УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет инфокоммуникаций

Кафедра инфокоммуникационных технологий

ВЕБ-ТЕХНОЛОГИИ В ИНФОКОММУНИКАЦИЯХ

лабораторный практикум

Минск 2021

Содержание

Лабораторная работа №2
Создание HTML-документа

Семантические элементы HTML5 доступно описывают свой смысл или назначение как для браузеров, так и для веб-разработчиков.

До появления стандарта HTML5 вся разметка страниц осуществлялась преимущественно с помощью элементов `<div>`, которым присваивали классы `class` или идентификаторы `id` для наглядности разметки (например, `<div id="header">`). С их помощью в HTML-документе размещали верхние и нижние колонтитулы, боковые панели, навигацию и многое другое.

Стандарт HTML5 предоставил новые элементы для структурирования, группировки контента и разметки текстового содержимого. Новые семантические элементы позволили улучшить структуру веб-страницы, добавив смысловое значение заключенному в них содержимому (было `<div id="header">`, стало `<header>`). Для отображения внешнего вида элементов не задано никаких правил, поэтому элементы можно стилизовать по своему усмотрению. Для всех элементов доступны глобальные атрибуты.

Согласно спецификации HTML5 каждый элемент принадлежит к определенной (ноль или более) категории. Каждая из них группирует элементы со схожими характеристиками. Выделяют следующие общие категории:

- Метаданные.
- Потоковое содержимое.
- Секционное содержимое.
- Заголовочное содержимое.
- Текстовое содержимое.
- Встроенное содержимое.

Элемент документа

Элемент <html>

Категории содержимого: нет.

Контекст, в котором этот элемент может быть использован: как корневой элемент HTML-документа. Везде, где разрешен фрагмент поддокумента в составном документе, например, внутри `<iframe>`.

Пропуск тегов: начальный тег `<html>` может быть пропущен, если сразу за тегом не идет комментарий. Закрывающий тег `</html>` также может быть пропущен, если перед ним нет комментария.

```
<!DOCTYPE html>
  <head>
    <title>Тест</title>
  </head>
  <body>
    <h1>Тестовая страница</h1>
  </body>
```

Элемент `<html>` представляет корень HTML-документа (элемент верхнего уровня). Рекомендуется указывать атрибут `lang` с указанием языка документа. Это помогает инструментам синтеза речи для определения произношения, инструментам перевода для определения правил перевода и т.д.

Все остальные элементы должны быть потомками элемента `<html>`. Все, что находится за пределами `<html>...</html>`, не воспринимается браузером как HTML-код и никак им не обрабатывается.

Базовый HTML-документ выглядит так:

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <title>Название документа</title>
  </head>
  <body>
    <h1>Заголовок документа</h1>
    <p>Абзац текста</p>
    <p>
      <a href="another-html-document.html">
        Текстовая ссылка на another-html-document.html
      </a>
    </p>
    <!-- это комментарий -->
  </body>
</html>
```

Над тегом <html> в самом начале каждого документа указывается тип документа, который объясняет, какой тип HTML следует ожидать и, следовательно, какие спецификации валидаторов (например, валидатор HTML W3C) должны проверять данный документ.

Тип документа также служит для того, чтобы браузер отображал страницу в так называемом «стандартном режиме». В стандартном режиме браузеры обычно пытаются отобразить страницу в соответствии со спецификациями CSS, то есть предполагается, что документ создан с учетом веб-стандартов.

Метаданные документа

К метаданным относится содержимое, которое устанавливает представление или поведение остального содержимого, отношения документа с другими документами, или передает другую «внешнюю» информацию.

Элемент <head>

Категории содержимого: нет.

Контекст, в котором этот элемент может быть использован: как первый элемент в элементе <html>.

Пропуск тегов: начальный тег <head> может быть пропущен, если элемент <head> пуст, или если сразу после него идет другой HTML-элемент. Закрывающий тег </head> может быть пропущен, если он не следует сразу за пробелом или за комментарием. Для элемента доступны глобальные атрибуты.

Раздел <head>...</head> содержит набор технической информации (метаданных) о текущей веб-странице: заголовок, описание, ключевые слова для поисковых машин, кодировку и т.д. Введенная в нем информация не отображается в окне браузера, однако содержит данные, которые указывают браузеру, как следует обрабатывать страницу.

```
<!DOCTYPE html>
<html lang="ru">
  <title>Тест</title>
   <body>
    <h1>Тестовая страница</h1>
   </body>
```

Набор метаданных может быть как большим, так и маленьким:

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <title>Документ с небольшим head</title>
  </head>
  <body>
  ...
```

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Документ с большим head</title>
    <link rel="stylesheet" href="default.css">
    <link rel="stylesheet alternate" href="big.css" title="Большой текст">
    <script src="main.js"></script>
  </head>
  <body>
  ...
```

Элемент <title>

Категории содержимого: метаданные.

Контекст, в котором этот элемент может быть использован: в элементе `<head>`, не содержащем других элементов `<title>`.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны глобальные атрибуты.

Элемент `<title>` представляет заголовок или название документа (веб-страницы). Авторы должны использовать заголовки, которые дают поисковой системе понять, что содержится на странице, даже если заголовки используются вне контекста, например, в истории, закладках пользователя или в результатах поиска. Заголовок документа может отличаться от заголовка первого уровня, поскольку `<h1>` не должен стоять отдельно, когда он вырван из контекста.

```
<title>Подробное руководство по HTML</title>
...
<h1>Что такое HTML</h1>
<p>...</p>
```

Текст внутри `<title>` отображается браузером в заголовке окна. Также этот текст будет содержать ссылку на ваш сайт на странице результатов поиска. Длина заголовка должна быть не более 60 символов, чтобы поместиться полностью.

В одном документе должно быть не более одного элемента `<title>`. Элемент `<title>` является обязательным в большинстве ситуаций, но, если протокол более высокого уровня предоставляет информацию о заголовке, например, в строке «Тема» электронного письма, когда HTML используется в качестве формата создания электронного письма, элемент `<title>` может быть опущен.

Элемент <base>

Категории содержимого: метаданные.

Контекст, в котором этот элемент может быть использован: в элементе `<head>`, не содержащем других элементов `<base>`.

Пропуск тегов: отсутствует закрывающий тег.

Для элемента доступны глобальные атрибуты.

Элемент `<base>` с помощью атрибута `href` предоставляет базовый URL документа для парсинга всех относительных URL-адресов на странице, установленных атрибутами `src` и `href`.

Атрибут target задает тип окна просмотра по умолчанию при переходе по всем гиперссылкам.

В одном документе может быть только один элемент <base> и он должен иметь атрибут href, target или оба сразу.

Элемент <base> должен находиться перед любыми другими элементами в дереве, которые имеют атрибуты, определенные как принимающие URL, кроме элемента <html>.

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <title>Пример для элемента base</title>
    <base href="https://www.bsuir.by/ru/kaf-ikt/index.html">
  </head>
  <body>
    <p>Посетите страницу <a href="o-kaf-ikt.html">кафедры</a>.</p>
  </body>
</html>
```

Ссылка в приведенном выше примере будет ссылкой на https://www.bsuir.by/ru/kaf-ikt/o-kaf-ikt.html.

Элемент <link>

Категории содержимого: метаданные. Если его использование разрешено в <body> – потоковое или текстовое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидаются метаданные. В элементе <noscript>, который является дочерним элементом элемента <head>. Если элемент разрешен в <body> – там, где ожидается текстовое содержимое.

Пропуск тегов: отсутствует закрывающий тег.

Для элемента доступны глобальные атрибуты, а также атрибуты, приведенные в таблице:

Атрибут	Описание, принимаемое значение
href	Задает адрес гиперссылки.
crossorigin	Описывает, как элемент обрабатывает CORS-запросы, предназначен для использования со ссылками на внешние ресурсы.
rel	Задает тип указанной ссылки, может содержать как одно значение, так и набор разделенных пробелами ключевых слов: alternate, dns-prefetch, icon, next, pingback, preconnect, prefetch, preload, prerender, search, serviceworker.
rev	Описывает обратное отношение текущего документа к связываемому документу, как определено атрибутом href.
media	Указывает, к какому типу медиа относится ресурс. Значение должно быть допустимым списком медиазапросов.
nonce	Используется в проверках политики безопасности контента, представляет криптографический одноразовый номер, который может использоваться политикой безопасности содержимого, чтобы определить, будет ли внешний ресурс, указанный в ссылке, загружаться и применяться к документу.
hreflang	Задает язык связанного ресурса.
type	Устанавливает подсказку для типа ссылочного ресурса.

referrerpolicy	Указывает URL источника запроса при переходе с одной страницы на другую.
sizes	Задает размеры иконок (для rel="icon") для визуальных медиа, значение носит рекомендательный характер.
title	Устанавливает заголовок ссылки, альтернативное имя таблицы стилей.

Атрибут href элемента <link> позволяет связывать HTML-документ с различным видами ресурсов, например, таблицами стилей, скриптами, альтернативными формами документа и ссылками навигации (оглавление, предыдущие и последующие страницы, уведомления об авторских правах и т.п.).

Тип связанного ресурса задается значением обязательного атрибута rel.

С помощью элемента <link> можно создать две категории ссылок: ссылки на внешние ресурсы и гиперссылки. Например, следующий элемент ссылки создает две гиперссылки (на одну и ту же страницу):

```
<link rel="author license" href="/about">
```

Семантика первой состоит в том, что целевая страница содержит информацию об авторе текущей страницы, семантика второй заключается в том, что целевая страница содержит информацию о лицензии, под которой предоставляется текущая страница.

Гиперссылки, созданные с помощью элемента <link> и его атрибута rel, применяются ко всему документу. Это отличается от атрибута rel элементов <a> и <area>, который указывает тип ссылки, контекст которой определяется местоположением ссылки в документе.

Если значения атрибута rel содержат только ключевые слова, разрешенные в <body>, элемент <link> можно использовать там, где ожидается фразовое содержание, то есть внутри <body>.

Элемент <meta>

Категории содержимого: метаданные.

Контекст, в котором этот элемент может быть использован: если для элемента указаны атрибуты charset и http-equiv, то в элементе <head>. Если значением атрибута не является content-type, то внутри элемента <noscript>, являющимся дочерним элементом <head>. Если присутствует атрибут name – там, где ожидаются метаданные.

Пропуск тегов: отсутствует закрывающий тег.

Для элемента доступны глобальные атрибуты, а также атрибуты, приведенные в таблице:

Атрибут	Описание, принимаемое значение
charset	Определяет кодировку символов, используемую в документе. В документе должен быть один элемент <meta> с атрибутом charset. Необходимо использовать utf-8 или другую кодировку, совместимую с ASCII.
content	Задает значение метаданных документа или прагма директив.
http-equiv	Задает прагма директиву.
name	Устанавливает название/имя метаданных документа.

Элемент <meta> представляет различные виды метаданных, которые не могут быть выражены с использованием элементов <title>, <base>, <link>, <style> и <script>.

Для элемента обязательно должен быть определен один из атрибутов – name, http-equiv или charset. Если указан атрибут name или http-equiv, также должен присутствовать атрибут content (или пропущен, если нет соответствующих значений).

Стандартные метаданные приведены ниже.

Для атрибута name доступны следующие значения (чувствительны к регистру):

- application-name – значение должно быть короткой строкой произвольной формы, содержащей название веб-приложения, которое представляет страница. Если страница не является веб-приложением, application-name не должно использоваться. В одном документе должно быть не более одного названия веб-приложения. Браузеры могут использовать название веб-приложения в пользовательском интерфейсе вместо <title>, поскольку <title> может содержать сообщения о состоянии и тому подобное, относящиеся к состоянию страницы в определенный момент времени, а не просто как название приложения.
- author – значение должно быть строкой произвольной формы с указанием имени одного из авторов страницы.
- description – значение должно быть строкой произвольной формы, описывающей страницу. Значение должно быть подходящим для использования в каталоге страниц, например, в поисковой системе.
- generator – значение должно быть строкой произвольной формы, которая идентифицирует один из пакетов программного обеспечения, использованных для создания документа. Это значение не должно использоваться на страницах, разметка которых не создается программным обеспечением, например на страницах, разметка которых была написана пользователем в текстовом редакторе. <meta name="generator" content="WordPress 5.8">.
- keywords – значение должно быть набором разделенных запятыми ключевых слов, относящихся к странице. Многие поисковые системы не рассматривают такие ключевые слова, потому что эта функция исторически использовалась ненадежно и спамила результаты поиска. <meta name="keywords" content="шрифт,шрифты,типографика">.
- referrer – необязательное поле заголовка HTTP, которое позволяет отслеживать перемещения пользователей между страницами в инструментах аналитики, а также понять происхождение входящего трафика. Реферер передается при переходе с http на любой тип сайта, при переходе с https на https, и не передается при переходе с https на http. В наиболее распространенной ситуации это означает, что, когда пользователь щелкает гиперссылку в браузере, на сервер отправляется запрос, содержащий целевую веб-страницу. Запрос может содержать поле referer, в котором указана последняя страница, на которой был пользователь (то есть та, на которой он щелкнул ссылку). Значения атрибута content:
    - no-referrer – не передает никакую информацию о реферере.
    - no-referrer-when-downgrade – передает реферальные данные только сайтам на HTTPS. Поведение браузера по умолчанию, если не указано иное.
    - unsafe-url – всегда передает полный URL реферера.
    - origin-when-cross-origin – отправляет полный URL при переходе на страницы в рамках одного сайта, вне зависимости от протокола, а на все остальные – только базовый домен/поддомен.

- viewport – позволяет определять конкретные характеристики области просмотра, например, ширину макета и коэффициент масштабирования веб-страниц. Можно запретить или ограничить пользователям возможность масштабирования, используя такие значения, как content="user-scalable=no" или content="width=device-width, initial-scale=1.0, maximum-scale=1.0".

Могут быть конкретные случаи использования, в которых целесообразно предотвращение масштабирования пользователями, например, приложения-карты – когда пользовательские функции масштабирования обрабатываются с помощью сценариев, но в целом такой практики следует избегать.

Распознаваемые свойства атрибута content:

- width – определяет ширину области просмотра, значением может быть определенное количество пикселей, например, width=768 или ключевое слово device-width (соответствует 100vw).
  <meta name="viewport" content="width=device-width, initial-scale=1">
- height – определяет высоту области просмотра, значением может быть определенное количество пикселей, например, height=480 или ключевое слово device-height (соответствует 100vh)
- initial-scale – указывает браузеру соотношение пикселей CSS и устройства.
- minimum-scale – определяет наименьший допустимый коэффициент масштабирования.
- maximum-scale – определяет максимально допустимый коэффициент масштабирования.
- user-scalable – указывает, может ли коэффициент масштабирования быть изменен в результате взаимодействия с пользователем или нет.

Когда атрибут http-equiv указан в элементе <meta>, элемент <meta> является прагма директивой, которая предоставляет дополнительную информацию о документе:

- content-type – является альтернативной формой установки атрибута charset. <meta http-equiv="content-type" content="text/html; charset=utf-8">
- default-style – задает имя альтернативной таблицы стилей, используемой по умолчанию. <meta http-equiv="default-style" content="default">.
- refresh – устанавливает таймер для обновления и перенаправления. Например, главная страница новостного сайта может содержать следующую разметку в элементе <head>, чтобы обеспечить автоматическую перезагрузку страницы с сервера каждые пять минут: <meta http-equiv="refresh" content="300">. Последовательность страниц может использоваться в качестве автоматического слайд-шоу, если каждая страница обновляется до следующей страницы в последовательности с использованием следующей разметки: <meta http-equiv="refresh" content="20; url=page4.html">.
- content-security-policy – позволяет настроить политику защиты содержимого, с помощью которой можно защищаться, например, от межсайтового скриптинга: <meta http-equiv="content-security-policy" content="script-src 'self'">.

Элемент <style>

Категории содержимого: метаданные.

Контекст, в котором этот элемент может быть использован: где ожидаются метаданные. Внутри элемента <noscript> который является дочерним элементом <head>. Внутри <body>, где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны глобальные атрибуты, а также атрибуты, приведенные в таблице:

Атрибут	Описание, принимаемое значение
media	Указывает, к каким медиа применяются стили. Значение должно быть допустимым списком медиазапросов. Браузер должен применять стили, когда значение атрибута media соответствует среде и применяются другие соответствующие условия. Если атрибут media пропущен, по умолчанию он принимает значение all, то есть стили применяются ко всем видам медиа.
nonce	Представляет криптографический одноразовый номер, который может использоваться политикой безопасности содержимого, чтобы определить, будет ли стиль, указанный элементом, применяться к документу.
type	Устанавливает язык таблиц стилей, значение должно быть допустимым MIME-типом. Значением по умолчанию является text/css.
title	Задает альтернативное имя таблиц стилей.

Элемент <style> позволяет авторам встраивать информацию о стилях в свои документы. Элемент не представляет какое-либо содержимое для пользователя.

Элемент <style> желательно использовать внутри раздела <head>.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Моя любимая книга</title>
    <style>
      em {
          font-style: normal;
          color: red;
      }
    </style>
  </head>
  <body>
    <p>Моя <em>любимая</em> книга - «Приключения Алисы в Стране чудес».</p>
  </body>
</html>
```

Разделы документа

Структура документа В HTML5 состоит из разделов и подразделов. Разделы могут быть представлены в виде схем документа по аналогии с оглавлением. Каждый секционный элемент имеет свою собственную схему, поэтому каждый раздел можно начинать с заголовка <h1>.

Браузеры могут обрабатывать эту информацию для создания содержания (оглавления) документа, которое впоследствии будет использоваться ассистивными (вспомогательными) технологиями для облегчения навигации по статье или поисковыми системами для улучшения результатов поиска.

Отдельно выделяют корневые секционные элементы. Они отличаются от секционных элементов, но могут также иметь схему.

Элемент <body>

Категории контента: корневой секционный.

Контекст, в котором этот элемент может быть использован: как второй элемент в элементе <html>.

Пропуск тегов: начальный тег <body> может быть опущен, если элемент пуст, или если первое, что внутри элемента не является пробелом или комментарием, за исключением случаев, когда первое, что идет за тегом <body> являются элементы <meta>, <link>, <script> или <style>.

Закрывающий тег </body> может быть опущен, если перед ним нет комментария.

```
<!DOCTYPE html>
  <title>Тест</title>
   <h1>Тестовая страница</h1>
```

Элемент <body> представляет содержимое документа.

Для элемента доступны глобальные атрибуты.

Элемент <article>

Категории контента: потоковое содержимое, секционное содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Элемент <article> представляет собой законченное или автономное произведение в документе, странице, приложении или сайте. Может дублироваться на других страницах сайта и содержать внутри другие элементы <article>, которые по содержанию имеют близкое отношение к содержанию внешней статьи. Если на странице присутствует только одна статья с заголовком и текстовым содержимым, она не нуждается в обертке элементом <article>.

Общее правило заключается в том, что элемент <article> уместен только в том случае, если содержимое элемента будет явно указано в схеме документа. Каждая статья должна быть идентифицирована, обычно путем включения заголовка (элемент <h1-h6>) в качестве дочернего элемента для элемента <article>.

```
<article>
    <header>
        <h2><a href="https://city.by">Короткая заметка о ношении шорт</a></h2>
        <p>Опубликовано в пятницу, 16 августа 2021 Васей Пупкиным.
            <a href="https://city.by/short-note/#comments">6 комментариев</a></p>
    </header>
    <p>Мне задали интересный вопрос: почему ты носишь шорты, а не длинные брюки? Шорты - это удобный и модный предмет одежды, который
    сочетается со многими вещами. Кроме того, в них так приятно ходить жарким летом.</p>
    <p>Короткий ответ: мне нравится носить шорты, длинный ответ ...</p>
    <p><a href="https://city.by/short-note/">Продолжить чтение: Короткая заметка о ношении шорт</a></p>
</article>
```

```
<section>
      <article>
        <h2><a href="">Весна приходит (и уходит) в графстве Суссекс</a></h2>
        <p>Вчера я присоединился к Brooklyn Bird Club для нашей ежегодной поездки в Западный Нью-Джерси, в частности, в Hyper Humus,
        относительно недавно обнаруженную «горячее место».</p>
      </article>

      <article>
        <h2><a href="">Как стать бердвотчером?</a></h2>
          <p>Птицы – почти единственная связь современного городского человека с дикой природой. Благодаря бердвотчингу вы, скорее
          всего, начнете больше путешествовать, причем по самым неожиданным местам. Если у вас есть дети, можно изучать птиц вместе –
          это идеальное семейное хобби. </p>
      </article>

    <nav>
      <a href="">&laquo; Предыдущие записи</a>
    </nav>
</section>
```

Для элемента доступны глобальные атрибуты.

Элемент <section>

Категории контента: потоковое содержимое, секционное содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Элемент <section> представляет общий раздел документа или приложения, группируя тематическое содержимое. Каждый раздел должен быть идентифицирован, обычно путем включения заголовка (элемент <h1-h6>) в качестве дочернего элемента <section>.

Примерами разделов могут быть главы, различные страницы во вкладках или пронумерованные разделы. Домашняя страница веб-сайта может быть разбита на разделы для введения, новостей и контактной информации.

Авторам рекомендуется использовать элемент <article> вместо элемента <section>, когда контент завершен или самодостаточен.

Элемент <section> не является универсальным контейнерным элементом. Когда элемент нужен только для стилизации или для удобства написания сценариев, рекомендуется использовать вместо этого элемент . Общее правило заключается в том, что элемент <section> уместен только в том случае, если содержимое элемента будет явно указано в схеме документа.

```
<article>
  <header>
    <h2>Яблоки</h2>
    <p>Вкусные, восхитительные фрукты!</p>
  </header>
  <p>Яблоко является плодом яблони.</p>
  <section aria-label="Красные яблоки">
    <h3>Ред Делишес</h3>
    <p>Эти ярко-красные яблоки чаще всего встречаются во многих супермаркетах.</p>
  </section>
  <section aria-label="Зеленые яблоки">
    <h3>Гренни Смит</h3>
    <p>Эти сочные зеленые яблоки станут отличной начинкой для яблочных пирогов.</p>
  </section>
</article>
```

Для элемента доступны глобальные атрибуты.

<article> внутри <section>

Можно создавать родительские элементы <section> с вложенными элементами <article>, в которых есть один или несколько элементов <article>. Не все страницы должны быть устроены именно так, но это допустимый способ вложения элементов. Например, основная область контента страницы содержит два блока со статьями разной тематики. Можно сделать на этом акцент, поместив каждую статью одной тематики внутрь элемента <section>.

```
<section>
  <h1>Мои заметки</h1>
  <h2>Заметки о природе</h2>
    <article>
      <h3>...</h3>
      <p>...</p>
    </article>
    <article>
      <h3>...</h3>
      <p>...</p>
    </article>
</section>
<section>
  <h2>Исторические заметки</h2>
    <article>
      <h3>...</h3>
      <p>...</p>
    </article>
    <article>
      <h3>...</h3>
      <p>...</p>
    </article>
</section>
```

Элемент <nav>

Категории контента: потоковое содержимое, секционное содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Элемент `<nav>` представляет собой раздел страницы с навигационными ссылками, который ссылается на другие страницы или части внутри страницы, при этом не обязательно должен находиться внутри `<header>`. На странице может быть несколько элементов `<nav>`.

В случаях, когда содержимое элемента `<nav>` представляет список элементов, рекомендуется использовать разметку списка. Не заменяет теги `<ul>` или `<ol>`, он просто их обрамляет.

Не все группы ссылок на странице должны быть обернуты элементом `<nav>` – этот элемент предназначен главным образом для разделов, состоящих из основных навигационных блоков. В частности, нижние колонтитулы обычно имеют краткий список ссылок на различные страницы сайта, такие как условия обслуживания, домашняя страница и страница об авторских правах. Для таких случаев достаточно одного элемента `<footer>`.

```
<body>
  <h1>Веб-технологии/h1>
  <nav>
    <ul>
      <li><a href="/">Главная</a></li>
      <li><a href="/events">Текущие события</a></li>
      ...
    </ul>
  </nav>
  <article>
    <header>
      <h2>Афиша мероприятий</h2>
    </header>
    <nav>
      <ul>
        <li><a href="#public">Лекции</a></li>
        <li><a href="#practice">Лабораторные работы</a></li>
        ...
      </ul>
    </nav>
    <div>
      <section id="public">
        <h2>Лекции</h2>
        <p>...</p>
      </section>
      <section id="practice">
        <h2>Лабораторные работы</h2>
        <p>...</p>
      </section>
      ...more...
    </div>
    <footer>
      <p><a href="?edit">Редактировать</a> | <a href="?delete">Удалить</a> | <a href="?rename">Переименовать</a></p>
    </footer>
  </article>
  <footer>
    <p><small>© 2021 Веб-технологии</small></p>
  </footer>
</body>
```

Внутрь элемента можно добавлять заголовки:

```
<nav>
  <h2>...</h2>
    <ul>
      <li><a>...</a></li>
      <li><a>...</a></li>
      <li><a>...</a></li>
    </ul>
</nav>
```

Для элемента доступны глобальные атрибуты.

Элемент <aside>

Категории контента: потоковое содержимое, секционное содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Элемент `<aside>` представляет раздел страницы, который состоит из содержимого, косвенно связанного с родительским секционным элементом и который можно рассматривать отдельно от него. Чаще всего элемент позиционируется как боковая колонка (как в книгах) и включает в себя группу элементов: `<nav>`, цифровые данные, цитаты, рекламные блоки, архивные записи. Не подходит для блоков, просто позиционированных в стороне.

```
<aside>
  <h2>Республика Беларусь</h2>
  <p>Беларусь – страна для жизни.</p>
</aside>
```

```
<body>
  <header>
    <h1>Моя замечательная страница</h1>
    <p>Мой слоган</p>
  </header>
  <aside>
    <nav>
      <h2>Обо мне</h2>
      <ul>
        <li><a href="https://www.bsuir.by/ru/kaf-ikt/makeychik-e-g">Интересная ссылка</a>
      </ul>
    </nav>
    <nav>
      <h2>Архивы</h2>
      <ol reversed>
        <li><a href="/last-post">Моя последняя запись</a>
        <li><a href="/first-post">Моя первая запись</a>
      </ol>
    </nav>
  </aside>
  <article>
    <h2>Моя последняя запись</h2>
    <p>Это моя последняя запись.</p>
    <footer>
      <p><a href="/last-post" rel=bookmark>Ссылка</a>
    </footer>
  </article>
  <article>
     <h2>Моя первая запись</h2>
    <p>Это моя первая запись.</p>
    <aside>
        <h1>Публикация</h1>
        <p>Опубликовывать статьи - это весело!</p>
    </aside>
    <footer>
      <p><a href="/first-post" rel=bookmark>Ссылка</a>
    </footer>
  </article>
  <footer>
    <nav>
      <a href="/archives">Архивы</a> –      <a href="/about">Обо мне</a> –      <a href="/copyright">Copyright</a>
    </nav>
  </footer>
</body>
```

Для элемента доступны глобальные атрибуты.

Элементы <h1>, <h2>, <h3>, <h4>, <h5> и <h6>

Категории контента: потоковое содержимое, секционное содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Элементы <h1-h6> представляют заголовки для своих разделов. Эти элементы имеют ранг, определяемый числом в их имени. Элемент <h1> имеет наивысший ранг, элемент <h6> имеет наименьший ранг, а два элемента с одинаковым именем имеют одинаковый ранг. Используйте ранг элементов заголовка, чтобы создать схему документа.

Элементы <h1-h6> не должны использоваться для разметки подзаголовков, альтернативных заголовков и слоганов, если только они не предназначены для заголовка нового раздела или подраздела.

```
<body>
<h1>Заголовок верхнего уровня</h1>
 <section><h2>Заголовок второго уровня</h2>
  <section><h3>Заголовок третьего уровня</h3>
   <section><h4>Заголовок четвертого уровня</h4>
    <section><h5>Заголовок пятого уровня</h5>
     <section><h6>Заголовок шестого уровня</h6>
     </section>
    </section>
  </section>
 </section>
</section>
</body>
```

Для элемента доступны глобальные атрибуты.

Элемент <header>

Категории контента: потоковое содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны глобальные атрибуты.

Элемент <header> представляет вводное содержимое для его ближайшего предка – элемента <main> или элемента из категории секционного содержимого или корневого секционного элемента. Элемент <header> обычно содержит группу вводных или навигационных элементов.

Если элемент <header> является ближайшим предком элемента <body> и не находится внутри <main>, он представляет вводное содержимое для страницы в целом.

Элемент <header> обычно содержит заголовок раздела (элемент <h1–h6>), но это не обязательно. Элемент <header> также можно использовать как элемент-обертку для оглавления раздела, формы поиска или любых уместных логотипов. В документе может содержаться одновременно несколько элементов <header>, и они могут располагаться в любой части страницы.

Элемент <header> не является секционным содержимым, он не вводит новый раздел.

```
<article>
  <header>
    <h1>HTML5: Полное руководство</h1>
    <aside>
      <header>
        <h2>Автор: Вася Пупкин</h2>
        <p><a href="">Связаться с ним!</a></p>
      </header>
      <p>Эксперт в HTML5</p>
    </aside>
  </header>
  <p><ins>Руководство о HTML5 должно было быть здесь, но оно оказалось, что Вася не был экспертом по HTML5.</ins></p>
</article>
```

Элемент <header> может содержать только <header> или <footer>, если они сами находятся внутри <article>, <aside>, <nav> или <section>.

Элемент <footer>

Категории контента: потоковое содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны глобальные атрибуты.

Элемент <footer> представляет нижний колонтитул для его ближайшего предка элемента <main>, или элемента из категории секционного содержимого или корневого секционного элемента.

Обычно содержит информацию об авторе статьи, данные о копирайте и т.д. Если используется как колонтитул всей страницы, содержимое дополняется сведениями об авторских правах, ссылками на условия использования, контактную информацию, ссылками на связанное содержимое и т.п.

```
...
  <footer>
    <nav>
      <section>
        <h2>Статьи</h2>
        <p><img src="images/..."> Олимпиада по инфокоммуникационным технологиям. <a href="articles/somersaults/1">Отборочный этап</a> · <a href="articles/somersaults/2">Заключительный этам</a></p>
        <p><img src="images/..."> Блины в буфете закончились. Что вы можете с этим сделать? <a href="articles/crisps/1">Читать далее...</a></p>
      </section>
      <ul>
        <li> <a href="/about">О нас...</a>
        <li> <a href="/feedback">Связаться с нами!</a>
        <li> <a href="/sitemap">Карта сайта</a>
      </ul>
    </nav>
    <p><small>© 2021 ИКТ — <a href="/ikt">Много всего</a></small></p>
  </footer>
</body>
```

В одном веб-документе может быть несколько элементов <footer>. Как каждая страница, так и каждая статья может иметь свой элемент <footer>. Также, <footer> можно поместить в элемент <blockquote>, чтобы указать источник цитирования.

HTML-элементы для группировки содержимого веб-страниц

Элементы для группировки содержимого <p>, <hr>, <pre>, <blockquote>, <ol>, <ul>, <li>, <dl>, <dt>, <dd> описаны в лабораторной работе №1.

Элемент <address>

Категории контента: потоковое содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны глобальные атрибуты.

Элемент <address> представляет контактную информацию о человеке или организации. Он должен включать физическое и/или цифровое местоположение/контактную информацию и средства идентификации лица (лиц) или организации, к которой относится эта информация. В браузере обычно отображается курсивом.

Например, твиттер-аккаунт W3C:

```
<address>
<p>W3C в Twitter:
<p><a href="https://twitter.com/w3c">@w3c</a>
</address>
```

Адрес, телефон и факс организации:

```
<address>
Учреждение образования "Белорусский государственный университет информатики и радиоэлектроники"<br>
Республика Беларусь, Минск
220013, ул. П. Бровки, 6<br>
Тел.: +375 17 379 32 35 | Факс: +375 17 270 20 33
</address>
```

Элемент <figure>

Категории контента: потоковое содержимое, корневой секционный, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Элемент <figure> представляет автономное содержимое (необязательно с подписью), являющееся самостоятельным элементом основного потока. С помощью элемента <figure> можно добавлять краткие характеристики к иллюстрациям, фотографиям, диаграммам, фрагментам кода и т.д.

```
<figure>
    <img src="picture.jpg" alt="Осень">
    <figcaption>Осенний лес</figcaption>
</figure>
```

Для элемента доступны глобальные атрибуты.

Элемент <figcaption>

Категории контента: отсутствуют.

Контекст, в котором этот элемент может быть использован: как потомок элемента <figure>.

Пропуск тегов: ни один из тегов не может быть пропущен.

Элемент <figcaption> представляет заголовок или легенду для остального содержимого родительского элемента <figure>.

Для элемента доступны глобальные атрибуты.

Элемент <main>

Категории контента: потоковое содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое; не может быть потомком таких элементов как <article>, <aside>, <footer>, <header> или <nav>.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны глобальные атрибуты.

Элемент <main> включает основное содержимое элемента <body> документа или приложения и исключает содержимое, которое повторяется на страницах сайта/приложения, таких как ссылки для навигации по сайту, информация об авторских правах, логотипы сайта и баннеры, а также поисковые формы (за исключением случаев, когда основной функцией документа или приложения является поисковая форма).

Не является секционным содержимым, поэтому не оказывает никакого влияния на структуру документа.

В документе должно быть не более одного элемента <main>.

```
<body>
<header>
  <h1>Пудель</h1>
    <nav>
      <ul>
        <li><a href="index.html">Главная</a></li>
        <li><a href="about.html">О породе</a></li>
        <li><a href="health.html">Здоровье</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <section>
      <header>
        <h2>О породе</h2>
          <nav>
            <ul>
              <li><a href="#basic">Разновидности</a></li>
              <li><a href="#app">Внешний вид</a></li>
              <li><a href="#temp">Характер</a></li>
            </ul>
          </nav>
      </header>
      <section id="basic">
        <h3>Разновидности</h3>
          <p>...</p>
      </section>
      <section id="app">
        <h3>Внешний вид</h3>
          <p>...</p>
      </section>
      <section id="temp">
        <h3>Характер</h3>
          <p>...</p>
      </section>
      <footer>
        <a href="#basic">Разновидности</a>
        <a href="#app">Внешний вид</a>
        <a href="#temp">Характер</a>
      </footer>
      </section>
    </main>
    <footer>
      <small>Copyright © <time datetime="2021">2021</time> Моя собака-улыбака.бу</small>
    </footer>
</body>
```

Элемент

Категории контента: потоковое содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается потоковое содержимое; как дочерний элемент <dl>.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны глобальные атрибуты.

Элемент сам по себе ничего не значит. Он представляет свои дочерние элементы. Может использоваться с атрибутами class, lang и title для разметки семантики, общей для группы последовательных элементов.

Рекомендуется использовать элемент в случаях, когда другой элемент не подходит. Использование более подходящих элементов вместо элемента обеспечивает лучшую доступность для читателей и облегчает обслуживание кода.

С другой стороны, элемент может быть полезен для стилистических целей или для обертывания нескольких абзацев внутри раздела, имеющих общие свойства.

HTML-элементы для текстового содержимого

Основные элементы для текстового содержимого описаны в лабораторной работе №1.

Элемент <s>

Категории контента: потоковое содержимое, текстовое содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается текстовое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны глобальные атрибуты.

Элемент <s> представляет содержимое, которое больше не является точным или актуальным. В этом примере рекомендованная розничная цена была помечена как неактуальная, поскольку у рассматриваемого продукта новая продажная цена.

```
<p>Купите наш чай со льдом и лимонад!</p>
<p><s>Рекомендованная розничная цена: 2,0 руб. за бутылку</s></p>
<p><strong>Успей купить сегодня всего за 2,50 руб. за бутылку!</strong></p>
```

Купите наш чай со льдом и лимонад!

~~Рекомендованная розничная цена: 2,0 руб. за бутылку~~

Успей купить сегодня всего за 2,50 руб. за бутылку!

Элемент <data>

Категории контента: потоковое содержимое, текстовое содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается текстовое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны глобальные атрибуты, а также обязательный атрибут value.

Элемент <data> представляет свое содержимое вместе с машиночитаемой формой этого содержимого в атрибуте value. Значение атрибута value должно быть представлением содержимого элемента в машиночитаемом формате.

```
<p>Новые поступления:</p>
<ul>
    <li><data value="44678">Альбом «На память о дне пятидесятилетия Эммануила Людвиговича Нобеля. 10 июня 1909 г.»</data></li>
    <li><data value="44668">Ноты. Травиата. Опера Джузеппе Верди</data></li>
    <li><data value="44688">Обух Н.К. Первый снег. Павловск. 1910-е годы</data></li>
</ul>
```

Элемент <data> может быть использован для следующих целей:

- в сочетании с микроформатами или микроданными для улучшения результатов выдачи в поисковых системах, за счет более точного описания содержимого элемента;
- в сочетании со сценариями на странице для хранения и обработки данных.

Элемент <time>

Категории контента: потоковое содержимое, текстовое содержимое, видимое содержимое.

Контекст, в котором этот элемент может быть использован: где ожидается текстовое содержимое.

Пропуск тегов: ни один из тегов не может быть пропущен.

Для элемента доступны глобальные атрибуты, а также необязательный атрибут datetime.

Элемент <time> представляет свое содержимое вместе с машиночитаемой формой этого содержимого в атрибуте datetime. Тип содержимого ограничен различными типами дат, времени, смещений часовых поясов и продолжительности. Содержимое элемента <time> должно быть валидным значением атрибута datetime:

1. Валидный месяц.
2. Валидная дата.
3-4. Валидное время.

5. Валидная дата и время.
6-7. Валидная временная зона.
8. Валидная неделя.
9. Валидная глобальная дата и время.

```
<time>2020-05</time>
<time>2020-05-05</time>
<time>09:54</time>
<time>09:54:39</time>
<time>2011-11-18 14:54</time>
<time>+0000</time>
<time>+00:00</time>
<time>2020-W47</time>
<time>2011-11-18T14:54+0000</time>
```

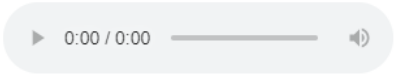
Если атрибут datetime присутствует, то содержимым элемента <time> может быть произвольный текст.

```
<div class="vevent">
    <a class="url" href="https://www.bsuir.by/ru/fik/olympiada-ict">https://www.bsuir.by/ru/fik/olympiada-ict</a>
    <span class="summary">Олимпиада по инфокоммуникационным технологиям</span>:
    <time class="dtstart" datetime="2021-05-05">5</time> -
    <time class="dtend" datetime="2005-05-07">7 мая</time>,
    at the <span class="location">Кафедра ИКТ, БГУИР</span>
</div>
```

HTML5-аудио

HTML5-аудио предоставляет улучшенные возможности работы с аудио контентом. До недавнего времени единственным способом добавления звуковых файлов на веб-страницы было интегрирование фонового звука с помощью тега <bgsound>, который проигрывался во время просмотра пользователем страницы без возможности выключения.

```
<audio src="name.ogg" controls></audio>
```



Благодаря добавлению в спецификацию HTML5 нового элемента <audio>, появилась возможность добавлять аудио содержимое со встроенным программным интерфейсом без привлечения подключаемых модулей.

Элемент <audio>

HTML5-элемент <audio> используется для внедрения звукового контента в веб-страницы.

В общем виде HTML-разметка имеет следующий вид:

Атрибут controls добавляет отображение браузерами интерфейса управления аудио плеера – кнопки воспроизведения, паузы, громкости.

В настоящий момент не существует аудио формата, который бы работал во всех браузерах, поэтому для обеспечения доступности контента максимально широкой аудитории рекомендуется включать несколько источников звука, представленных с использованием атрибута src элемента <source>. Одновременно можно добавить резервный контент для браузеров, которые не поддерживают элемент <audio>.

Атрибут	Описание, принимаемое значение
autoplay	Автоматическое воспроизведение аудио файла сразу же после загрузки страницы.
controls	Указывает браузеру, что нужно отобразить базовые элементы управления воспроизведением (начинать и останавливать

	воспроизведение, переходить в другое место записи, регулировать громкость).
loop	Циклическое воспроизведение аудио файла.
muted	Выключает звук при воспроизведении аудио файла.
preload	Атрибут, отвечающий за предварительную загрузку аудио контента. Не является обязательным, некоторые браузеры игнорируют его. Возможные значения: auto – браузер загружает аудио файл полностью, чтобы он был доступен, когда пользователь начнет его воспроизведение. metadata – браузер загружает первую небольшую часть аудио файла, чтобы определить его основные характеристики. none – отсутствие автоматической загрузки аудио файла.
src	Содержит абсолютный или относительный URL-адрес аудио файла.

Аудио кодек (декодер) представляет собой программу для преобразования цифровых данных в формат звукового файла или звукового потока. Популярными аудио форматами являются следующие:

- MP3 – самый популярный аудио формат, использующий сжатие с потерями и позволяющий уменьшить размер файла в несколько раз.
- AAC (Advanced Audio Codec) – закрытый кодек, аналог MP3, но по сравнению с последним, поддерживает более высокое качество звука при сходном размере файла.
- Ogg Vorbis – бесплатный формат с открытым кодом. Обеспечивает хорошее качество звука, но недостаточно широко поддерживается аппаратными проигрывателями.

Элемент <source> используется для добавления нескольких медиаресурсов для <audio> и <video>. Указывает на альтернативные видео/аудио файлы, которые браузер может выбрать из предложенных на основании поддерживаемого им типа носителя или кодека.

Атрибут	Описание, принимаемое значение
media	Определяет тип медиаустройства (т.е. для каких устройств оптимизирован файл).
src	Содержит абсолютный или относительный URL-адрес медиафайла.
type	Определяет MIME-тип медиафайла.

Элемент <track> используется в качестве дочернего элемента <audio> и <video>. Добавляет текстовую дорожку для субтитров, заголовков медиафайлов или другой текстовой информации, которая должна быть видна во время воспроизведения аудио или видеофайла.

Атрибут	Описание, принимаемое значение
default	Указывает, что данная дорожка воспроизводится по умолчанию. Только один элемент <track> может содержать данный атрибут.
kind	Указывает тип текстовой дорожки, по умолчанию выводятся субтитры (subtitles). Принимаемые значения: captions – перевод диалогов и звуковых эффектов, отображаемый в виде текста поверх видео (для глухих пользователей).

	chapters – добавляет названия глав в виде списка для навигации по медиафайлу. descriptions – добавляет звуковое описание происходящего в видео (для слепых пользователей). metadata – метаданные, используемые скриптами, не отображаются для пользователей. subtitles – текстовое дублирование звуковой дорожки видео, отображается в виде субтитров к видео.
label	Добавляет название дорожки. Если этот атрибут не задан, браузер применит значение по умолчанию.
src	Содержит абсолютный или относительный URL-адрес данных текстовой дорожки.
srclang	Язык воспроизводимой дорожки.

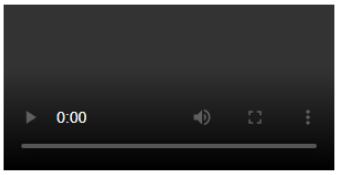
HTML5-видео

HTML5-видео – новый стандарт для размещения мультимедийных файлов в сети с оригинальным программным интерфейсом без привлечения подключаемых модулей.

Элемент <video>

С помощью элемента <video> появилась возможность добавлять видеосодержимое на веб-страницы, а также стилизовать внешний вид видеоплеера при помощи css-стилей.

```
<video src="video.ogv" controls></video>
```



Атрибут controls отвечает за появление элементов управления видеоплеером. Вы можете добавить изображение с помощью атрибута poster, которое браузер будет использовать, пока загружается видео или пока пользователь не нажмет на кнопку воспроизведения, а также задать высоту и ширину видео.

Как и в случае с аудиофайлами, рекомендуется перечислять в <source> все форматы, начиная с более предпочтительного. Также нужно указывать MIME-тип для каждого видеофайла.

```
<video controls width="400" height="300">
  <source src="video.mp4" type="video/mp4"><!-- MP4 для Safari, IE9, iPhone, iPad, Android, и Windows Phone 7 -->
  <source src="video.webm" type="video/webm"><!-- WebM/VP8 для Firefox4, Opera, и Chrome -->
  <source src="video.ogv" type="video/ogg"><!-- Ogg/Vorbis для старых версий браузеров Firefox и Opera -->
  <object data="video.swf" type="application/x-shockwave-flash"><!-- добавляем видеоконтент для устаревших браузеров, в которых нет поддержки элемента video -->
    <param name="movie" value="video.swf">
  </object>
</video>
```

Атрибут	Описание, принимаемое значение
autoplay	Автоматическое воспроизведение видеоофайла сразу же после загрузки страницы.
controls	Указывает браузеру, что нужно отобразить базовые элементы управления воспроизведением (воспроизведение, пауза, громкость).
height	Задает высоту окна для отображения видеоданных, возможные значения: px или %
loop	Циклическое воспроизведение видеофайла.

muted	Выключает звук при воспроизведении видеофайла.
poster	URL файла изображения, которое будет отображаться во время загрузки видеофайла или до тех пор, пока пользователь не нажмет на кнопку PLAY. Если атрибут не задан, то будет отображаться первый кадр видеофайла.
preload	Атрибут, отвечающий за предварительную загрузку видеоконтента. Не является обязательным, некоторые браузеры игнорируют его. Возможные значения: auto – браузер загружает видеофайл полностью, чтобы он был доступен, когда пользователь начнет его воспроизведение. metadata – браузер загружает первую небольшую часть видеофайла, чтобы определить его основные характеристики. none – отсутствие автоматической загрузки видеофайла.
src	Содержит абсолютный или относительный URL-адрес видеофайла.
width	Задает ширину окна для отображения видеоданных, возможные значения: px или %

Видеокодеки

При просмотре видео проигрыватель должен его декодировать. Одни проигрыватели используют программное декодирование видеопотока, другие используют аппаратное декодирование.

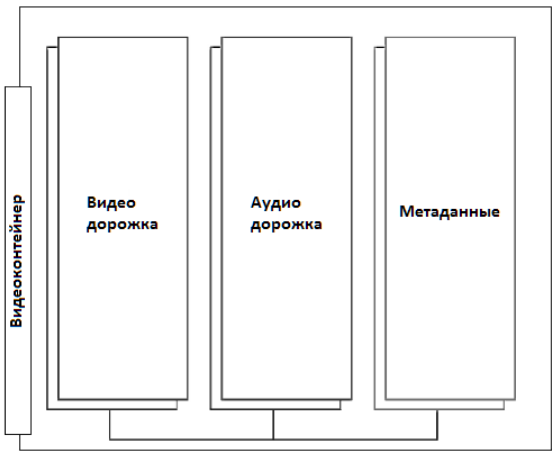
Поскольку каждый браузер поддерживает определенный кодек, поэтому, чтобы обеспечить воспроизведение видео-контента во всех браузерах, видеофайл нужно размещать в нескольких форматах.

H.264 – высококачественный кодек от фирмы MPEG, делится на профили для поддержки как устройств с минимальными возможностями, так и устройств высокого разрешения.

Ogg Theora – открытый бесплатный стандарт для видео, качество и производительность несколько ниже стандарта H.264.

VP8 – открытый бесплатный кодек, сходный по качеству с H.264. Поддерживается в Firefox, Chrome и Opera.

Видеоконтейнеры



Любой видеофайл является файловым контейнером, в котором хранятся другие файлы. Аудио- и видеодорожки объединяются для воспроизведения видеоролика.

Метаданные содержат информацию о данном видеоролике – изображение обложки, субтитры и пр. К популярным форматам видеоконтейнеров относятся следующие:

- Ogg (.ogv, .oga, .ogx, .ogg) – формат-контейнер с открытым исходным кодом для видеокодека Theora и аудио Vorbis. Работает в Firefox, Chrome и Opera. MIME-тип: video/ogg.
- MPEG 4 (.mp4) – формат-контейнер для видеокодека H.264 и аудиокодека AAC. Работает в Safari и Chrome. Кодирует видео, в том числе высокой четкости, для полного спектра устройств, таких как iPhone, iPod и iPad. MIME-тип: video/mp4.
- WebM (.webm) – формат-контейнер с открытым исходным кодом для видеокодека VP8 от Google и аудиокодека Ogg Vorbis. Работает в Firefox, Chrome, Opera. MIME-тип: video/webm.
- Audio Video Interleave (.avi) – формат предназначен для записи звука и движущихся изображений, соответствует спецификации RIFF. MIME-тип: video/vnd.avi, video/avi, video/msvideo, video/x-msvideo.
- Matroska (.mkv) – популярный видеоконтейнер, может содержать видео в формате H.264, VP8 или Theora. MIME-тип: video/x-matroska, audio/x-matroska.

На данный момент браузеры поддерживают три основных видео формата:

Формат	Видеокодек	Аудиокодек
.mp4	H.264	AAC
.ogg/.ogv	Theora	Vorbis
.webm	VP8	Vorbis

Видео в формате **.avi** на сайте средствами HTML5 не воспроизводится. Поэтому его необходимо перекодировать в эти три формата с соответствующими видео и аудиокодеками для вывода на сайте. Для этого можно использовать видеоконвертеры.

Альтернативные медиаресурсы

Элемент <source> используется для указания нескольких медиаресурсов для <audio> и <video>. Добавляет альтернативные видео/аудио файлы, которые браузер может выбрать из предложенных на основании поддерживаемого им типа носителя или кодека.

Атрибут	Описание, принимаемое значение
media	Определяет тип медиаустройства (т.е. для каких устройств оптимизирован файл).
src	Содержит абсолютный или относительный URL-адрес медиафайла.
type	Определяет MIME-тип медиафайла.

Элемент <track> используется в качестве дочернего элемента <audio> и <video>. Добавляет текстовую дорожку для субтитров, заголовков медиафайлов или другой текстовой информации, которая должна быть видна во время воспроизведения медиаресурса. Описание атрибутов приведено выше.

Размещение видео на сайте

Для вставки видео на сайт необходимо разместить код с помощью HTML5-разметки, используя атрибуты для задания видео требуемых параметров:

```
<video controls width="710" height="538" poster="/examples/media/martynko.png" preload="none">
  <source src="/examples/media/martynko.mp4" type="video/mp4">
</video>
```

Если вы хотите выровнять видеоплеер на странице, нужно обернуть элемент <video> в контейнер с присвоенным классом, для которого задаются ширина и высота, соответствующие размерам вашего видео. Далее, с помощью css-свойств можно задать отступы, выравнивание на странице и т.д.

HTML5-формы

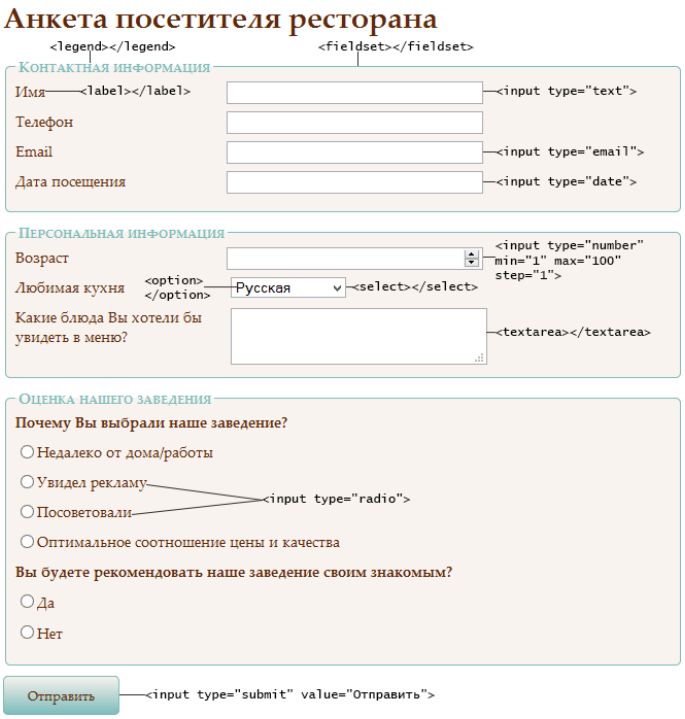
HTML-формы являются элементами управления, которые применяются для сбора информации от посетителей веб-сайта.

Веб-формы состоят из набора текстовых полей, кнопок, списков и других элементов управления, которые активизируются щелчком мыши. Технически формы передают данные от пользователя удаленному серверу.

Для получения и обработки данных форм используются языки веб-программирования, такие как **PHP**, **Perl**.

До появления HTML5 веб-формы представляли собой набор нескольких элементов <input type="text">, <input type="password">, завершающихся кнопкой <input type="submit">. Для стилизации форм в разных браузерах приходилось прилагать немало усилий. Кроме того, формы требовали применения JavaScript для проверки введенных данных, а также были лишены специфических типов полей ввода для указания повседневной информации типа дат, адресов электронной почты и URL-адресов.

HTML5-формы решили большинство этих распространенных проблем благодаря наличию новых атрибутов, предоставив возможность изменять внешний вид элементов форм за счет **CSS3**.



Элемент <form>

Основу любой формы составляет элемент <form>...</form>. Он не предусматривает ввод данных, так как является контейнером, удерживая вместе все элементы управления формы – поля. Атрибуты этого элемента содержат информацию, общую для всех полей формы, поэтому в одну форму нужно включать поля, объединенные логически.

Атрибут	Описание, принимаемое значение
accept-charset	Значение атрибута представляет собой разделенный пробелами список кодировок символов, которые будут

	использоваться для отправки формы, например, <form accept-charset="ISO-8859-1">.
action	Обязательный атрибут, который указывает url обработчика формы на сервере, которому передаются данные. Представляет из себя файл (например, action.php), в котором описано, что нужно делать с данными формы. Если значение атрибута не будет указано, то после перезагрузки страницы элементы формы примут значения по умолчанию. В случае, если вся работа будет выполняться на стороне клиента сценариями JavaScript, то для атрибута action можно указать значение #. Также можно сделать так, чтобы заполненная посетителем форма приходила вам на почту. Для этого нужно внести следующую запись: <form action="mailto:адрес вашей электронной почты" enctype="text/plain"></form>
autocomplete	Отвечает за запоминание введенных в текстовое поле значений и автоподстановку их при последующем вводе: on – означает, что поле не защищено, и его значение можно сохранять и извлекать, off – отключает автозаполнение для полей форм.
enctype	Используется для указания MIME-типа данных, отправляемых вместе с формой, например, enctype="multipart/form-data". Указывается только в случае method="post". application/x-www-form-urlencoded – тип содержимого по умолчанию, указывает на то, что передаваемые данные представляют список URL-кодированных переменных формы. Символы пробела (ASCII 32) будут закодированы как +, а специальный символ, например, такой как ! будет закодирован шестнадцатиричной форме как %21. multipart/form-data – используется для отправки форм, содержащих файлы, не-ASCII данные и бинарные данные, состоит из нескольких частей, каждая из которых представляет содержимое отдельного элемента формы. text/plain – указывает на то, что передается обычный (не html) текст.
method	Задает способ передачи данных формы. Метод get передает данные на сервер через адресную строку браузера. При формировании запроса к серверу все переменные и их значения формируют последовательность вида www.anysite.ru/form.php?var1=1&var2=2. Имена и значения переменных присоединяются к адресу сервера после знака ? и разделяются между собой знаком &. Все специальные символы и буквы, отличные от латинских, кодируются в формате %nn, пробел заменяется на +. Этот метод нужно использовать, если вы не передаете больших объемов информации. Если вместе с формой предполагается отправка какого-либо файла, этот метод не подойдет. Метод post применяется для пересылки данных больших объемов, а также конфиденциальной информации и паролей. Данные, отправляемые с помощью этого метода, не видны в заголовке URL, так как они содержатся в теле сообщения. <form action="action.php" enctype="multipart/form-data" method="post"></form>

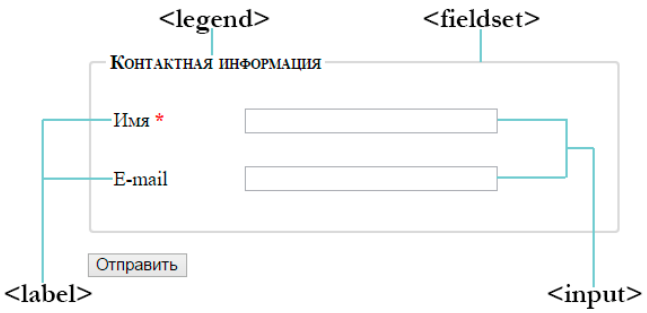
name	Задает имя формы, которое будет использоваться для доступа к элементам формы через сценарии, например, name="opros".
novalidate	Отключает проверку в кнопке для отправки формы. Атрибут используется без указания значения.
target	Указывает окно, в которое будет направлена информация: _blank – новое окно. _self – тот же фрейм. _parent – родительский фрейм (если он существует, если нет – то в текущий). _top – окно верхнего уровня по отношению к данному фрейму. Если вызов происходит не из дочернего фрейма, то в тот же фрейм.

Элемент <fieldset>...</fieldset>

Предназначен для группировки элементов, связанных друг с другом, разделяя таким образом форму на логические фрагменты.

Каждой группе элементов можно присвоить название с помощью элемента <legend>, который идет сразу за открывающим тегом элемента <fieldset>. Название группы проявляется слева в верхней границе <fieldset>. Например, если в элементе <fieldset> хранится контактная информация:

```
<form>
  <fieldset>
    <legend>Контактная информация</legend>
    <p><label for="name">Имя <em>*</em></label><input type="text" id="name"></p>
    <p><label for="email">E-mail</label><input type="email" id="email"></p>
  </fieldset>
<p><input type="submit" value="Отправить"></p>
</form>
```



Атрибут	Описание, принимаемое значение
disabled	Если атрибут присутствует, то группа связанных элементов формы, находящихся внутри контейнера <fieldset>, отключены для заполнения и редактирования. Используется для ограничения доступа к некоторым полям формы, содержащих ранее введенные данные. Атрибут используется без указания значения – <fieldset disabled>.
form	Значение атрибута должно быть равно атрибуту id элемента <form> в этом же документе. Указывает на одну или несколько форм, к которым принадлежит данная группа элементов. На данный момент атрибут не поддерживается ни одним браузером.
name	Определяет имя, которое будет использоваться для ссылки на элементы в JavaScript, или для ссылки на данные формы после заполнения и отправки формы. Является аналогом атрибута id.

Элемент <input>

Элемент <input> создает большинство полей формы. Атрибуты элемента отличаются в зависимости от типа поля, для создания которого используется этот элемент.

С помощью css-стилей можно изменить размер шрифта, тип шрифта, цвет и другие свойства текста, а также добавить границы, цвет фона и фоновое изображение. Ширина поля задается свойством width.

Атрибут	Описание, принимаемое значение
accept	Определяет тип файла, разрешенных для отправки на сервер. Указывается только для <input type="file">. Возможные значения: file_extension – разрешает загрузку файлов с указанным расширением, например, accept=".gif", accept=".pdf", accept=".doc". audio/* – разрешает загрузку аудиофайлов. video/* – разрешает загрузку видеофайлов. image/* – разрешает загрузку изображений. media_type — указывает на медиа-тип загружаемых файлов.
alt	Определяет альтернативный текст для изображений, указывается только для <input type="image">.
autocomplete	Отвечает за запоминание введенных в текстовое поле значений и автоподстановку их при последующем вводе: on – означает, что поле не защищено, и его значение можно сохранять и извлекать, off – отключает автозаполнение для полей форм.
autofocus	Позволяет сделать так, чтобы в загружаемой форме то или иное поле ввода уже имело фокус (было выбрано), являясь готовым к вводу значения.
checked	Атрибут проверяет, установлен ли флажок по умолчанию при загрузке страницы для полей типа type="checkbox" и type="radio".
disabled	Отключает возможность редактирования и копирования содержимого поля.
form	Значение атрибута должно быть равно атрибуту id элемента <form> в этом же документе. Определяет одну или несколько форм, которым принадлежит данное поле формы.
formaction	Задает url файла, который будет обрабатывать введенные в поля данные при отправке формы. Задается только для полей типа type="submit" и type="image". Атрибут переопределяет значение атрибута action самой формы.
formenctype	Определяет, как будут кодироваться данные полей формы при отправке на сервер. Переопределяет значение атрибута enctype формы. Задается только для полей типа type="submit" и type="image". Варианты: application/-x-www-form-urlencoded – значение по умолчанию. Все символы кодируются перед отправкой (пробелы заменяются на символ +, специальные символы преобразуются в значения ASCII HEX) multipart/form-data – символы не кодируются text/plain – пробелы заменяются на символ +, а специальные символы не кодируются.

formmethod	Атрибут определяет метод, который браузер будет использовать для отправки данных формы на сервер. Задается только для полей типа type="submit" и type="image". Переопределяет значение атрибута method формы. Варианты: get – значение по умолчанию. Данные из формы (пара имя/значение) добавляются в url-адрес и отправляются на сервер: URL?имя=значение&имя=значение. post – данные формы отправляются в виде http-запроса.
formnovalidate	Определяет, что данные полей формы не должны проверяться при отправке формы. Переопределяет значение атрибута novalidate формы. Можно использовать без указания значения атрибута.
formtarget	Определяет, где выводить ответ, полученный после отправки формы. Задается только для полей типа type="submit" и type="image". Переопределяет значение атрибута target формы. _blank – загружает ответ в новое окно/вкладку. _self – загружает ответ в то же окно (значение по умолчанию). _parent – загружает ответ в родительский фрейм. _top – загружает ответ во весь экран. framename – загружает ответ во фрейм с указанным именем.
height	Значение атрибута содержит количество пикселей без указания единицы измерения. Устанавливает высоту поля формы типа type="image", например, <input type="image" src="img_submit.gif" height="50">. Рекомендуется одновременно устанавливать как высоту, так и ширину поля.
list	Является ссылкой на элемент <datalist>, содержит его id. Позволяет предоставить пользователю несколько вариантов на выбор, когда он начинает вводить значение в соответствующем поле.
max	Позволяет ограничить допустимый ввод числовых данных максимальным значением, значение атрибута может содержать целое или дробное число. Рекомендуется использовать этот атрибут вместе с атрибутом min. Работает со следующими типами полей: number, range, date, datetime, datetime-local, month, time и week.
maxlength	Атрибут задает максимальное количество символов, вводимых в поле. Значение по умолчанию 524288 символов.
min	Позволяет ограничить допустимый ввод числовых данных минимальным значением.
multiple	Позволяет пользователю ввести несколько значений атрибутов, разделяя их запятой. Применяется в отношении файлов и адресов электронной почты. Указывается без значения атрибута.
name	Определяет имя, которое будет использоваться для доступа к элементу <form>, к примеру, в таблицах стилей css. Является аналогом атрибута id.
pattern	Позволяет определять с помощью регулярного выражения синтаксис данных, ввод которых должен быть разрешен в определенном поле. Например, pattern="[a-z]{3}-[0-9]{3}" – квадратные скобки устанавливают диапазон допустимых

	символов, в данном случае – любые строчные буквы, число в фигурных скобках указывает, что нужны три строчные буквы, после которых следует тире, далее – три цифры в диапазоне от 0 до 9.
placeholder	Содержит текст, который отображается в поле ввода до заполнения (чаще всего это подсказка).
readonly	Не позволяет пользователю изменять значения элементов формы, выделение и копирование текста при этом доступно. Указывается без значения атрибута.
required	Выводит сообщение о том, что данное поле является обязательным для заполнения. Если пользователь попытается отправить форму, не введя в это поле требуемое значение, то на экране отобразится предупреждающее сообщение. Указывается без значения атрибута.
size	Задает видимую ширину поля в символах. Значение по умолчанию – 20. Работает со следующими типами полей: text, search, tel, url, email и password.
src	Задает url изображения, используемого в качестве кнопки отправки данных формы. Указывается только для поля <input type="image">.
step	Используется для элементов, предполагающих ввод числовых значений, указывает величину увеличения или уменьшения значений в процессе регулировки диапазона (шаг).
type	button – создает кнопку.
	checkbox – превращает поле ввода во флажок, который можно установить или очистить,
	color – генерирует палитры цветов в поддерживающих браузерах, давая пользователям возможность выбирать значения цветов в шестнадцатеричном формате.
	date – позволяет вводить дату в формате дд.мм.гггг.
	datetime-local – позволяет вводить дату и время, разделенные прописной английской буквой T по шаблону дд.мм.гггг чч:мм.
	email – браузеры, поддерживающие данный атрибут, будут ожидать, что пользователь введет данные, соответствующие синтаксису адресов электронной почты.
	file – позволяет загружать файлы с компьютера пользователя.
	hidden – скрывает элемент управления, который не отображается браузером и не дает пользователю изменять значения по умолчанию.
	image – создает кнопку, позволяя вместо текста на кнопке вставить изображение.
	month – позволяет пользователю вводить год и номер месяца по шаблону гггг-мм.
	number – предназначено для ввода целочисленных значений. Его атрибуты min, max и step задают верхний, нижний пределы и шаг между значениями соответственно. Эти атрибуты предполагаются у всех элементов, имеющих численные

	показатели. Их значения по умолчанию зависят от типа элемента. password – создает текстовые поля в форме, при этом вводимые пользователем символы заменяются на звездочки, маркеры, либо другие, установленные браузером значки. radio – создает переключатель – элемент управления в виде небольшого кружка, который можно включить или выключить. range – позволит создать такой элемент интерфейса, как ползунок, min / max – позволят установить диапазон выбора. reset – создает кнопку, которая очищает поля формы от введенных пользователем данных. search – обозначает поле поиска, по умолчанию поле ввода имеет прямоугольную форму. submit – создает стандартную кнопку, активизируемую щелчком мыши. Кнопка собирает информацию с формы и отправляет ее для обработки. text – создает текстовые поля в форме, выводя однострочное текстовое поле для ввода текста. time – позволяет вводить время в 24-часовом формате по шаблону чч:мм. В поддерживающих браузерах оно отображается как элемент управления в виде числового поля ввода со значением, изменяемым с помощью мыши, и допускает ввод только значений времени. url – поле предназначено для указания URL-адресов. week – соответствующий инструмент-указатель позволяет пользователю выбрать одну неделю в году, после чего обеспечит ввод данных в формате нн-гггг. В зависимости от года число недель может быть 52 или 53.
value	Определяет текст, отображаемый на кнопке, в поле или связанный текст. Не указывается для полей типа file.
width	Значение атрибута содержит количество пикселей. Позволяет задать ширину полей формы.

Элемент <textarea>...</textarea>

Элемент <textarea>...</textarea> используется вместо элемента <input type="text">, когда нужно создать большие текстовые поля. Текст, отображаемый как исходное значение, помещается внутрь.

Размеры поля устанавливаются при помощи атрибутов cols – размеры по горизонтали, rows – размеры по вертикали. Высоту поля можно задать свойством height. Все размеры считаются исходя из размера одного символа моноширинного шрифта.

Атрибут	Описание, принимаемое значение
autofocus	Устанавливает фокус на нужном начальном текстовом поле автоматически.
cols	Устанавливает ширину через количество символов. Если пользователь вводит больше текста, появляется полоса прокрутки.
disabled	Отключает возможность редактирования и копирования содержимого поля.

form	Значение атрибута должно быть равно значению атрибута id элемента <form> в этом же документе. Определяет одну или несколько форм, которым принадлежит данное текстовое поле.
maxlength	Значение атрибута задает максимальное число символов для ввода в поле.
name	Задает имя текстового поля.
placeholder	Определяет короткую текстовую подсказку, которая описывает ожидаемое вводимое значение.
readonly	Отключает возможность редактирования содержимого поля.
required	Выводит сообщение о том, что данное поле является обязательным для заполнения.
rows	Указывает число, которое означает, сколько строк должно отображаться в текстовой области.
wrap	Определяет, должен ли текст сохранять переносы строк при отправке формы. Значение hard сохраняет перенос, а значение soft не сохраняет. Если используется значение hard, то должно указываться значение атрибута cols.

Элемент <select>...</select>

Списки дают возможность расположить большое количество пунктов компактно. Раскрывающиеся списки создаются при помощи элемента <select>...</select>. Они позволяют выбрать одно или несколько значений из предложенного множества. По умолчанию в поле списка отображается его первый элемент.

Атрибут	Описание, принимаемое значение
autofocus	Устанавливает автоматический фокус на элементе при загрузке страницы.
disabled	Отключает раскрывающийся список.
form	Определяет форму, которой принадлежит данный список. В качестве значения атрибута указывается идентификатор формы.
multiple	Дает возможность выбора одного или нескольких пунктов, для этого при выборе нужно нажать и удерживать нажатой клавишу Ctrl.
name	Определяет имя для выпадающего списка. Значение атрибута содержит название, отражающее тематику списка.
required	Выводит сообщение о том, что пользователь должен выбрать значение из раскрывающегося списка перед отправкой формы.
size	Задает количество одновременно видимых на экране элементов списка. Если количество элементов списка превышает установленное количество, появляется полоса прокрутки. Значение атрибута задается целым положительным числом.

Элемент <option>...</option>

Для добавления в список пунктов используются элементы <option>...</option>, которые располагаются внутри <select>.

Атрибут	Описание, принимаемое значение
disabled	Делает недоступным для выбора элемент списка.
label	Задает укороченную версию для элемента, которая будет отражаться в выпадающем списке. Значение атрибута содержит текст, описывающий соответствующий пункт выпадающего списка.
selected	Отображает выбранный элемент списка по умолчанию при загрузке страницы браузером.
value	Указывает значение, которое будет отправлено на сервер при отправке формы.

Элемент <optgroup>...</optgroup>

Для систематизации списков применяется элемент <optgroup>...</optgroup>, который создает заголовки в списках.

Для списков возможно изменить размер шрифта, тип шрифта, цвет и другие свойства текста, а также добавить границы, цвет фона и фоновое изображение.

Атрибут	Описание, принимаемое значение
disabled	Отключает данную группу элементов списка для выбора.
label	Задает заголовок для группы элементов выпадающего списка. Значение атрибута содержит текст, недоступный для выбора, который будет располагаться над соответствующими пунктами списка. Текст выделяется в браузере жирным начертанием.

Элемент <label>...</label>

Надписи к элементам формы создаются с помощью элемента <label>...</label>. Существует два способа группировки надписи и поля. Если поле находится внутри элемента <label>, то атрибут for указывать не нужно.

```
<!-- с указанием атрибута for -->
<label for="comments">Когда вы последний раз летали на самолете?</label>
<textarea id="comments"></textarea>

<!-- без атрибута for -->
<label>Кошка<input id="cat" type="checkbox"></label></p>
```

Атрибут	Описание, принимаемое значение
for	Определяет, к какому полю формы привязан данный элемент. Можно создавать поясняющие надписи к следующим элементам формы: <input>, <textarea>, <select>. Значение атрибута содержит идентификатор поля формы.

Элемент <button>...</button>

Элемент <button>...</button> создает кликабельные кнопки. В отличие от кнопок, созданных <input> (<input type="submit"></input>, <input type="image">, <input type="reset">, <input type="button">), внутрь элемента <button> можно поместить контент – текст или изображение.

Для корректного отображения элемента <button> разными браузерами нужно указывать атрибут type, например, <button type="submit"></button>.

Кнопки позволяют пользователям передавать данные в форму, очищать содержимое формы или предпринимать какие-либо другие действия. Можно создавать границы, изменять фон и выравнивать текст на кнопке.

Атрибут	Описание, принимаемое значение
autofocus	Устанавливает фокус на кнопке при загрузке страницы.
disabled	Отключает кнопку, делая ее некликабельной.
form	Указывает на одну или несколько форм, которым принадлежит данная кнопка. Значение атрибута – идентификатор соответствующей формы.
formaction	Значение атрибута содержит url-адрес обработчика данных формы, отправляемых при нажатии на кнопку. Только для кнопки типа type="submit". Переопределяет значение атрибута action, указанного для элемента <form>.
formenctype	Задает тип кодировки данных формы перед отправкой на сервер при нажатии на кнопки типа type="submit". Переопределяет значение атрибута enctype, указанного для элемента <form>. Возможные значения: application/x-www-form-urlencoded – значение по умолчанию. Все символы перед отправкой будут закодированы. multipart/form-data – символы не кодируются. Используется в случае, когда с помощью формы загружаются файлы. text/plain – символы не кодируются, а пробелы заменяются на символ +.
formmethod	Атрибут определяет метод, который браузер будет использовать для отправки формы. Переопределяет значение атрибута method, указанного для элемента <form>. Указывается только для кнопок типа type="submit". Возможные значения: get – данные из формы (пара имя/значение) добавляются в url-адрес и отправляются на сервер. Данный способ имеет ограничения по размеру отправляемых данных и не подходит для отправки паролей и конфиденциальной информации. post – данные из формы добавляются в виде http-запроса. Метод является более надежным и безопасным, чем get и не имеет ограничений по размеру.
formnovalidate	Атрибут задает, что данные формы не должны проверяться при отправке. Указывается только для кнопок типа type="submit".
formtarget	Атрибут задает, в каком окне выводить результат после отправки формы. Указывается только для кнопок типа type="submit". Переопределяет значение атрибута target, указанного для элемента <form>. _blank – загружает ответ в новое окно/вкладку. _self – загружает ответ в то же окно (значение по умолчанию). _parent – загружает ответ в родительский фрейм. _top – загружает ответ во весь экран. framename – загружает ответ во фрейм с указанным именем.
name	Задает имя кнопки, значение атрибута – текст. Используется для ссылки на данные формы, после того как форма была отправлена, или для ссылки на данную кнопку (кнопки) в JavaScript.
type	Определяет тип кнопки. Возможные значения:

	button – кликабельная кнопка. reset – кнопка сброса, возвращает первоначальное значение. submit — кнопка для отправки данных формы.
value	Задает значение по умолчанию, отправляемое при нажатии на кнопку.

Элементы <input type="checkbox">, <input type="radio">

Флажки в формах задаются с помощью конструкции <input type="checkbox">, а переключатель – при помощи <input type="radio">.

Флажков, в отличие от переключателей, в одной форме может быть установлено несколько. Если для флажков указан атрибут checked, то при загрузке станицы на соответствующих полях формы флажки уже будут установлены.

Элемент <label> применяется при реализации выбора с помощью переключателей и флажков. Можно выбрать нужный пункт, просто щелкая кнопкой мыши на тексте, связанном с ним. Для этого нужно поместить <input> внутрь элемента <label>.

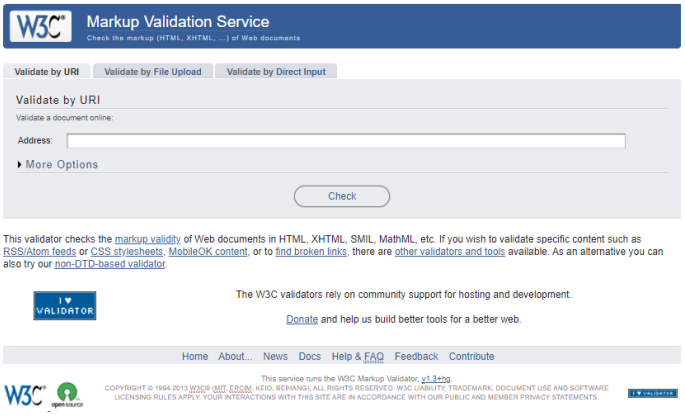
Валидация HTML

Валидацией называется проверка документа на соответствие веб-стандартам и выявление существующих ошибок. Соответственно, валидным является такой веб-документ, который прошел подобную процедуру и не имеет замечаний по коду.

Валидный HTML-код, валидная разметка – это HTML-код, который написан в соответствии с определенными стандартами. Их разработал Консорциум Всемирной Паутины – World Wide Web Consortium (W3C).

HTML-валидатор производит несколько проверок Вашего кода. Основные из них:

- Валидация синтаксиса – проверка на наличие синтаксических ошибок.
- Проверка вложенности тэгов – тэги должны быть закрыты в обратном порядке относительно их открытия. Например, эта проверка отлавливает ошибки с неправильно закрытыми .
- Валидация DTD – проверка соответствия Вашего кода указанному Document Type Definition. Она включает проверку названий тэгов, атрибутов, и «встраивания» тэгов (тэги одного типа внутри тэгов другого типа).
- Проверка на посторонние элементы – проверка выявляет все, что есть в коде, но отсутствует в DTD. Например, пользовательские тэги и атрибуты.



Можно проверять код по ссылке, можно загрузить в сервис HTML-файл, а можно фрагмент кода.

Можно задать дополнительные параметры валидации. Они нужны, чтобы структурировать и детализировать результаты.

Результаты валидации делятся на предупреждения и ошибки. Разница в критичности. Ошибки с максимальной вероятностью могут создать проблемы в работе кода. К предупреждениям относятся неточности, которые с минимальной вероятностью навредят работе страницы, но не соответствуют стандартам. Это избыточный код, бессмысленные элементы и др.

Что такое Git?

Git – это консольная утилита, для отслеживания и ведения истории изменения файлов, в проекте. Чаще всего его используют для кода, но можно и для других файлов. Например, для изображений – полезно для дизайнеров.

С помощью Git можно откатить свой проект до более старой версии, сравнивать, анализировать или заливать свои изменения в репозиторий.

Репозиторием называют хранилище кода и историю его изменений. Git работает локально и все репозитории хранятся в определенных папках на жестком диске.

Каждая точка сохранения проекта носит название **коммит (commit)**. У каждого commit есть hash (уникальный идентификатор) и комментарий. Из таких commit собирается ветка. **Ветка** – это история изменений. У каждой ветки есть свое название. Репозиторий может содержать в себе несколько веток, которые создаются из других веток или вливаются в них.

Что такое Github и в чем отличие от Git?

Git-репозиторий, загруженный на GitHub, доступен с помощью интерфейса командной строки Git и Git-команд. Также есть и другие функции: документация, запросы на принятие изменений (pull requests), история коммитов, интеграция со множеством популярных сервисов, email-уведомления, эмодзи, графики, вложенные списки задач и т.д.

Кроме GitHub есть другие сервисы, которые используют Git, – например, Bitbucket и GitLab. Можно разместить Git-репозиторий на любом из них. В рамках учебной программы мы будем работать с Github.

Регистрация на Github и его установка

Скачайте систему контроля версий Git. Ссылки на скачивание:

- для ОС Windows.
- для ОС MacOS.

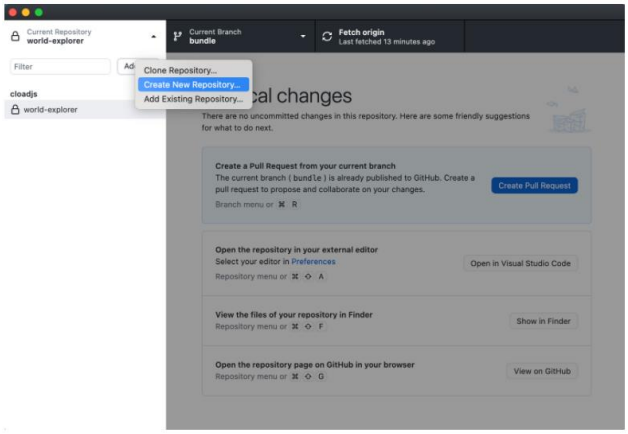
На странице https://github.com/ нажмите на кнопку "Sign up" и пройдите все шаги регистрации. Если вы уже зарегистрированы, достаточно войти в свой аккаунт.

Чтобы упростить работу с Git необходимо загрузить и установить визуальный инструмент Github Desktop.

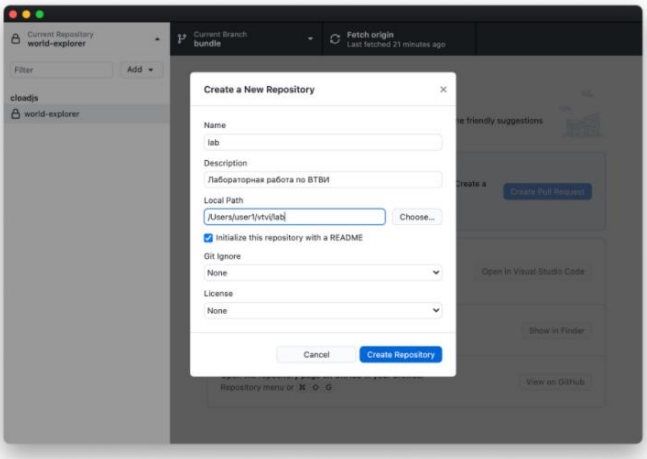
Первый коммит

1. Откройте Github Desktop и войдите под своей учетной записью Github.

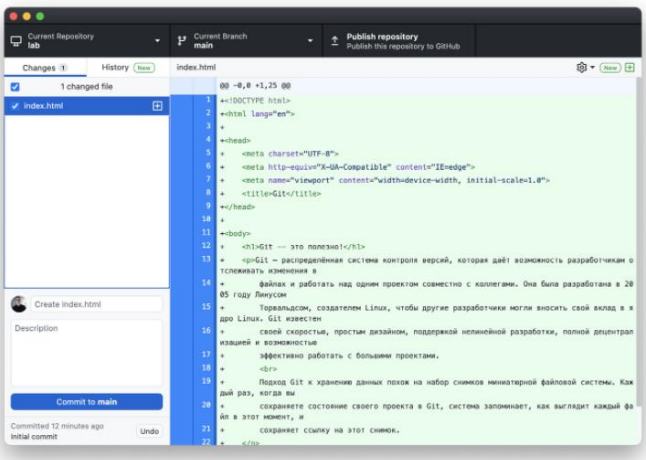
2. Для создания нового репозитория нажмите на кнопку Add → Create New Repository...



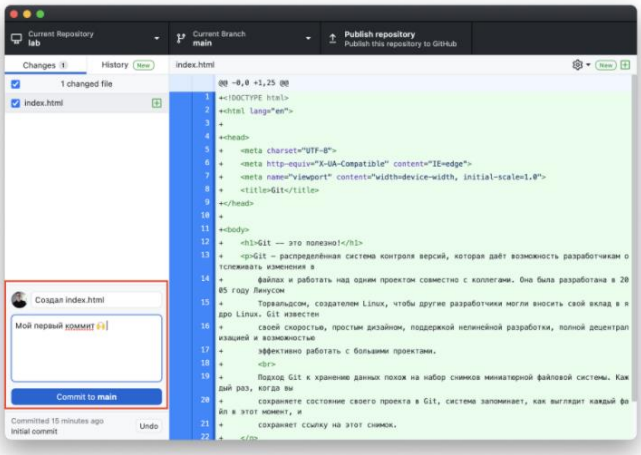
3. В появившемся окне введите название репозитория, описание (опционально) и путь к папке (она создастся автоматически, если ее не существует; можно также указать путь к папке, в которой уже лежат файлы для добавления в репозиторий). Активируйте чекбокс для добавления в репозиторий README файла. Нажмите на кнопку "Create Repository".



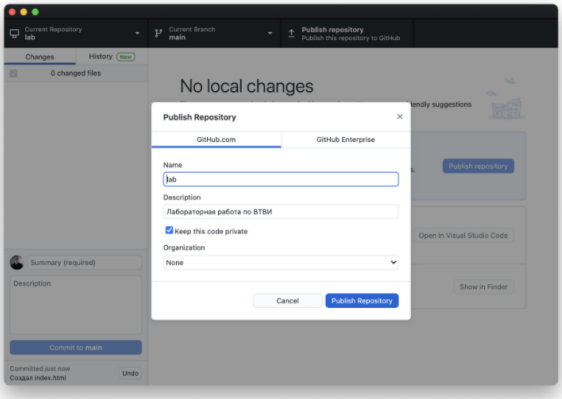
4. В редакторе кода создайте или отредактируйте файл, находящийся в этой папке: он отобразится в Github Desktop как измененный.

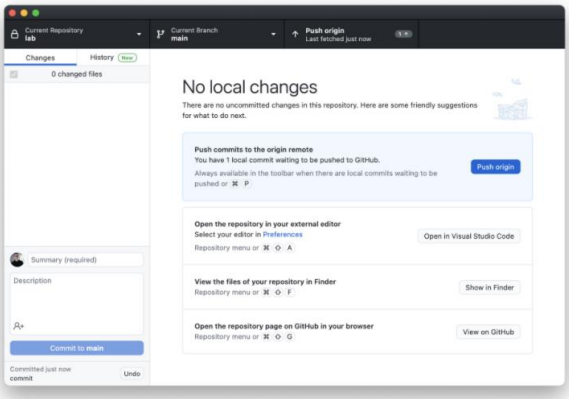


5. Теперь необходимо сделать коммит, для этого напишите название коммита и его описание (опционально) и нажмите на кнопку "Commit to main" (main – название ветки, созданной по умолчанию).

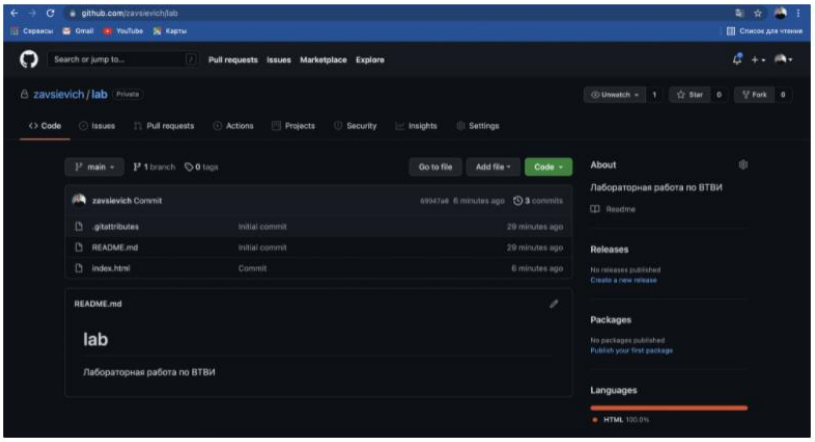


6. После коммита необходимо отправить изменения на удаленный репозиторий Github. Если репозиторий не был до этого создан, то будет использоваться кнопка с название "Publish repository", иначе ее описание сменится на "Push origin".





7. После отправки изменений можно зайти на сайт github.com, авторизоваться и в списке своих репозиториев найти только что созданный репозиторий, в котором будут все файлы проекта.



Задание к лабораторной работе №2

Для выполнения лабораторной работы необходимо установить и настроить редактор кода. Для управления версиями необходимо использовать Github Desktop.

Задание к лабораторной работе №2 состоит из задач разного уровня сложности. Выполнение задач 1-5 оценивается максимально в 5 баллов, задач 1-9 в 10 баллов, согласно модульно-рейтинговой системе.

Задача 1

Условие: Скачайте файлы для выполнения задания:

1. В index.html файле создайте скелет HTML документа.
2. В элементе <body> создайте абзац, в котором создайте две ссылки. Первая ссылка с текстом «Главная», вторая ссылка с текстом «Контакты». После абзаца с ссылками добавьте заголовок первого уровня с текстом «Главная», и еще один абзац с каким-нибудь текстом, можете использовать сайт-генератор случайного текста.
3. Скопируйте содержимое страницы index.html в файл contacts.html и поменяйте текст заголовка на «Контакты».
4. Перелинкуйте страницы между собой. В одном и другом файле пропишите ссылки в атрибуте href. Ссылка с текстом «Главная» должна вести на страницу index.html, ссылка с текстом «Контакты» должна вести на страницу contacts.html.
5. На странице контактов создайте еще один абзац с двумя ссылками. Первая ссылка с текстом «Наш телефон: +375291234567» и вторая ссылка с текстом «Наш e-mail: info@website.by». У ссылки с номером телефона используйте соответствующий протокол звонка при клике на эту ссылку. У ссылки с e-mail адресом используйте соответствующий протокол написания e-mail письма при клике на эту ссылку.
6. В навигации, там, где ссылки «Главная» и «Контакты», добавьте еще одну ссылку с текстом «Наш партнер». Эта ссылка должна быть на двух страницах. В ссылке «Наш партнер» добавьте ссылку на Google.ru, которая должна открываться в новой вкладке.
7. На главной странице создайте еще один абзац, с текстом «Наша презентация», в атрибуте href укажите ссылку на файл present.pdf и добавьте атрибут download.
8. Осуществите проверку файлов на валидность.

Задача 2

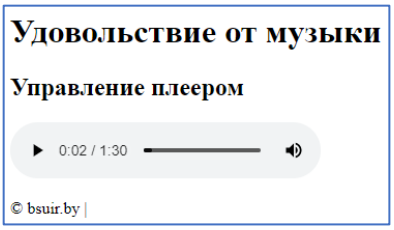
Условие: Измените HTML разметку документа используя семантические элементы. Используйте 9 разновидностей семантических тегов. Осуществите проверку файла на валидность. Рисунок приведен в качестве примера правильного использования семантических элементов.

```
<!-- Header -->
<header class="header">
    <nav class="nav">
     <ul>
        <a class="nav-item" href="#">Главная</a>
        <a class="nav-item" href="#">О нас</a>
        <a class="nav-item" href="#">Услуги</a>
        <a class="nav-item" href="#">Контакты</a>
     </ul>
    </nav>
</header>


<!-- Content -->
<div class="content">
    <main class="main">
        <section class="section">
            <h1>Кто мы такие?</h1>
            <p>Компания имеет своей целью разработку, производство и маркетинг инновационных продуктов с высокой технологической
            ценностью.</p>
        </section>
```

Задача 3

Условие: Создайте html-файл, результат которого показан на рисунке. Скачать звуковой файл. Для элемента <audio> необходимо использовать атрибут controls. Предусмотрите несколько источников звука, представленных с использованием атрибута src элемента <source>.

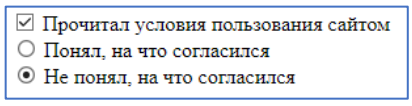


Задача 4

Условие: Создайте html-файл с формой регистрации, как показано на рисунке. Отправка данных осуществляется методом post.

Задача 5

Условие: Создайте html-файл с флажком (checkbox) и переключателями (radio) в формах, как показано на рисунке.

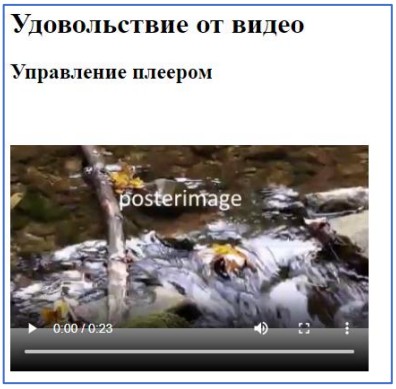


Задача 6

Условие: Необходимо правильно оформить HTML разметку и исправить ошибки в данном файле. Соблюдайте вложенность элементов. Проверьте все атрибуты и кавычки. Осуществите проверку файла на валидность.

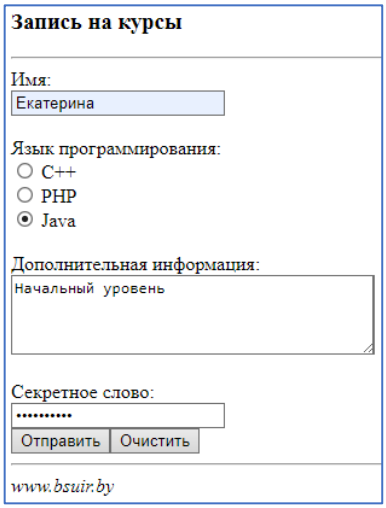
Задача 7

Условие: Создайте html-файл, результат которого показан на рисунке. Скачать видеофайл. Размер высоты окна для отображения видеоданных 300 px, а ширины 400 px. До начала воспроизведения видео на его месте необходимо отобразить изображение "posterimage.jpg". Предусмотрите несколько источников видеоданных, представленных с использованием атрибута src элемента <source>. Подключите субтитры к видео используя элемент <track> с атрибутами kind, src, srclang, label.



Задача 8

Условие: Создайте html-файл, результат которого показан на рисунке. Используйте атрибуты action, method элемента <form>. Для элемента <input> необходимо учитывать правильность значений атрибута type для определенных полей формы. (например, для поля «Секретное слово» <input type="password">). При клике на кнопку «Очистить» поля формы обновляются, на кнопку «Отправить» – происходит отправка данных на сервер методом post.



Задача 9

Условие: Создайте html-файл, результат которого показан на рисунке. Используйте элементы <option>, <fieldset>.