# Структура html-документа

Электронное пособие

# Структура HTML документа

Чтобы создать любую веб-страницу, в ней должен присутствовать минимальный набор тегов. Они расположены в примере ниже.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Профессиональное создание сайта в интернете с примерами</title>
        <meta charset="utf-8">
    </head>
    <body>
        <h1>Как создать сайт в интернете</h1>
        <p>Здесь размещается содержание документа, которое видно всем пользователям.</p>
    </body>
</html>
```

Браузер сам интерпретирует все эти теги и выводит их в одном для пользователя виде информации. Вы и сами замечали, что, когда заходите на сайт, на экране не отображаются эти элементы. Вы видите только красивую обертку, которую выводит браузер.

**Тег DOCTYPE**

Тег DOCTYPE всегда идет в самом начале веб-страницы. Он необходим для указания типа текущего документа. А также помимо этого браузер понимает какая версия html используется.

**Тег HTML**

Тег html является контейнером, который заключает в себя все содержимое веб-страницы. Открывающие и закрывающие элементы в документе не обязательны. Однако хороший стиль требует их использования.

Как правило, открывающий тег html идет вторым после определения документа при помощи doctype. Закрывающий всегда идет на странице последним.

Этот тег сообщает браузеру, где наш html документ начинается и где заканчивается. Поэтому все остальные элементы (*кроме doctype*) должны быть вложены внутри этого контейнера.

**Тег head**

Тег head предназначен для хранения других элементов, цель которых помочь браузеру при работе с данными. Также внутри контейнера находятся мета теги, которые используются для хранения информации. Она предназначена для браузеров и поисковых систем.

У head есть один обязательный компонент. Это тег title. Он отвечает за то, как будет выглядеть html заголовок страницы. То, что вы напишите сюда, будет отображаться во вкладке браузера.

```
<head>
    <title>Профессиональное создание сайта в интернете с примерами</title>
</head>
```

**Тег meta**

Еще один элемент, это тег meta. Он не парный и особой ценности не представляет. Однако наиболее важными элементами являются его атрибуты.

Таким образом, с помощью параметров и атрибутов, тег может хранить важную информацию о вашей странице.

Давайте к элементу пропишем параметр charset и в нем укажем предпочтительную кодировку документа.

```
<head>
<title>Профессиональное создание сайта в интернете с примерами</title>
<meta charset="utf-8">
</head>
```

Обратите внимание, как указывается параметр. Его вставляем внутри тега и перед правой закрывающей скобкой. Далее ставим знак равно и в кавычках указываем кодировку документа.

- Обязательно ее указывайте в своих документах. Иначе на вашем сайте вместо текста появятся некрасивые иероглифы.

## Синтаксис html

Немного стоит упомянуть и про синтаксис html. Сразу старайтесь располагать теги в таком удобном виде, как на примере выше. Конечно же, можно расположить их и в хаотичном порядке.

```
<!DOCTYPE html>
<html><head>
<title>Профессиональное создание сайта в интернете с примерами</title><meta charset="utf-8">
</head><body><p>Здесь размещается содержание документа, которое видно всем пользователям.</p></bo
</html>
```

Браузеру будет все равно.

Он отобразит страницу правильно. Но вам самим будет очень неудобно редактировать html код страницы. Отдельные теги всегда располагайте в разных строчках. Причем соблюдайте **правило вложенности**. Все вложенные теги делайте отступом в одну табуляцию от родительского элемента.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Название документа</title>
        <meta charset="utf-8">
    </head>
    <body>
        <h1>Название материала</h1>
        <p>Основной текст</p>
    </body>
</html>
```

То есть тег head должен идти отступом в одну табуляцию относительно тега html. Body идет на уровне с head поскольку он тоже вложен в контейнер html.

В общем, для быстрой и удобной работы с исходным кодом обязательно соблюдайте отступы по уровню вложенности элементов.

Исключение лишь могут составлять head и body. Они используются всего лишь один раз на странице. Поэтому чтобы не уходить далеко вправо при большой вложенности, эти элементы можно делать без отступов.

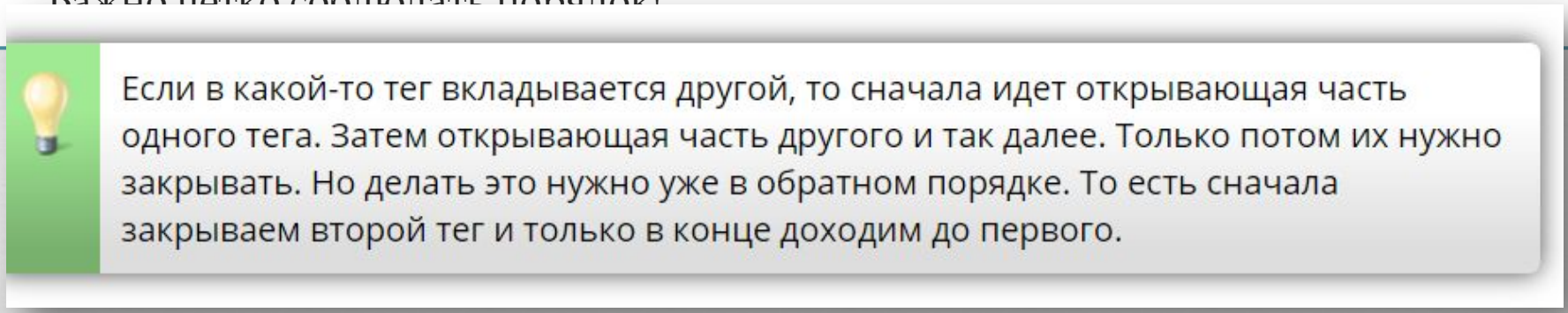
```
<!DOCTYPE html>
<html>
<head>
    <title>Название документа</title>
    <meta charset="utf-8">
</head>
<body>
    <p>Основной текст</p>
</body>
</html>
```

При большой вложенности этот метод поможет сэкономить место и не потеряться в структуре html.

Как вы знаете, теги можно вкладывать друг в друга. Например, текст и картинка находятся в разных тегах. Однако при этом они имеют общий родительский элемент.

Если тег открывается, то он должен быть закрыт. Если он пустой и без зарывающей части, то в такой тег ничего вложить нельзя.

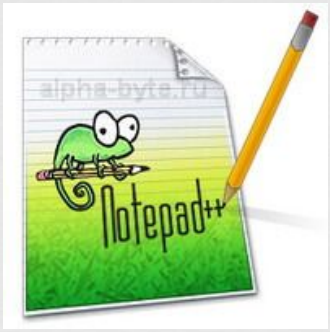
- Важно четко соблюдать порядок!



# Как создать файл html

Возможно, некоторые из вас не знают, как создать файл html. Для этой задачи есть много специальных программ:

- ✔ Brackets
- ✔ Блокнот
- ✔ Notepad++
- ✔ Sublime Text
- ✔ Atom и другие




Если у вас Windows, то Блокнот будет уже доступен по умолчанию. Но им не удобно пользоваться. Рекомендую для начала использовать бесплатный редактор html страниц Notepad++. В нем есть много возможностей для удобной работы с кодом.

Когда запустите редактор, то перед вами уже будет новый пустой файл. Пропишите в нем тот скелет html страницы (*каркас*), который мы выше разбирали. Он должен всегда присутствовать.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Профессиональное создание сайта в интернете с примерами</title>
        <meta charset="utf-8">
    </head>

    <body>
        <h1>Как создать сайт в интернете</h1>
        <p>Здесь размещается содержание документа, которое видно всем пользователям.</p>
   </body>
</html>
```

Текст можете изменить под себя. В общем, это будет самая простая html страница.

# HTML кодировка

При создании страниц всегда прописывайте кодировку в `meta` теге. Это служебный тег, который вставляют в `head`.

Вообще, этот `meta` элемент многогранный. Он может отвечать не только за кодировку, но и за многие другие вещи. Однако об этом поговорим в следующих статьях.

Сейчас у нас он играет роль указателя кодировки. Здесь есть специальная запись. Она говорит браузеру о том, что данный документ создан в кодировке **utf-8**.

```
<meta charset="utf-8">
```

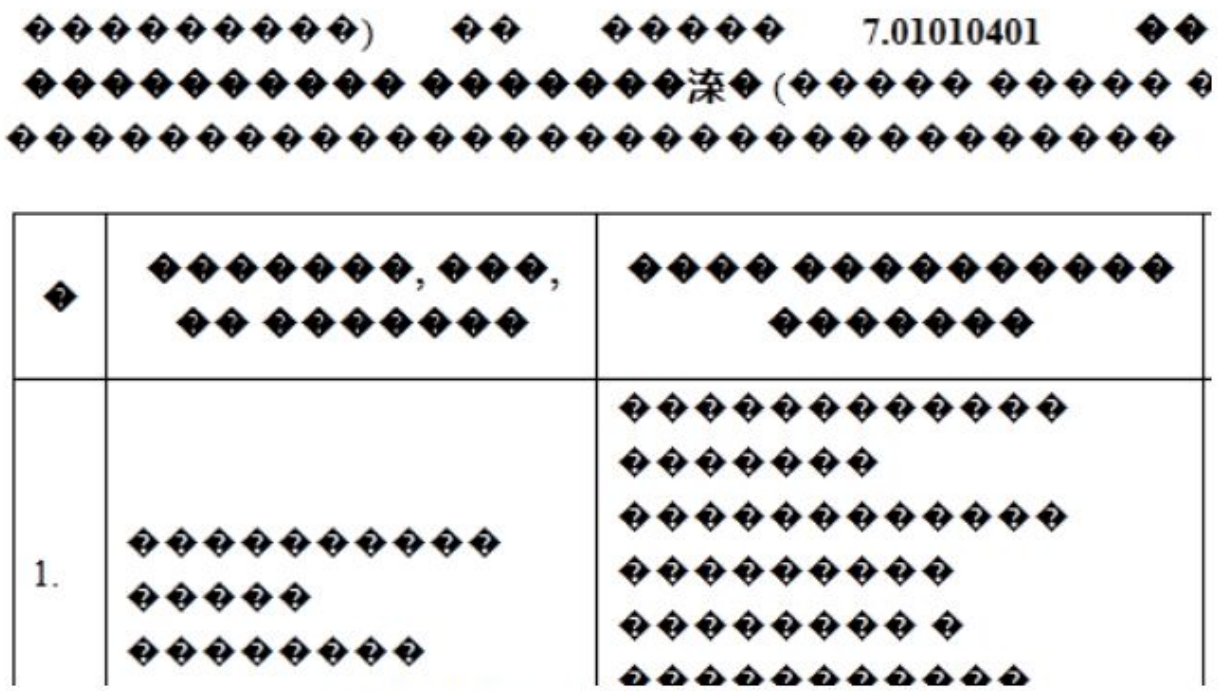
Для чего вообще нужна кодировка?

Не секрет, что во всем мире пишут и разговаривают на разных языках. Если мы создаем страницу и пишем на русском языке, то в Германии будут создавать и писать уже на немецком. В других странах уже свой язык.

Когда мы заходим на страницу, браузер уже понимает, на каком языке написан сайт. Также он понимает какой язык для текста используется. Соответственно, он пытается отобразить этот текстовый контент в зависимости от заданной кодировки.

К сожалению, это не всегда получалось.

Если бы мы заходили на китайские сайты, то у нас вместо их символов могли высвечиваться черные квадраты или знаки вопросов. То есть браузер просто не понимал, что это за символы.



| | | |
|---|---|---|
| 1. | | |

Неправильно задана html кодировка

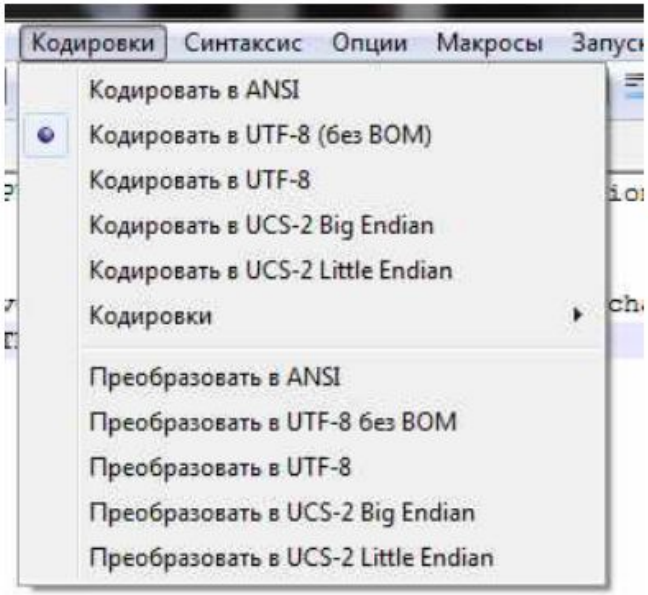
Как решение этой проблемы была создана универсальная кодировка utf-8.

Когда она стала уже стандартом, отпала необходимость указывать конкретную кодировку для определенного языка. Достаточно было указать универсальную **utf-8** и у вас сайт отображался бы нормально во всех браузерах.

## Сохраняем html utf-8

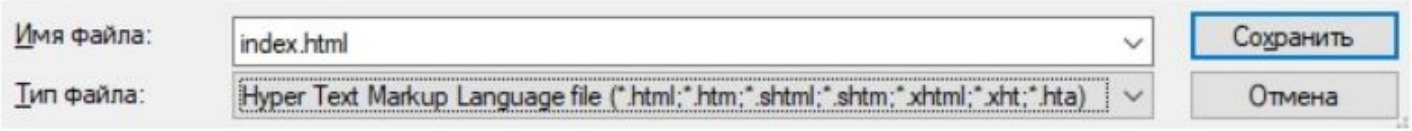
Чтобы все работало и браузер отображал страницу, желательно сохранить сам html документ в utf-8. В этом программа Notepad ++ нам очень хорошо помогает.

Здесь есть пункт «Кодировки». Если у вас сейчас кодировка не стоит (*как у меня на нижнем рисунке*), то щелкните по пункту «Преобразовать в UTF-8 без BOM».



# Как сохранить html страницу

Сейчас расскажу, как сохранить html страницу правильно. После того, как задали кодировку и структуру html страницы, вам нужно нажать на «сохранение». Заходим во вкладку «Файл» —> «Сохранить как...».




Имя файла может быть любое. Но по умолчанию и особенно если это главная страница сайта, то сохраняйте ее как **index.html**.

Когда набираем адрес сайта без указания конкретной страницы, это значит, что мы запрашиваем страницу index.html. То есть, когда мы обращаемся к серверу, он развернет именно индексную страницу.

Если же мы указываем какую-то конкретную страницу, то в итоге она и будет выводиться. Но если адрес не указан, то будет возвращена индексная страница. Поэтому по умолчанию мы называем страницу index.html.

И как вы уже должны заметить, расширение файла должно быть **html**. То есть в типе файла выбираем «Hyper Text Markup Language» и нажимаем «Сохранить».

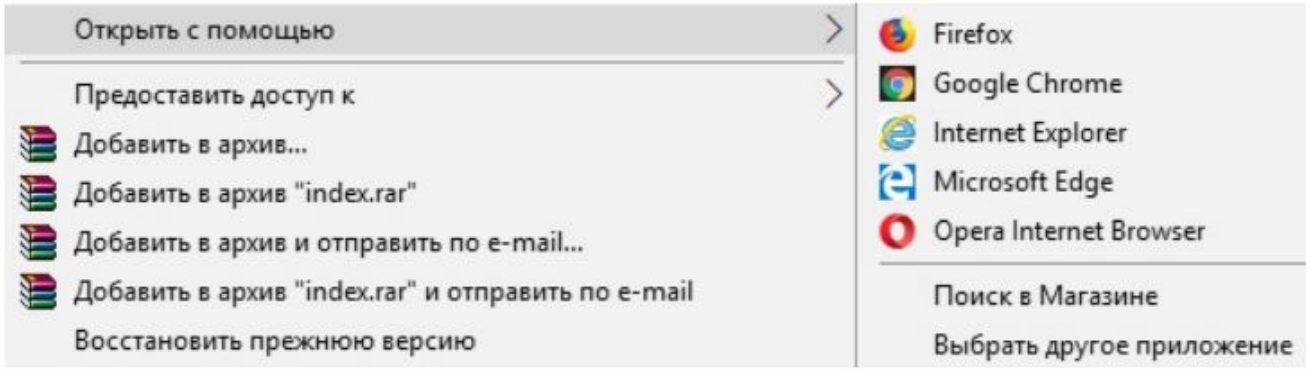
# Как открыть html файл

Как открыть html файл чтобы посмотреть наглядное отображение страницы. Для этого используем любой браузер: Google Chrome, Opera, Edge, Mozilla Firefox, Safari и другие.

Не зря мы сохраняем файлы с расширением html. Ведь только этот тип документа воспроизводят браузеры. А сам язык HTML — это стандарт разметки страницы для последующего воспроизведения браузерами.

В общем, кликаем по файлу правой кнопкой мыши.

В меню наводим на «Открыть с помощью» и выбираем нужный вариант браузера. Если они не отобразятся, то кликаем на «Выбрать другое приложение». Там уже среди списка выбираем нужную программу для чтения html страниц.



# Структура HTML документа:

```
<HTML>
  <HEAD>
    <TITLE>
  название окна
    </TITLE>
  </HEAD>

  <BODY>
  информация страницы
  </BODY>
</HTML>
```

заголовок

тело

**<HTML>** - указывает программе просмотра что это HTML документ;

**<HEAD>** - заголовок, содержит название документа и справочную информацию;

**<TITLE>** - название Web-страницы, отображается в строке заголовка браузера;

**<BODY>** - располагается всё содержание страницы.