**iframe**

По-английски iframe — сокращение от inline frame, что значит «встроенная в строку рамка». Это контейнер, в который помещают информацию с других сайтов. По сути — обычный элемент страницы. Можно задать ему ширину и высоту: в рамке таких размеров и будет отображаться подгружаемая информация. В текстах на русском языке элемент с тегом iframe часто называют словом «фрейм».

**Атрибуты тега iframe**

height — высота фрейма в пикселях:

Скопировать кодHTML

<iframe height="250px"></iframe>

width — ширина в пикселях:

Скопировать кодHTML

<iframe width="250px"></iframe>

src — ссылка на веб-страницу, откуда поступают данные:

Скопировать кодHTML

<iframe src="https://music.yandex.ru/iframe/#track/41284288/5400781/"></iframe>

allowfullscreen — разрешает полноэкранный режим:

Скопировать кодHTML

<iframe width="100%" allowfullscreen src="https://music.yandex.ru/iframe/#track/41284288/5400781/"></iframe>

frameborder — это устаревший атрибут, он отображает вокруг содержимого фрейма рамку с эффектом трёхмерности. По умолчанию рамка рисуется и значение этого атрибута frameborder="1". Значение "0" её скрывает.

В HTML5 не применяют этот атрибут, а управляют рамкой через свойство border в CSS. Но frameborder встречается в популярных готовых фреймах, к примеру — в коде вставки видео с Youtube. Вам нужно знать этот атрибут, чтобы найти его и удалить, иначе [валидатор](https://validator.w3.org/" \t "_blank) выдаст ошибку.

Скопировать кодHTML

<iframe frameborder="0" width="600" height="100" src="https://music.yandex.ru/iframe/#track/41284288/5400781/"></iframe>

Начнём вёрстку макета. Проект будет посвящён Транссибирской магистрали.

***ИЗ ПРАКТИКИ!!!!!!!!!!!!!!!!!!***

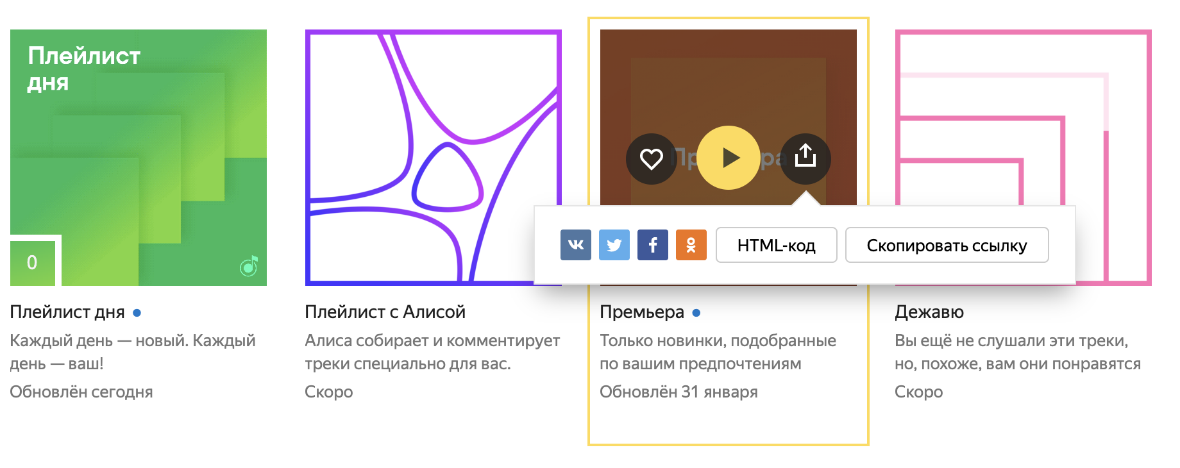
Сначала нам нужно вставить плейлист с композициями, которые будем слушать в пути.

Переходите на Яндекс.Музыку: <https://music.yandex.ru/home>.

Вы можете выбрать любой плейлист. Главное, чтобы внешний вид соответствовал макету.

Например, найдите в поиске исполнителя A$AP ROCKY. Среди его плейлистов, указаных правее композиций и альбомов, в верхнем меню выберите «Лучшее: A$AP ROCKY».

Наведите на него мышь и кликните пиктограмму со стрелочкой «Поделиться»:



Или сразу перейдите по ссылке на плейлист: <https://music.yandex.ru/users/yamusic-bestsongs/playlists/631822>, кликните на иконку с многоточием и нажмите «Поделиться».

Во всплывающем меню выберите «HTML-код». Откроется окно с конструктором, где можно задать размеры фрейма, его оформление, а после скопировать HTML-код.

Вставьте его в заготовленный раздел «Что послушать».

Код для iframe приходит с заданными стилями. Но их можно менять! Добавьте в атрибуте style сплошную рамку красного цвета для плейлиста.

Задайте стиль рамки свойством border: 1px solid red; в инлайновом стиле тега iframe.

Теперь нужно подумать, что почитать во время путешествия. Выберем книгу в интернет-магазине Amazon: <https://www.amazon.com/Kindle-eBooks/b?ie=UTF8&node=154606011>. Можно выбрать в поиске книгу на свой вкус.

Мы предлагаем почитать о понятии Икигай и узнать, как стать самым счастливым человеком. По крайней мере, на время путешествия и хотя бы в пределах бесплатного предпросмотра.

Для этого в поиске введите ”Ikigai“ и выберите книгу.

Или просто перейдите по ссылке: <https://www.amazon.com/Ikigai-Japanese-Philosophy-Alan-Daron-ebook/dp/B07BPGCS41/ref=pd_sim_351_1/143-1361162-3352612>

Справа, рядом со значками соцсетей, находится кнопка ”embed“. Кликнув по ней, выберите в открывающемся окне ”Embed on your site (HTML)“. Вставьте код фрейма в раздел «Что почитать» нашего проекта. Управляйте в теге iframe значениями атрибутов width и height.

В социальных сетях есть много разных iframe, которые вы можете использовать в своих проектах. Здесь мы сошлёмся на пост в Facebook. Как это сделать, описано на официальном сайте Facebook для разработчиков: <https://developers.facebook.com/docs/plugins/page-plugin/>.

Можете написать свой пост и добавить его в проект. Мы для примера предлагаем использовать этот:

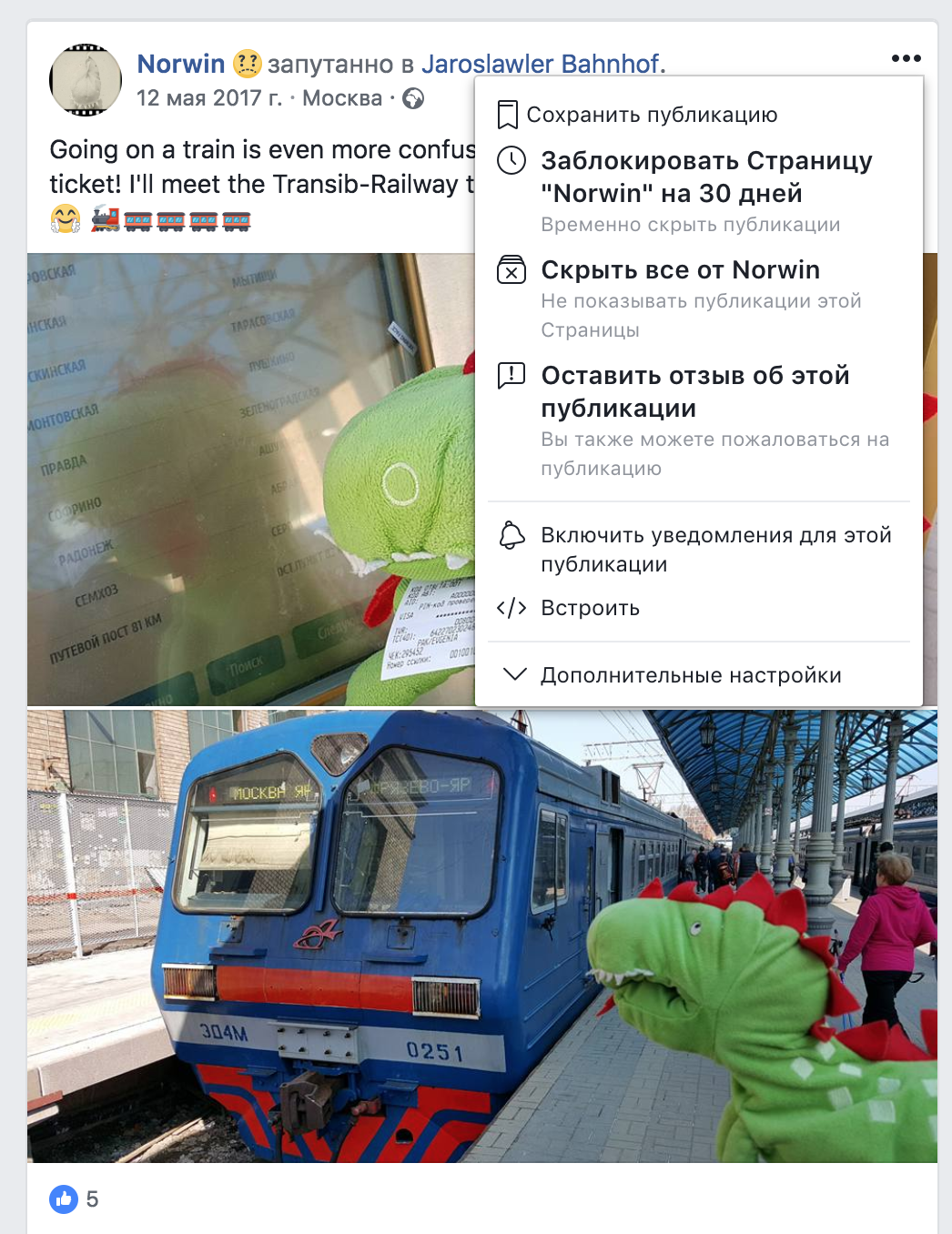
<https://www.facebook.com/Drachiosaurus/posts/1857733597882973>

Откройте этот пост и кликните справа пиктограмму с многоточием. Она доступна, если у вас есть аккаунт на Facebook. А если нет, сразу скопируйте в буфер код:

Скопировать кодHTML

<iframe src="https://www.facebook.com/plugins/post.php?href=https%3A%2F%2Fwww.facebook.com%2FDrachiosaurus%2Fposts%2F1857733597882973&width=500" width="500" height="688" style="border:none;overflow:hidden" scrolling="no" allowtransparency="true" allow="encrypted-media"></iframe>

Появится всплывающее меню:



Выберите пункт «Встроить». При клике откроется код, готовый для использования в проекте. Вставьте его в раздел «Мой отчёт».

Мы разобрали возможные трудности в работе с iframe. Продолжим вёрстку проекта. Будет здорово знать, откуда и куда мы едем. Для этого добавим карту.

У Яндекс.Карт есть конструктор создания карт: <https://yandex.ru/map-constructor/>

Выберите «Создать карту» → «Новая карта».

Придумайте заголовок и введите его в поле «Название».

Теперь создадим метки для отправного и конечного пунктов путешествия. В поле «Найти» введите Москва (ставим метку на неё, нажимаем на кнопку «Готово»). Далее то же самое с Владивостоком. Можно выбрать внешний вид меток и сделать их для каждого города, где будут остановки.

После этого нажмите «Линии». Соедините метки линией, начав от Москвы и до Владивостока. Когда линия вас устроит, выбирайте «Завершить» и «Готово».

Если карта выглядит так, как вам нужно — нажмите «Сохранить и продолжить». Для встраивания выберите опцию «Статическая карта». Задайте ей нужные размеры. Теперь нажмите кнопку «Получить код карты». Яндекс предложит код ссылки, который надо вставить в HTML-файл вашего проекта.

**Кроссбраузерность iframe**

Проверим на сайте [Can I Use](https://caniuse.com/" \t "_blank), какие браузеры поддерживают фрейм. Зайдите на сайт и введите iframe в окно, рядом со знаком вопроса.

Один из способов разрешения этой ситуации — разместить информацию для пользователей между двумя тегами <iframe>:

Скопировать кодHTML

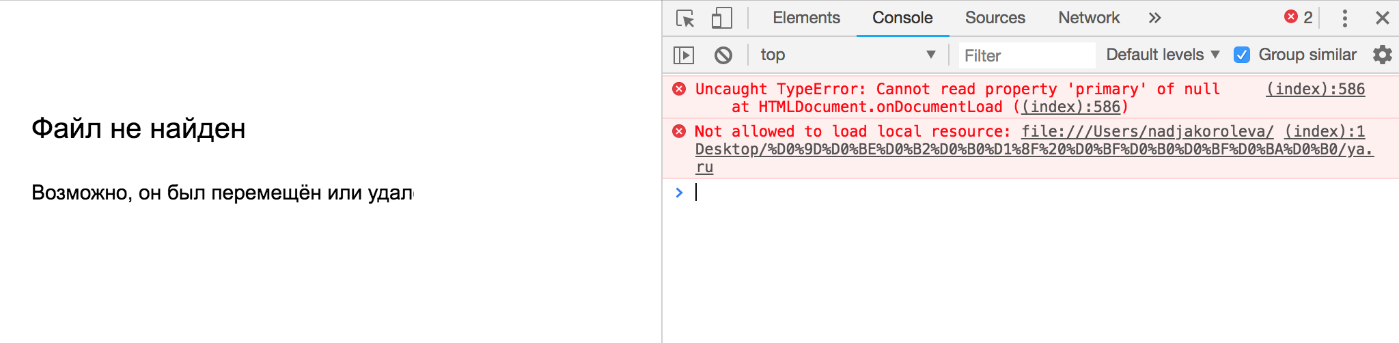
<iframe>Здравствуйте, ваш браузер не может показать классную штуку, потому что не поддерживает iframe.</iframe>

Пользователь с Internet Explorer не посмотрит видео, но увидит текст, написанный между тегов <iframe></iframe>. Мы делаем это для того, чтобы пользователь не пугался при виде пустого блока.

Такой подход допустим при работе со старыми браузерами. Он описан в документации HTML ([W3C](https://www.w3.org/TR/2011/WD-html5-20110525/the-iframe-element.html#the-iframe-element), [WHATWG](https://html.spec.whatwg.org/multipage/iframe-embed-object.html#the-iframe-element)). Тем не менее, [W3C-валидатор](https://validator.w3.org/) укажет на ошибку, когда встретит код с текстом внутри фрейма. То ли разработчики валидатора смотрят в будущее, где нет устаревших браузеров, то ли проблема связана с особенностью работы сервиса. Учитывайте эту особенность в работе и отталкивайтесь от задачи. Если нужно не получить ошибок от валидатора, не пишите текст внутри фрейма. Если важнее поддерживать старые браузеры — пишите. В заданиях вы проделаете это упражнение, несмотря на любые неопределённости.

Бывает, что вы сделали фрейм по правилам, а он всё равно не отображается. Скорее всего, разработчик запретил показ своего ресурса во фреймах сторонних сайтов.

Если создатель сайта запретил показ своего фрейма на других ресурсах, в консоль разработчика будет выведено сообщение об ошибке.



Решение этой проблемы — связаться с разработчиком сайта и попросить его дать разрешение на встраивание контента.

# Виджеты-информеры

Один из видов iframe — виджеты. Слово widget получилось при слиянии ”window“, «окно» и ”gadget“, «устройство». Это «графический гаджет» — модуль, встраиваемый на сайт. Виджет — это готовый инструмент. Как будет выглядеть виджет, решает его разработчик. Виджеты бывают двух типов:

интерактивные — с которыми можно взаимодействовать (например, поисковые строки);

статичные — независимые от действий пользователя (такие как Яндекс.Погода или Яндекс.Время). Подобные виджеты называются «информеры».

Полезный информер — расписание. Его можно собрать в конструкторе сервиса Яндекс.Расписания.

<https://rasp.yandex.ru/informers/informer/>

Расписание транссибирских экспрессов нам пригодится. Перейдите на Яндекс.Расписания: <https://rasp.yandex.ru/informers/informer/>

В меню «Настройка» выберите «Расписание по направлению». Укажите пункты отправления и прибытия: Москва и Владивосток; и вид транспорта — поезд.

Перейдите в меню «Оформление» и украсьте виджет на свой вкус.

Вставьте полученный код фрейма в элемент с классом iframe\_\_item блока iframe\_\_tickets.

Добавьте альтернативный текст для фрейма: Расписание поездов по Транссибирской магистрали.

# Тег <video>

Для размещения видео на сайте используют тег <video>. Он создаёт на веб-странице область для видеоролика. Видеоролик размещают на сервере, и прописывают адрес к видеофайлу как значение атрибута src:

Скопировать кодHTML

<video src="url" >

*<!-- Здесь url заменяет реальный адрес ролика в интернете -->*

Сейчас тег работает с четырьмя форматами: MP4, MOV, WebM и OGG. Форматов несколько, потому что нет единого формата, который поддерживался бы всеми браузерами.

По этой причине каждый формат нужно проверять на сайте [Can I Use](https://caniuse.com/). Если поддержки в нужном браузере нет, можно задействовать дополнительный тег <source> (источник). Тогда форматы указывают в значениях атрибута type:

Скопировать кодHTML

<video width="320" height="240" controls>

<source src="kino.mp4" type="video/mp4">

<source src="kino.ogg" type="video/ogg">

*<!-- Если не сработает формат mp4, браузер попробует ogg -->*

</video>

## **Атрибуты тега <video>**

autoplay — видео запускается, как только загрузится. При использовании этого атрибута нужно иметь в виду, что некоторые браузеры по умолчанию его игнорируют. Например, в Safari атрибут сработает, только если пользователь самостоятельно указал, что хочет получать видео с автоматическим воспроизведением (см. [Can I Use](https://caniuse.com/" \t "blank)). Цель этого — оградить пользователей от навязчивой рекламы.

Скопировать кодHTML

<video autoplay>

controls — станет доступен виртуальный пульт управления с ползунком изменения громкости, кнопками воспроизведения, паузы, перемотки и прочими инструментами:

Скопировать кодHTML

<video controls>

loop — как только видео закончится, оно запускается с начала:

Скопировать кодHTML

<video loop>

muted — звук будет отключён:

Скопировать кодHTML

<video muted>

src — ссылка на воспроизводимый ролик:

Скопировать кодHTML

<video src="URL">

type — формат видео:

Скопировать кодHTML

<video type="video/mp4">

height — высота области воспроизведения в пикселях:

Скопировать кодHTML

<video height="250px">

width — ширина области воспроизведения в пикселях:

Скопировать кодHTML

<video width="250px">

poster — ссылка на изображение, которое пользователь видит до того, как нажмёт ”Play“, или пока видео загружается:

Скопировать кодHTML

<video poster="/images/example.gif">

Если poster не указан, то в качестве превью браузер покажет первый кадр видео, когда он загрузится.

# Тег <audio>

Тег работает с тремя форматами аудио: MP3, WAV и OGG. Каждый из них нужно проверить на сайте [Can I Use](https://caniuse.com/" \t "blank). Если браузер не поддерживает необходимый формат, мы снова применим тег <source>:

Скопировать кодHTML

<audio>

<source src="sound.mp3" type="audio/mp3">

<source src="sound.ogg" type="audio/ogg">

*<!-- Если не сработает mp3, браузер попробует ogg -->*

</audio>

## **Атрибуты тега audio**

autoplay — аудиозапись исполняется, как только загрузится:

Скопировать кодHTML

<audio autoplay>

Не стоит злоупотреблять этим атрибутом. Более того, хорошей практикой считается выключение автоматического воспроизведения. Некоторые современные браузеры даже блокируют его. Это улучшает пользовательский опыт и помогает избежать загрузки лишних данных (ведь интернет может быть дорогой или медленный). Так мы дадим пользователю самому решать, проигрывать аудиозапись или нет, и в какой момент это сделать. В этом поможет следующий атрибут.

controls — станет доступен виртуальный пульт управления с ползунком изменения громкости, кнопками воспроизведения, паузы, перемотки и прочими инструментами:

Скопировать кодHTML

<audio controls>

loop — как только аудио закончится, оно заиграет снова:

Скопировать кодHTML

<audio loop>

muted — режим отключения звука:

Скопировать кодHTML

<audio muted>

src — ссылка на путь к аудиофайлу:

Скопировать кодHTML

<audio src="URL">

type — формат аудио:

Скопировать кодHTML

<audio type="audio/ogg">

# API YouTube

API (от англ. application programming interface, «программный интерфейс приложения») — интерфейс взаимодействия между сайтом и сторонними программами и серверами. Это когда встраиваемый в веб-страницу контент несёт с собой набор команд, программ, которыми можно пользоваться на сайте.

Самый популярный пример использования API — авторизация через социальные сети. Сервисы и приложения используют базы данных социальных сетей, чтобы вы смогли авторизоваться на сайте, не регистрируясь на нём. При этом сервис получает информацию о вас и использует её в своих целях. Amazon предлагает книги, основываясь на выборе ваших друзей в Facebook. Работает это так: Facebook имеет API, сообщающий интернет-сервису Amazon об интересах пользователя.

## **YouTube API**

Простейший API — конструктор видео. Он даёт возможность снабдить встраиваемое видео атрибутами вне стандартов HTML.

По ссылке можно загрузить идентификатор ролика, поставить время начала воспроизведения и прописать начальную громкость. Идентификатор можно найти, открыв нужное видео и скопировав последние символы его адреса после символов ?v=.

Например, в адресе https://www.youtube.com/watch?v=RFFt6ge-yxE идентификатор — RFFt6ge-yxE.

<https://developers.google.com/youtube/youtube_player_demo?hl=ru>

Определив нужные параметры, нажимаем «Обновить проигрыватель» и получаем новый код фрейма для своего проекта:

Скопировать кодHTML

<iframe id="ytplayer" type="text/html" width="720" height="405" src="https://www.youtube.com/embed/BZsUo1KuMWU?start=3" frameborder="0" allowfullscreen>

## **Как формируется ссылка. Параметры запроса**

Внутри <iframe> мы передаём ссылку на видео и стандартные атрибуты фрейма, которые вам уже знакомы: frameborder="0" и allowfullscreen.

Значение атрибута src стоит рассмотреть подробнее:

Скопировать кодHTML

src="https://www.youtube.com/embed/BZsUo1KuMWU?start=3" frameborder="0" allowfullscreen

Внутри src после ссылки на сайт youtube.com прописано /embed/ (англ. embed, «врезать», «внедрять»). Это означает, что в ход идёт API YouTube, т. е. код с готовым набором команд. Они сработают, только если мы напишем код по всем правилам. Такие правила называются документацией API. Если поставить /embed/ в другое место, ничего не выйдет.

Например, вот так видео не заработает:

Скопировать кодHTML

src="https://www.youtube.com/BZsUo1KuMWU?start=3/embed"

Оставим так, как было в начале:

Скопировать кодHTML

src="https://www.youtube.com/embed/BZsUo1KuMWU?start=3" frameborder="0" allowfullscreen

За /embed/ следует идентификатор ролика, который мы хотим показывать. Здесь идентификатором выступает набор символов BZsUo1KuMWU.

Затем идёт знак вопроса — в данном случае это сепаратор, разделитель. За ним следуют параметры, которые мы передаём серверу.

В примере параметр — время старта ролика. Слово start означает старт, начало ролика, знак = присваивает значение. Мы передаём атрибуту start значение 3 (...?start=3...). Это значит, что ролик будет воспроизводиться с третьей секунды.

**JS Scroll**

Добавим последний штрих. Скролл в нашем блоге имитирует путешествие по Транссибу: при вертикальной прокрутке меню показывает время в пути и количество преодолённых километров.

Чтобы информация обновлялась, нужно «взять» в меню содержание элементов, в которых записаны километры и дни. Получить элемент можно вызовом уже знакомого метода querySelector().

Объявим переменную, в которой хранится счётчик километров — элемент с id="counter-km".

И ещё одну переменную для хранения счётчика дней, то есть элемента с id="counter-days".

Скопировать кодJAVASCRIPT

*// объявляем переменную для хранения счётчика километров*

let kmCounter = document.querySelector('#counter-km');

*// объявляем переменную для хранения счётчика дней*

let dayCounter = document.querySelector('#counter-days');

Возьмём каждый из блоков, внутри которых размещены города. Используем тот же метод querySelector.

Блоки расставлены сверху вниз. В свойстве offsetTop (от англ. offset, «смещение» и top, «верх») сохраняется расстояние от верхней границы каждого блока до верха страницы, т. е. границы элемента body. Обычно свойство offset ведёт отсчёт от верхнего края блока-родителя, у которого свойство position имеет одно из трёх значений: relative, absolute или fixed. Но в нашем примере такого родителя нет, и «нулевым километром» становится верх body*.* Для блока с идентификатором id='some\_id' координату его верхней границы относительно тела документа можно получить выражением document.querySelector('#some\_id').offsetTop.

Объявим переменные для каждого блока и сохраним в них соответствующие значения границ блоков:

Скопировать кодJAVASCRIPT

*// смещение блока с видео про перегон Москва — Кострома*

let kostromaOffset = document.querySelector('#city-kostroma').offsetTop;

*// смещения остальных блоков*

let ekbOffset = document.querySelector('#city-ekb').offsetTop;

let omskOffset = document.querySelector('#city-omsk').offsetTop;

let yablPerevalOffset = document.querySelector('#city-yablonovy-pereval').offsetTop;

let khabarovskOffset = document.querySelector('#city-khabarovsk').offsetTop;

Когда мы скроллим страницу, координата нашей позиции по вертикальной оси сохраняется в свойстве pageYOffset (от англ. page Y offset, «смещение на странице по вертикальной оси») объекта window. Будем хранить это значение в переменной scrollY (от английского scroll Y, «прокрутка по вертикали»):

Скопировать кодJAVASCRIPT

let scrollY = window.pageYOffset;

Чтобы вовремя изменить содержание пункта меню, нужно отслеживать скролл в режиме реального времени. Это значит следить за событием. В этом поможет знакомый метод объекта document — addEventListener(событие, функция-обработчик события).

Событие, которое мы будем отслеживать — scroll.

Осталось написать функцию-обработчик события. Она при прокрутке окна должна «слушать» скролл, сравнивать позицию scrollY с высотой границы блока и решать, в каком блоке мы сейчас находимся.

Скопировать кодJAVASCRIPT

document.addEventListener('scroll', function () {

*// здесь будем сравнивать положения скролла и блоков*

});

Обработчик будет сопоставлять текущую позицию скролла и верхние границы блоков. Если позиция скролла меньше или равна позиции блока с определённым городом, запишем в меню количество пройденных километров и дней через свойство textContent (англ. «текстовое содержимое»).

Свойство textContent устанавливает или получает разметку элемента. Нам это свойство позволит показывать новые значения в меню через переменные kmCounter и dayCounter.

В этих двух переменных хранятся элементы с идентификаторами counter-km и counter-days.

Скопировать кодJAVASCRIPT

*// если позиция скролла меньше или равна позиции блока с городом Кострома*

if (scrollY <= kostromaOffset) {

*// тогда в HTML переменных kmCounter и dayCounter записываем «ноль»*

kmCounter.textContent = 0;

dayCounter.textContent = 0;

*// если позиция скролла меньше или равна позиции блока с городом Екатеринбург*

} else if (scrollY <= ekbOffset) {

*// тогда в HTML переменных kmCounter и dayCounter записываем 376 (км) и 1 (первый день в пути)*

kmCounter.textContent = 376;

dayCounter.textContent = 1;

*// а когда позиция скролла больше, чем у блока с Екатеринбургом, и меньше, чем у блока с Омском, сохраняем в переменных 1819 (км) и 2 (дня в пути)*

} else if (scrollY <= omskOffset) {

kmCounter.textContent = 1819;

dayCounter.textContent = 2;

*// далее по аналогии*

} else if (scrollY <= yablPerevalOffset) {

kmCounter.textContent = 6110;

dayCounter.textContent = 4;

} else if (scrollY <= khabarovskOffset) {

kmCounter.textContent = 8534;

dayCounter.textContent = 6;

*// если позиция скролла превысила координату блока с Хабаровском - мы на последнем перегоне*

} else {

kmCounter.textContent = 9300;

dayCounter.textContent = 6;

}

Получается не слишком красиво: предыдущий блок прокручен вверх не полностью, а меню уже показывает дни и километры следующего. Добавим каждой переменной слагаемое, чтобы отсрочить перемену:

Скопировать кодJAVASCRIPT

if (scrollY <= kostromaOffset + 200) {

kmCounter.textContent = 0;

dayCounter.textContent = 0;

} else if (scrollY <= ekbOffset + 500) {

kmCounter.textContent = 376;

dayCounter.textContent = 1;

} else if (scrollY <= omskOffset + 500) {

kmCounter.textContent = 1819;

dayCounter.textContent = 2;

} else if (scrollY <= yablPerevalOffset + 500) {

kmCounter.textContent = 6110;

dayCounter.textContent = 4;

} else if (scrollY <= khabarovskOffset + 500) {

kmCounter.textContent = 8534;

dayCounter.textContent = 6;

} else {

kmCounter.textContent = 9300;

dayCounter.textContent = 6;

}

Это довольно простой пример скролла. Существует много других идей и решений, осуществляемых при прокрутке страницы. Например, при скролле может увеличиваться шрифт заголовка или появляться анимация.

Важно проверять, как ведут себя элементы и при скролле вверх, и при скролле вниз.

# Заключение

Как вещи в чемодан, мы сложили из разных «кусочков» других сайтов целый проект о путешествии по Транссибирской магистрали.

Теперь вы знаете:

* Что такое iframe, где найти его код, как стилизовать iframe под себя, и почему он иногда не отображается.
* Что такое виджеты и информеры, чем они друг от друга отличаются.
* Как вставлять в проект видео и аудио тегами <video> и <audio>. Вы познакомились с их атрибутами и разобрались, как быть, если один из браузеров не поддерживает мультимедийный формат.
* Что такое API и как работать с конструктором API YouTube.

Вы добавили скрипт с обновлением информации о пройденных километрах и днях путешествия и сделали веб-страницу интерактивной. Скоро вы разберётесь подробно, как создавать такие скрипты на JS.

Следующая тема будет не менее интересной. Мы рассмотрим декоративные эффекты: анимации, трансформации, тени, градиенты. С их помощью вы научитесь делать сайты более интерактивными и увлекательными. Пользователи обращают внимание на кнопку, красиво и плавно отбрасывающую тень. Или на анимированный шрифт по наведению.

А пока вы можете изучить несколько дополнительных источников по текущей теме:

[Документация тега iframe](https://www.w3.org/TR/html52/semantics-embedded-content.html#the-iframe-element) (на английском языке).

[Как встраивать медиаконтент](https://developer.mozilla.org/ru/docs/Learn/HTML/Multimedia_and_embedding/Video_and_audio_content).

[Поддерживаемые форматы медиа](https://developer.mozilla.org/ru/docs/Web/HTML/%D0%9F%D0%BE%D0%B4%D0%B4%D0%B5%D1%80%D0%B6%D0%B8%D0%B2%D0%B0%D0%B5%D0%BC%D1%8B%D0%B5_%D0%BC%D0%B5%D0%B4%D0%B8%D0%B0_%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D1%8B).

[Как сделать собственный видео-плеер](https://habr.com/ru/company/microsoft/blog/127295/).