

HTML/CSS. Интерактивный курс

# Основы HTML5

[HTML5/CSS3]

---



# На этом уроке

1. Познакомимся с основами html5.
2. Научимся создавать списки, добавлять ссылки и изображения на странице.
3. Узнаем, как создавать поля ввода на странице.
4. Разберёмся, как добавить интерактивную карту на сайт.

## Оглавление

[Введение](#)

[Спецсимволы](#)

[Типограф](#)

[Комментарии](#)

[Списки](#)

[Структура вложенного списка](#)

[Гиперссылки](#)

[Виды ссылок](#)

[Относительные ссылки](#)

[Абсолютные ссылки](#)

[Якоря](#)

[Изображения](#)

[JPEG](#)

[GIF](#)

[PNG](#)

[PNG – 8](#)

[PNG – 32](#)

[SVG](#)

[Загрузка изображений на страницу](#)

[Формы и их элементы](#)

[Элементы форм](#)

[Тег <label>](#)

[Добавление интерактивной Яндекс.Карты](#)

[Дополнительные материалы](#)

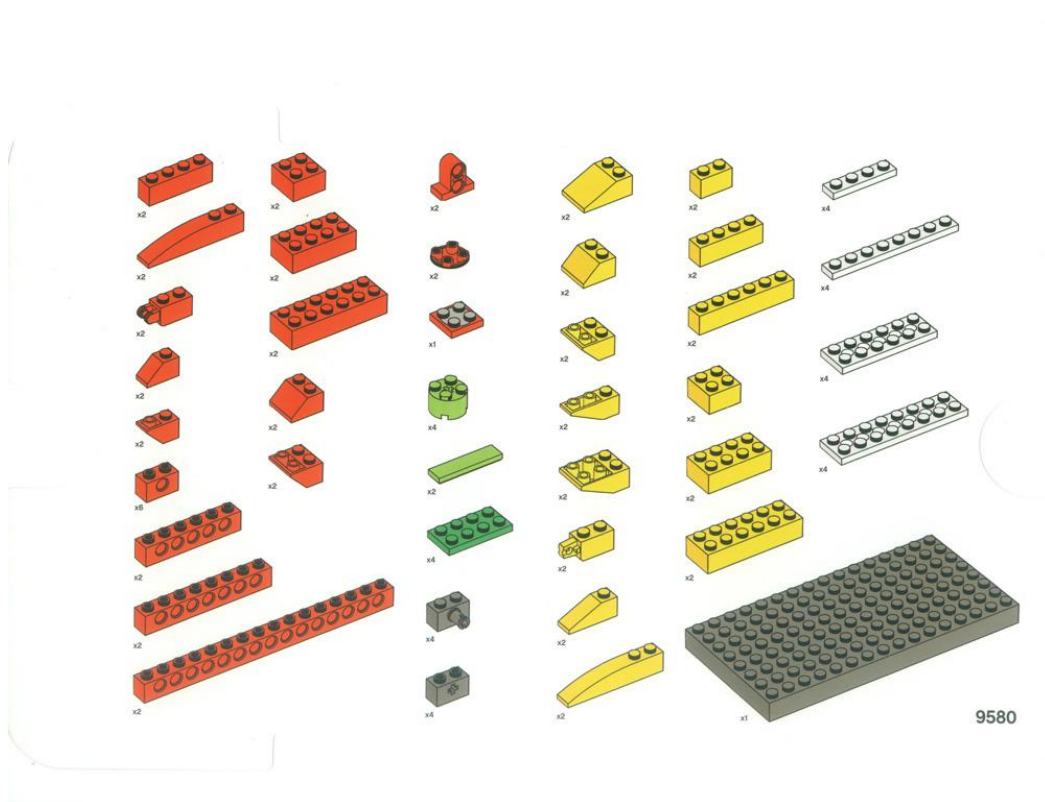
[Используемые источники](#)

[Практическое задание](#)

# Введение

HTML (HyperText Markup Language) — это язык гипертекстовой разметки страницы. Он состоит из тэгов. Каждый элемент обозначается в исходном документе начальным (открывающим) и конечным (закрывающим) тегом. Есть несколько визуальных сравнений, чтобы лучше представить html.

1. Любая html-страница похожа на конструктор, в котором детали — html-тэги: заголовки, параграфы, списки, ссылки, картинки. В итоге html — это наполнение нашего сайта.



У нас есть все необходимые элементы для создания абсолютно любого контента на сайте.

2. Чтобы понять, где открывается и закрывается элемент, используются парные тэги.

Закрывающий тег образуется путём добавления слэша «/» перед именем тега: `<имя тега>...</имя тега>`. Между начальным и закрывающим тегами находится содержимое — контент.

Здесь проще всего представить, как вы путешествуете на автомобиле. Когда въезжаете в город, видите табличку города «Москва». На выезде из города — такая же табличка, только название города перечёркнуто наклонной чертой «~~Москва~~».



Браузер просматривает (интерпретирует) HTML-документ, выстраивая его структуру. Он отображает её в соответствии с инструкциями, включёнными в файл (таблицы стилей, скрипты). Если разметка правильная, то в окне браузера отобразится HTML-страница, содержащая HTML-элементы — заголовки, таблицы, изображения и т. д.

Разберём, какие ещё элементы могут быть кроме тегов и текстовой информации.

## Спецсимволы

Используя спецсимволы в HTML, можно заменять уже имеющиеся на клавиатуре компьютера символы или указать те, что отсутствуют на клавиатуре. Например, значок копирайта.

Любые теги никак не показываются в окне браузера, поскольку воспринимаются им как команды для вставки элементов и изменения их свойств. Но иногда требуется на веб-странице вывести теги, например, для демонстрации HTML-кода. В этом случае используйте спецсимволы `&lt;` и `&gt;` для замены угловых скобок `<` и `>`.

Некоторые варианты спецсимволов:

Код в HTML	Внешний вид	Описание
<code>&amp;nbsp;</code>		неразрывный пробел
<code>&amp;copy;</code>	©	знак copyright
<code>&amp;quot;</code> и <code>&amp;raquo;</code>	« и »	двойные кавычки
<code>&amp;lt;</code> и <code>&amp;gt;</code>	< и >	символы
<code>&amp;prime;</code>	'	одиночный штрих

Спецсимволов много, поэтому есть простой способ добавления их на страницу — типограф.

# Типограф

**Типограф** помогает автоматически расставить неразрывные пробелы, исправить мелкие опечатки, привести кавычки к правильному виду, заменить дефисы на тире в нужных местах. Можно выбирать любой понравившийся типограф.

1. [Типограф 1.](#)
2. [Типограф 2.](#)
3. [Типограф 3.](#)

Принцип работы типографа — простой. Нужно выделить весь html-код с текстом, который вы создали, скопировать его, добавить в типограф и нажать клавишу «типографировать». Далее заменяем всё содержимое на уже оттипографированный код.

# Комментарии

В HTML-документах можно оставить какой-либо комментарий, например, название части страницы. Это не отобразится в браузере. Но любой желающий сможет увидеть этот текст, если воспользуется функцией «просмотр кода страницы» в любом браузере.

Комментарии применяются для того, чтобы не отображать теги в браузере. Это может пригодиться при редактировании веб-сать целые блоки кода.

Пример:

```
<!--Текст, который не будет отображаться на сайте-->  
<p>Текст основной части сайта</p>
```

На прошлом уроке мы рассмотрели самые часто используемые теги на странице. Но это далеко не весь список html-элементов. Поэтому рассмотрим, какие html-элементы потребуются ещё.

1. Списки.
2. Изображения.
3. Ссылки.
4. Элементы форм.

# Списки

Списки в HTML-документах бывают двух видов: маркированные и нумерованные. Отличаются они тем, что у маркированных элементы списка начинаются с точек (маркеров), а у нумерованного — с цифр или букв.

Структура маркированного списка:

```
<ul>  
  <li>Первый элемент списка</li>  
  <li>Второй элемент списка</li>
```

```
<li>Третий элемент списка</li>
</ul>
```

Структура нумерованного списка:

```
<ol>
  <li>Первый элемент списка</li>
  <li>Второй элемент списка</li>
  <li>Третий элемент списка</li>
</ol>
```

Из примеров видим — структура обоих типов списков одинаковая. Единственное различие в том, что в случае с маркированным списком контейнер будет называться `<ul>`, а в нумерованном — `<ol>`. В нумерованном списке иногда требуется начать список с какой-нибудь определённой цифры или буквы. Для этого существует атрибут `start`. В его значении указывается, с какого элемента будет начинаться список. В списках есть возможность менять вид маркеров. Например, арабские цифры изменить на римские, либо вообще убрать маркеры.

## Структура вложенного списка

```
<ol>
  <li>Первый элемент списка</li>
  <li>Второй элемент списка
    <ul>
      <li>Первый элемент вложенного списка</li>
      <li>Второй элемент вложенного списка</li>
      <li>Третий элемент вложенного списка</li>
    </ul>
  </li>
  <li>Третий элемент списка</li>
</ol>
```

Вложенные списки предназначены для организации сложной иерархической структуры текста, обычно юридических или технических документов. Такие списки используются для создания многоуровневых меню и навигации по сайту.

## Гиперссылки

Гиперссылка — активный элемент (текст, изображение, кнопка и т. п.) одного документа. При нажатии на неё происходит переход к другому документу или к его элементу. Документы, содержащие гиперссылки, называются гипертекстовыми. Гиперссылка может быть добавлена к любому элементу гипертекстового документа и обычно выделяется графически.

В HTML-документах текстовые ссылки по умолчанию выделяются синим цветом. При наведении на них курсора мыши в окне браузера меняют цвет или выделяются подчёркиванием. Посещённая ранее ссылка обычно выделяется цветом, отличным от цвета непосещённой ссылки.

Тег ссылки называется <a>. У него есть обязательный атрибут href, в значении которого мы указываем на файл, место в документе или страницу другого сайта, куда хотим перейти. В содержимом этого тега нужно написать текст, который увидит пользователь.

## Виды ссылок

1. Относительные ссылки.
2. Абсолютные ссылки.
3. Якоря.

### Относительные ссылки

Используются для перемещения внутри документа или сайта.

```
<a href="file_name.html">текст ссылки, который видит пользователь</a>
```

### Абсолютные ссылки

Используются для перехода на страницы внешнего сайта. Для этого в значении атрибута href нужно указать полный путь до той страницы, на которую мы хотим перейти, включая тип протокола.

```
<a href="https://mail.ru" target="_blank">
  страница mail.ru откроется в новой вкладке
</a>
```

У тега ссылки существует атрибут target. Если указать target="\_blank", то страница откроется в новой вкладке браузера. Раньше это значение официально не поддерживалось в HTML. Но люди всё равно использовали его, поскольку оно работало. Теперь в HTML5 это полностью поддерживается.

Когда использовать target="\_blank":

- если нужно открыть сторонний ресурс и не потерять страничку вашего сайта;
- если пользователь работает над чем-то, что может потеряться при изменении текущей страницы;
- если ссылка ведёт на сторонний ресурс в интернете.

Когда не использовать target="\_blank":

- в меню, иначе его пункты будут открываться в новых вкладках;
- из-за личных предпочтений;
- если клиент думает, что так будет лучше.

## Якоря

При помощи гиперссылок можно перемещаться не только между страницами документа, но и внутри страницы. Это полезно, когда ваша статья состоит из нескольких разделов и имеет объемное содержание.

В этом случае разумно использовать так называемые якоря, чтобы быстро переходить в интересующий раздел. В том элементе, на который следует перейти, нужно определить уникальный идентификатор.

Для этого существует атрибут `id`, который можно определять практически для любого тега. Название этого атрибута может быть любым и обязательно уникальным. В значении атрибута `href` тега-ссылки ставится символ решётки. После него указывается имя идентификатора, к которому нужно обратиться.

```
<a href="#p10">Прочитать 10 параграф</a>
<p id="p1">Текст параграфа №1</p>
<p id="p2">Текст параграфа №2</p>
...
<p id="p10">Вот и долгожданный параграф №10</p>
```

Пример работы якоря по [ссылке](#).

## Изображения

В настоящее время в веб-документах используются четыре формата изображений: jpeg, gif, png, svg. Рассмотрим каждый из них подробнее.

### JPEG

**Плюсы:**

- небольшой размер файла;
- поддерживает 16 млн цветов;
- можно управлять качеством изображения при сохранении.

**Минусы:**

- при сжатии размеров теряется качество;
- не поддерживает прозрачность.

### GIF

**Плюсы:**



- поддерживает анимацию;
- поддерживает прозрачность;
- при сжатии не теряет в качестве.

#### Минусы:

- количество цветов до 256.

## PNG

Растровый графический формат PNG, становясь популярнее, появился в 1995 году как замена GIF. Формат PNG подразделяется на форматы PNG-8 — аналог формата GIF, кроме того, что PNG-8 не поддерживает анимацию — и PNG-32.

## PNG – 8

#### Плюсы:

- поддерживает прозрачность;
- при сжатии не теряет в качестве.

#### Минусы:

- количество цветов до 256.

## PNG – 32

#### Плюсы:

- При сжатии не теряет в качестве.
- Использует 16 млн цветов.
- Плавный переход от прозрачной области к цветной.

#### Минусы:

- Большой размер файла.

## SVG

SVG – формат векторной графики. Файлы SVG можно читать и редактировать (если есть некоторые навыки) при помощи обычных текстовых редакторов. Есть возможность увеличить любую часть изображения SVG без потери качества.

Растровая графика состоит из разноцветных точек, а векторная — из геометрических фигур. Примечательно то, что векторную графику можно легко перевести в растровую.

Доступно использование растровой графики в SVG-документах. Имеется возможность вставлять элементы с изображениями в форматах PNG, GIF или JPG. Текст в графике SVG — это текст, а не

изображение, поэтому его можно выделять и копировать. Он индексируется поисковыми машинами, не нужно создавать дополнительные метафайлы для поисковых роботов.

#### Плюсы:

- небольшие размеры файлов, отличное сжатие;
- возможно увеличить любую часть изображения SVG без потери качества.

#### Минусы:

- если скопировать фоновое изображение в этом формате, получится большой размер файла (подходит для иконок на сайте);
- сложность использования.

Для сохранения фотографий с чёткими краями лучше всего подходит формат JPEG, так как в нём размер файла получается небольшим. Для анимированных изображений единственный вариант — формат GIF. А для сохранения качественных изображений, в которых используется прозрачность, лучше всего подойдёт формат PNG-32. Векторной графике подходит SVG.

## Загрузка изображений на страницу

Для отображения изображений на странице есть тег `<img>`, он одиночный. У него имеется обязательный атрибут `src`, в значении которого указывают путь к изображению. Этот путь может быть как относительным, так и абсолютным. Второй обязательный атрибут — `alt`.

Атрибут `alt` означает альтернативный текст. Используется, чтобы:

1. Если картинка не загрузится на страницу, пользователь увидит текст, который указали в этом атрибуте.
2. Альтернативный текст нужен при продвижении сайта. В нём можно указать ключевые слова, и тогда это изображение будет участвовать в поиске по картинкам.

```

```

У тега `<img>` можно указать дополнительные атрибуты.

**title** универсален, его можно использовать практически для любого тега. То, что вы напишете в значении, будет выводиться в виде всплывающей подсказки, при наведении на изображение.

Любому изображению можно задать ширину и высоту, указав эти значения в атрибутах **width** и **height** соответственно. При помощи этих атрибутов изображение лучше не увеличивать, иначе получится плохое качество. Если задать ширину и высоту изображению, браузер при загрузке страницы будет

сразу выделять заданную область под картинку. Эти атрибуты используются редко, ведь вся стилизация находится в CSS.

**Важно!** Всегда храните все изображения в отдельной папке, чтобы знать, где искать то или иное изображение.

## Формы и их элементы

Формы — один из важных элементов любого сайта. Они нужны, чтобы получать от пользователя информацию, отправлять эти данные на сервер, а там уже их обрабатывать. Документ может содержать любое количество форм. Но одновременно на сервер может быть отправлена только одна. По этой причине данные форм должны не зависеть друг от друга.

Допускается внутрь контейнера `<form>` помещать другие теги. При этом сама форма никак не отображается на веб-странице, видны только её элементы и результаты вложенных тегов.

**Важно!** В HTML нет возможности реализовать опции форм, для этого используется серверный язык программирования.

## Элементы форм

В HTML существует три основных тега элементов форм:

1. Тег `<input>` в зависимости от значения атрибута `type` будет выглядеть по-разному и иметь различные значения. Тег `<input>` одиночный.
2. `<textarea>` используется, чтобы сформировать многострочное поле ввода для информации. С помощью атрибута `cols` можно задать ширину этого поля, а с помощью атрибута `rows` — указать число строк. Тег `<textarea>` парный.
3. `<select>` представляет собой выпадающий список, элементы которого указываются в теге `<option>`. У этого тега есть несколько атрибутов. Чтобы сделать возможность выбора нескольких пунктов, нужно задать атрибут `multiple`, в значении которого указывается `multiple`. В атрибуте `size` определяется число строк выпадающего списка. Их увидит пользователь в браузере. Если необходимо, чтобы какой-то из элементов выпадающего списка был выбран, нужно этому элементу в теге `<option>` задать атрибут `selected`. Тег `<select>` также парный.

Рассмотрим более подробно тег `<input>`.

```
<input type="text" size="30" placeholder="Ваше Имя">
<input type="password" size="8">
<input type="checkbox" checked="checked">
<input type="radio" name="radio">
<input type="file">
<input type="submit" value="Сохранить">
<input type="reset" value="Очистить">
<input type="button" value="просто кнопка">
```

У первого тега `<input>` значение атрибута `type="text"`, значит, это обычное однострочное поле ввода. В значении атрибута `size` указывается размер этого поля в символах. Атрибут `placeholder` выводит текст внутри текстового поля, который исчезает при начале ввода символов.

**Важно!** *Атрибут `size` не ограничивает количество введенных символов в поле ввода, а только указывает, сколько символов будет видно пользователю при вводе данных.*

`<input>` со значением `password` выглядит так же, как и предыдущий. Но он используется для ввода пароля, и все символы будут показываться как точки, чтобы пароль нельзя было подсмотреть.

Следующие два варианта — `<input type="checkbox">` и `<input type="radio">` — предоставляют пользователю возможность выбора некоторых пунктов.

В первом случае есть возможность множественного выбора нескольких пунктов в виде флажков, а во втором — единственного выбора в виде переключателей. В обоих случаях можно отметить любой пункт по умолчанию, если задать атрибут `checked="checked"`. Помните, что в html5, если название атрибута совпадает с его значением, можно записать сокращённо `<input type="checkbox" checked>`.

`<input type="file">` предоставляет пользователю возможность выбора файла на своём компьютере, чтобы загрузить его на сервер.

Оставшиеся 3 тега `<input>` — кнопки. `submit` отправляет данные на сервер, `reset` предназначена для очистки всех полей в форме. А третья `type="button"` — просто кнопка, при нажатии на которую ничего не произойдёт. Но её, например, можно привязать к событию javascript. У всех кнопок в значении атрибута `value` указывается текст, который будет отображаться на кнопке.

В html5 появилась возможность добавить кнопки с помощью тега `button`.

```
<button type="submit">Отправить</button>
<button type="reset">Очистить</button>
<button>Кнопка html5</button>
```

## Тег `<label>`

Тег `<label>` устанавливает связь между определенной меткой, в качестве которой обычно выступает текст, и элементом формы (`<input>`, `<select>`, `<textarea>`). Такая связь нужна, чтобы изменять значения элементов формы при нажатии курсором мыши на текст. С помощью `<label>` можно устанавливать горячие клавиши на клавиатуре и переходить на активный элемент подобно ссылкам.

Существует два способа связывания объекта и метки. Первый заключается в использовании идентификатора `id` внутри элемента формы и указании его имени в качестве атрибута `for` тега `<label>`. При втором способе элемент формы помещается внутрь контейнера `<label>`.

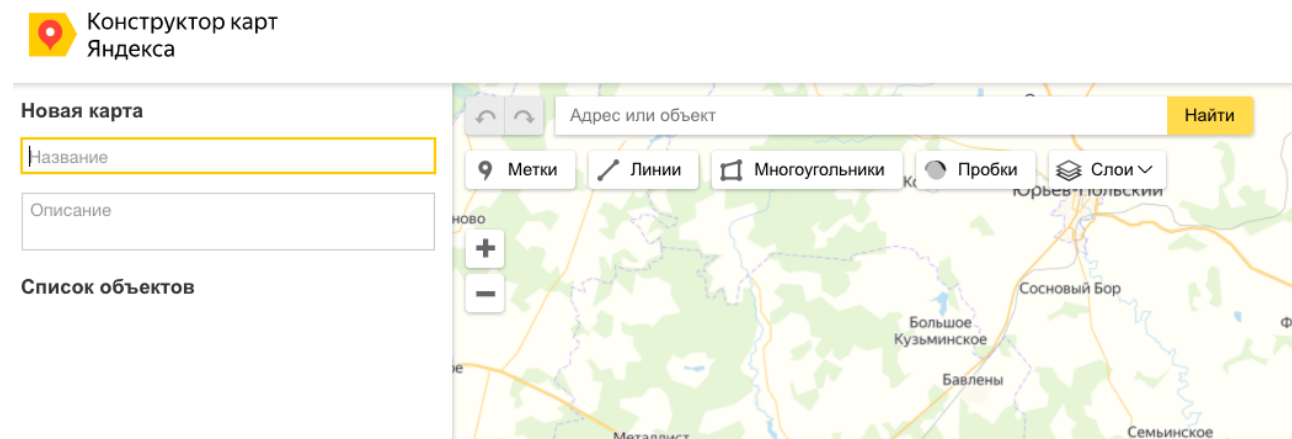
```
<!-- Первый способ -->
<input id="идентификатор" type="checkbox">
<label for="идентификатор">Текст</label>
<!-- Второй способ -->
<label><input type="checkbox">Текст</label>
```

[Пример работы кода](#). Нажмите на текст и убедитесь, что `checkbox` связан с ним.

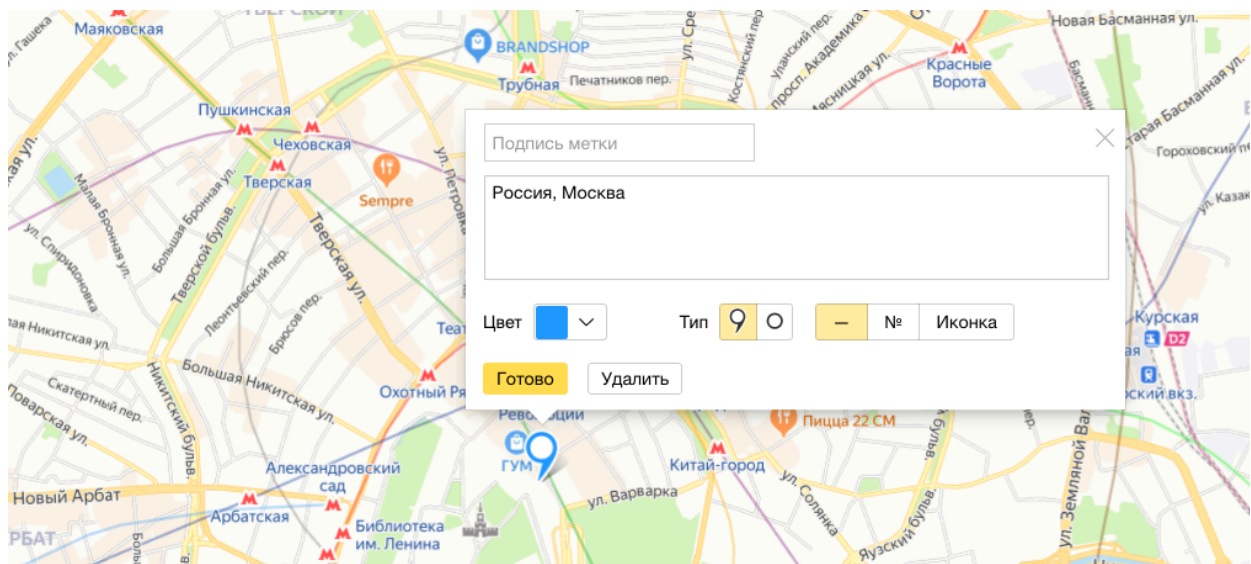
## Добавление интерактивной Яндекс.Карты

Разберём, как можно добавить интерактивную карту на сайт. Для этого не нужно создавать карту абсолютно с 0 или использовать какие-то языки программирования. В этих целях используются [конструкторы](#), рассмотрим один из них.

[Конструктор карт Яндекс](#) [Конструктор карт Яндекс](#)



1. Можете придумать название вашей карты и описание.
2. Далее нужно ввести адрес или объект, который хотите добавить на сайт.
3. Нажать клавишу «Найти».



4. \* По желанию можно настроить цвет, тип и иконку отображения.
5. Нажать клавишу «Готово».
6. Кликнуть на клавишу «Сохранить и продолжить», которая располагается в левом нижнем углу конструктора.

Карта готова

Экспорт

Вставить на сайт

Распечатать

☒ Интерактивная карта
 ☐ Статическая карта

Такую карту можно двигать, а объекты отзываются на действия пользователя

Размеры карты:

500

×

400

px

☐ Растянуть по ширине

Получить код карты

Ссылка на карту:

https://yandex.ru/ma

📍

7. Можно настроить размеры карты или растянуть её по ширине.
8. Нажать клавишу «Получить код карты».
9. Добавить код в ту часть сайта, где хотите видеть карту у себя на странице.

## Практическое задание

Продолжаем работу над проектом, который начали делать на первом уроке.

1. Шапка сайта (на каждой странице):
  - Меню сайта:
    1. Главная.
    2. Контакты.
2. Всё содержимое главной страницы из ПЗ 1.
3. После элемента `h1` (из первого ПЗ) добавить список контактов, содержащий 3 элемента списка.
  - произвольный текст не более 5 слов
  - номер телефона;
  - адрес электронной почты.
4. Создать страницу «Контакты». При создании папок и страниц использовать только английские буквы.
5. Страница «Контакты»:
  - шапка сайта;
  - заголовок `Contact Us` (`h2`);
  - параграф с произвольным текстом: не более 10 символов;
  - произвольное изображение (не более 350px);
  - заголовок `Address` (`h4`);
  - параграф с произвольным текстом: не более 7 символов;
  - спсзаголовок `Phone` (`h4`);
  - список из 2 номеров телефонов;
  - заголовок `Online service` (`h4`);
  - список из 2 ссылок на сайт и почту;
  - произвольный адрес и номер телефона;
  - заголовок `Send us message` (`h4`);
  - текст `Full Name` и текстовое поле ввода;
  - текст `Email` и поле ввода почты;
  - текст `Message` и элемент `textarea`;
  - кнопка отправки с текстом `submit`;
  - интерактивная Яндекс.Карта;
  - подвал сайта: скопировать с главной страницы, заново создавать элементы не нужно, ниже перечислено, что нужно скопировать.
    1. Заголовок (`h3`): не более 3 слов.
    2. Параграф с произвольным текстом: не более 10 слов.
    3. Разделительная линия (`hr`).
    4. Список контактов.
      - произвольный текст — не более 5 слов;
      - номер телефона;
      - адрес электронной почты.

5. Параграф с текстом «Все права защищены».

## Дополнительные материалы

1. [Статья про типограф, с чего всё начиналось и для чего же он нужен.](#)
2. [Чуть подробнее про гипертекст.](#)
3. [Какие изображения применяются для html.](#)
4. [Формы html.](#)

## Используемые источники

1. <http://htmlbook.ru/samhtml/tekst/spetssimvoly>.
2. <http://htmlweb.ru/html/symbols.php>.
3. <http://htmlbook.ru/html/a>.
4. <http://htmlbook.ru/html/img>.
5. <http://htmlbook.ru/html/form>.
6. <https://html5book.ru/osnovy-html/>.