:

<head>

<title>Курсы — HTML Academy</title>

</head>

Так выглядит заголовок страницы во вкладке браузера Mozilla Firefox.



# Кодировка HTML-страницы [4/14]

Кодировку HTML-страницы нужно указывать для того, чтобы веб-браузер мог правильно отображать текст на странице. Если браузер неправильно «угадает» кодировку, то вместо текста будут отображаться иероглифы.

Чтобы сообщить браузеру кодировку HTML-страницы, необходимо внутри тега <head> использовать тег:

<meta charset="имя кодировки">

Самая распространённая современная кодировка — utf-8. Используйте её во всех своих проектах.

Для кириллицы в Windows charset часто задавали как windows-1251. Но сейчас это считается плохой практикой.

# Ключевые слова [5/14]

Есть целое семейство тегов <meta>, называемых мета-тегами. Их можно использовать внутри тега <head>.

Мета-теги различаются набором атрибутов и их значений, вот некоторые из атрибутов: content, http-equiv, name и scheme.

Мета-теги хранят полезную для браузеров и поисковых систем информацию. Один из таких тегов — это описание ключевых слов страницы. Задаётся он так:

<meta name="keywords" content="разные, ключевые, слова">

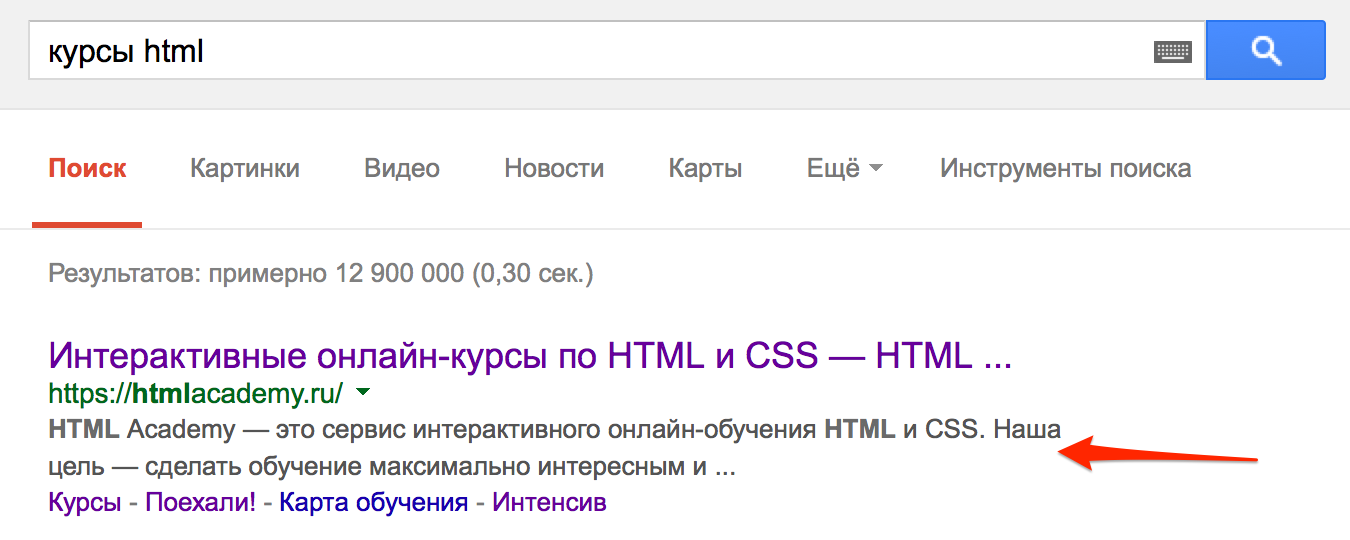
В атрибуте content через запятую перечисляются самые важные слова из содержания страницы. Раньше этот тег был очень важен для поисковиков. Каково положение дел сейчас — большой секрет Яндекса и Гугла.

# Описание содержания страницы [6/14]

Ещё один полезный для поисковых систем мета-тег — краткое описание страницы. Оно задаётся так:

<meta name="description" content="краткое описание">

В атрибуте content должно быть краткое содержание или аннотация страницы. Оно часто используется поисковиками при отображении результатов поиска.



Пойманный нами инженер из Яндекса не признался, важен ли этот тег для ранжирования, но дал ссылку на [рекомендации по составлению описаний](https://help.yandex.ru/webmaster/recommendations/using-meta-desc.xml). Инженера из Гугла мы ещё только выслеживаем, так что следите за новостями.

# HTML-комментарии [7/14]

Комментарий в HTML-коде задаётся так:

<!-- любой текст -->

Текст внутри комментария не отображается браузером на странице. Комментарии обычно используются в следующих случаях:

* Для комментирования кода. Всегда полезно оставить подсказку.
* Для временного отключения кода. Удалять код неудобно, так как его надо будет восстанавливать, а закомментировать и потом раскомментировать — самое лучшее решение.

Комментарии можно использовать в любом месте страницы, кроме тега <title> — внутри него они не работают.

Чтобы быстро закомментировать или раскомментировать строку кода в HTML или CSS редакторе, можете использовать сочетание клавиш **ctrl + /** или **cmd + /**.

# Подключение стилей [8/14]

CSS-стили можно писать внутри HTML-кода страницы или подключать их как внешний файл.

В первом случае стили называются «встроенными» или «инлайновыми», а писать их нужно внутри тега <style>. Этот тег обычно размещают внутри <head>. Например:

<head>

<style>

CSS-код

</style>

</head>

Внутри <style> пишут обычный CSS-код.

Инлайновые стили используют не так часто, например, для оптимизации скорости загрузки страницы. Чаще используют внешние стили, c которыми мы познакомимся позже.

# Тайна CSS-редактора [9/14]

Теперь вы знаете про встроенные стили и можете узнать тайну нашего CSS-редактора.

CSS-код из редактора незаметно добавляется внутрь тега <style>, а этот тег добавляется в мини-браузер.

В этом задании CSS-стили такие же, как и в предыдущем, но вынесены в CSS-редактор и закомментированы. Комментарии в CSS работают так же, как в HTML — позволяют временно отключить какой-то кусок кода.

CSS-комментарии задаются с помощью символов /\* и \*/:

/\*

h1 {

color: red;

}

\*/

# Подключение внешних стилей [10/14]

Чаще всего стили подключают из внешнего файла с расширением .css. Для этого используется тег <link>. Например:

<head>

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

</head>

В атрибуте href задают адрес файла, а атрибут rel="stylesheet" говорит браузеру, что мы подключаем стили, а не что-то другое.

Лучше подключать стили внутри <head>, но это необязательно. Тег <link> будет работать и в другом месте страницы.

В этом задании вы подключите внешний стилевой файл, который расположен по адресу [/assets/course2/style.css](https://npoed.htmlacademy.ru/assets/course2/style.css)щёлкните по ссылке, чтобы открыть этот файл в браузере.

# Подключение скриптов [11/14]

В вебе следующее разделение ролей: HTML отвечает за структуру документа, стили — за его внешний вид, а скрипты — за поведение. С помощью скриптов, например, можно «оживлять» страницу, добавляя анимацию и другие эффекты. Скрипты создаются с помощью языка JavaScript.

Скрипты подключаются так же, как и стили: их либо пишут внутри страницы, либо подключают как внешние файлы.

Встроенные скрипты пишут внутри тега <script>. Например:

<script>

JavaScript-код

</script>

Тег <script> можно использовать в любом месте HTML-документа, но лучше вставлять его в самом конце перед закрывающим тегом </body>.

Часть возможностей JavaScript постепенно переходит в CSS, например, возможность задавать плавное изменение значений свойств. Вы увидите это в задании.

# Подключение внешних скриптов [12/14]

Скрипты чаще всего подключают из внешних файлов с расширением .js. Для этого используют тег <script> с атрибутом src, в котором указывается путь к файлу. Например:

<script src="scripts.js"></script>

Обратите внимание, что тег <script> парный. Если вы подключаете внешние скрипты, то просто ничего не пишите внутри тега.

Внешние скрипты лучше подключать перед закрывающим тегом </body>.

В этом задании вы подключите внешний скрипт, который расположен по адресу: [/assets/course2/scripts.js](https://npoed.htmlacademy.ru/assets/course2/scripts.js)щёлкните по ссылке, чтобы открыть этот файл в браузере.

# Первое итоговое задание [13/14]

Вы познакомились с тегами, необходимыми для создания простейшей HTML-страницы, а также со служебными тегами, которые очень важны для её корректного отображения, оптимизации и продвижения в поисковиках.

Пришло время повторить изученное и выполнить два более сложных задания.

В первом задании закрепим навыки описания мета-информации:

* заголовка страницы;
* ключевых слов;
* краткого описания страницы.

Прочитайте анонс статьи в мини-браузере и дополните его важной мета-информацией.

# Второе итоговое задание [14/14]

Во втором итоговом задании вам нужно самостоятельно подключить к странице внешние ресурсы:

* Стилевой файл [/assets/course2/final.css](https://npoed.htmlacademy.ru/assets/course2/final.css" \t "_blank).
* Файл со скриптами [/assets/course2/final.js](https://npoed.htmlacademy.ru/assets/course2/final.js" \t "_blank).

Не забывайте, что хорошим тоном считается подключать внешние стили в <head>, а скрипты перед закрывающим тегом </body>.

В подключаемых файлах — демонстрация возможностей CSS и JavaScript, которые мы будем изучать в дальнейшем.