Для работы рекомендуется использовать приложение *VSCode* и *node.js*. В *VSCode* рекомендуется поставить следующие расширения:

* *Code Runner* для быстрого запуска
* *LiveServer* для автоперезагрузки страницы
* *Auto Rename Tag* для автозамены закрывающегося тега при редактирования открывающегося
* *W3C Web Validator* для проверки кодстайла
* *Prettier* для автоформатирования кода
* *EditorConfig* для замены табуляции на два пробела
* *PostCSS Sorting* для задания правила сортировки свойств (запуск через *F1 -> PostCSS Sorting: Run*)

Также рекомендуется включить автосохранение документа, настроить файл *User Settings (JSON)* или скопировать настройки из приложенного файла ***settings.json***.

Для сравнения макета и свёрстанного проекта используются несколько приемов:

1. Добавить макет в разметку, сделать полупрозрачным и поменять цвета.

*.pixel-perfect-image {*

*position: fixed;*

*top: 0;*

*left: 50%;*

*transform: translateX(-50%);*

*opacity: 0.5;*

*filter: sepia(0.6);*

*}*

1. Использовать плагин *PerfectPixel*.

Для комментирования строк используются специальные символы:

*<!-- в HTML-коде комментарий окружён такими символами -->*

*/\* в CSS — такими \*/*

**Относительный путь** – начинается с точки (“***./***” или “***../***”), указывает путь относительно текущего места.

**Абсолютный путь** – начинается со слеша “***/***”, указывает путь от корня проекта.

***Hyper Text Marker Language*** – язык *HTML*, используется для создания каркаса сайта.

*HTML*-элемент состоит из открывающего и закрывающего тегов, а также размещенного между ними содержания.

**Атрибуты** (или **свойства**) сообщают дополнительные сведения об элементе. Атрибут встраивается только в открывающий тег. Атрибуты пишутся подряд через пробел, порядок не имеет значение. Значение атрибута приводят в ‘одинарных’ или “двойных” кавычках (без разницы). Для *HTML* рекомендуется использовать двойные кавычки, а для *CSS* – одинарные.

Все возможные атрибуты элементов можно посмотреть на <https://doka.guide/>

***<!DOCTYPE html>*** - одиночный тег обозначающий, что перед нами *html*-страница.

***<head lang=”ru”> </head>*** - теги метаданных страницы. Необязательный универсальный атрибут ***lang*** подсказывает поисковым системам, скринридерам и расширениям перевода на каком языке написана страница (в теге *html*) или отдельные блоки текста. В качестве значений используется двухбуквенный код языка, который можно посмотреть [тут](https://www.iana.org/assignments/language-subtag-registry/language-subtag-registry).

***<meta charset=”UTF-8” />*** - одиночный тег для подключения кодировки UTF-8.

***<meta name="viewport" content="width=device-width, initial-scale=1" />*** - одиночный тег, который сообщает браузеру две инструкции. Первая ***width=device-width***, что видимая пользователю часть занимает всю ширину экрана. Вторая ***initial-scale=1***, что страница не масштабируется при загрузке.

***<meta name="description" content="Описание страницы" />*** - описание страницы для поисковиков.

Формат ***Open Graph*** (или ***OG***) позволяет сформировать превью страницы в социальных сетях при передаче ссылки на неё. Подробнее можно почитать [тут](https://ogp.me/).

***<meta property="og:title" content="Можно мягче" />***

***<meta property="og:url" content=”https://maybesofter.com/” />***

***<meta property="og:image" content="/images/og.jpg" />***

***<meta property="twitter:card" content="summary\_large\_image" />*** *// у twitter есть свои теги*

**Фавикон** – это иконка вкладки страницы.

***<link rel=”icon” href=”favicon.ico” sizes=”any” />*** - одиночный тег для подключения фавикона в формате ***ICO*** (32х32 пикселей) для старых браузеров.

***<link rel=”icon” href=”icon.svg” type=”image/svg+xml” />*** - одиночный тег для подключения фавикона в формате ***SVG*** для актуальных браузеров.

***<link rel=”apple-touch-icon” href=”apple-touch-icon.png” />*** - одиночный тег для подключения фавикона в формате ***PNG*** (512х512 пикселей) для иконки смартфонов и планшетов, если пользователь захочет добавить его на экран.

***<title></title>*** - теги названия вкладки страницы.

***<link rel=”stylesheet” href=”./fonts/fonts.css” />*** - одиночный тег для подключения шрифтов.

***<link rel=”stylesheet” href=”./styles/global.css” />*** - одиночный тег для подключения глобальных таблиц стилей.

***<link rel=”stylesheet” href=”./ styles/styles.css” />*** - одиночный тег для подключения таблиц стилей.

***<h1></h1>*** - теги заголовка. Всего их 6 размеров, *<h6>* самый маленький.

***<p></p>*** - теги абзаца. В них заключается основная часть текста.

***<a></a>*** - теги ссылки-якоря (anchor). Необязательный атрибут ***href*** хранит адрес ссылки.

*<a href=”https://takiedela.ru/”>Перейти по ссылке</a>*

Необязательный атрибут ***target*** открывает ссылку в этой вкладке или в новой.

*<a target=”\_blank” href=”https://takiedela.ru/”>Перейти по ссылке</a>*  *// в новой вкладке*

***<img>*** - тег изображения. Создается только одним открывающимся тегом с обязательным атрибутом ***src***.

*<img src=”images/mobile.jpg”>*

Необязательный атрибут ***alt*** отображает текст, который пользователь видит, когда картинка не загружается:

*<img src=”images/mobile.jpg” alt=”здесь должна быть картинка, но что-то пошло не так…”>*

Необязательный атрибут ***width*** позволяет задать ширину картинки в пикселях. Это также позволит избежать рывков страницы при медленной загрузке изображений.

*<img src=”images/mobile.jpg” width=230>*

***<q></q>*** - теги цитирования фразы прямо внутри текста. Заключают текст в кавычки, вид которых зависит от языка документа.

***<blockquote></blockquote>*** - теги для блочного цитирования фразы. Необязательный атрибут ***cite*** позволяет указать автора цитаты.

*<blockquote cite="https://praktikum.yandex.ru">Достаточно начать и не бросить.</blockquote>*

***<body></body>*** - теги тела страницы (видимой части).

***<div></div>*** - универсальные теги элемента-блока (абстрактная пустая коробка), позволяющие объединять другие элементы. Расположен в *body*.

***<header></header>*** - теги элемента-блока шапки страницы, позволяющая объединять другие элементы (меню, логотип, заголовок страницы, вводная информация). Расположен в *body* и занимает всю ширину страницы.

***<section></section>*** - теги элемента-блока страницы по аналогии с *div*, но лучше читается поисковиками. Подходят для разделения контента на большие логические блоки. Расположен в *body*.

***<footer></footer>*** - теги элемента-блока подвала страницы, позволяющая объединять другие элементы (дополнительная навигация, копирайт, ссылки на партнеров, другая доп.информация). Расположен в *body*.

***<nav></nav>*** - теги элемента-блока навигации по сайту. В него принято оборачивать ссылки навигации по сайту или отдельной странице. Расположен в *body*.

***<main></ main >*** - теги элемента-блока основного содержимого страницы. На многостраничных сайтах, этот контейнер содержит уникальный контент одной страницы. Расположен в *body*.

***<aside></ aside >*** - теги элемента-блока дополнительной информации, часто оформляется как боковая колонка. Расположен в *body*.

***<article></ article>*** - теги элемента-блока, не привязанного к окружающему его контенту. Такой блок можно убрать со страницы и вставить в любое другое место сайта.

В *HTML* существуют два вида списков – маркированный и нумерованный.

***<ul></ul>*** - теги маркированного списка, оформленного маркерами в виде точек.

***<ol></ol>*** - теги нумерованного списка.

***<li></li>*** - теги элементов любого списка.

Необязательное *CSS*-свойство ***list-style*** отвечает за управление маркерами и нумерацией списков. Например, список можно оформить так:

*<ul style=” list-style: none”> // данный стиль вообще скрывает маркер*

*<li>*

*<div>Я на своём месте =)</div>*

*</li>*

*</ul>*

***<button></ button>*** - теги простой кнопки.

***<table></table>*** - теги элементов таблицы.

***<caption></caption>*** - теги краткого описания таблицы. Идет сразу после *<table>*.

***<tr></tr>*** - теги задания строки таблицы.

***<th></th>*** - теги задания ячейки первой строки таблицы.

***<td></td>*** - теги задания ячейки остальных строк таблицы.

***<thead></thead>*** - теги элемента-блока шапки таблицы.

***<tbody></tbody>*** - теги элемента-блока тела таблицы. Можно использовать несколько в одной таблице.

***<tfoot></tfoot>*** - теги элемента-блока подвала таблицы.

Необязательные атрибуты ***rowspan*** и ***colspan*** помогают растянуть одну ячейку на несколько по вертикали и горизонтали соответственно. Соседние ячейки при этом смещаются.

Например, таблицу можно оформить так:

*<table>*

*<caption>*

*Три романа с большим количеством букв*

*</caption>*

*<thead>*

*<tr> // первая строка с заголовками*

*<th>Название книги</th>*

*<th>Автор</th>*

*<th>Количество слов</th>*

*</tr>*

*</thead>*

*<tbody>*

*<tr> // вторая и последующие строки с данными таблицы*

*<td>Келидар</td>*

*<td>Махмуд Довлатабади</td>*

*<td>950 000</td>*

*</tr>*

*</tbody>*

*<tfoot>*

*<tr>*

*<td colspan="2">Среднее количество слов</td> // растянуть ячейку на две в ширину*

*<td>1502009.3</td>*

*</tr>*

*</tfoot>*

*</table>*

К таблице также можно применять стили. Например:

*td, th {*

*border: 1px solid #000; // нарисовать рамку вокруг каждой ячейки*

*}*

*table {*

*background-color: #fff; // задать фоновый цвет таблицы*

*border-collapse: collapse; // убрать расстояние между рамками ячеек таблицы*

*}*

**Якорные ссылки** – элемент навигации на сайте, позволяющие переместиться к заданному блоку с атрибутом идентификатором ***id***. Значение атрибута не может начинаться с цифр или спец.символов, и не должно содержать пробелов. Для идентификаторов можно задавать *CSS* селекторы стилей, но не рекомендуется из-за сложности поддержки.

Если якорная ссылка и элемент, на который она ссылается, находятся на одной странице, то в качестве адреса ссылки нужно прописывать ***#имя\_идентификатора***. Чтобы сослаться на элемент с любой другой страницы, нужно указать ***полный\_адрес#имя\_идентификатора***.

*<section class="section about" id="about">*

*<a href="#about">Что это</a>*

*<a href="https://atychiphobia.com/article#about">Якорная ссылка с другой страницы</a>*

Для плавной прокрутки страницы при переходе по якорной ссылке необходимо в *CSS* стилях создать селектор *html* с атрибутом ***scroll-behavior*** и значением *smooth*:

*html {*

*scroll-behavior: smooth;*

*}*

**Фрейм** – это окошко, в котором видно другую страницу сайта. Для его создания нужны теги ***iframe***. Технически фреймы ведут себя в потоке как *div*.

*<iframe src=”https://ya.ru/” frameborder=”0”></iframe>*

Атрибут ***frameborder*** со значением 0 позволяет отключить серую рамку фрейма.

Свойство ***boarder*** позволяет модифицировать или отключить серую рамку фрейма.

***Application Programming Interface*** (или ***API***) – это интерфейс взаимодействия между сайтом и сторонними программами и серверами.

Рассмотрим пример с подключением *API Youtube* через фрейм. Для этого используется ссылка вида *https://www.yotube.com/embed/идентификатор\_ролика*.

*<iframe*

*src="https://www.youtube.com/embed/RhlQvbvMg-0?start=30&mute=true" // без звука и с 30й секунды*

*title="YouTube video player"*

*frameborder="0"*

*allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"*

*allowfullscreen*

*></iframe*>

Рассмотрим пример с подключением Яндекс.Карт через фрейм. Для этого воспользуемся специальным конструктором *https://yandex.ru/map-constructor/*.

*<iframe*

*src="https://yandex.ru/map-widget/v1/?um=constructor%3A6fc6359a9fea00ada5a4032946d1df7ac9ad308d11f5e45b29cf78e2bdf6a738&amp;source=constructor"*

*width="100%"*

*height="500"*

*frameborder="0"*

*></iframe>*

Рассмотрим пример подключения своего видео без использования сторонних сервисов. Для этого нам нужны теги ***video***. Не рекомендуется позиционировать видео относительно элемента html, т.к. на некоторых размерах экранов будет неправильно высчитывать ширину в 100%, и вырвать видео из потока ***position:relative***.

Атрибут ***controls*** позволяет отобразить кнопки управления видео.

Атрибут ***muted*** позволяет выключить звук видео.

Атрибут ***autoplay*** позволяет включить автовоспроизведение видео. Поддерживается не всеми браузерами и работает в связке с ***muted***.

Атрибут ***loop*** позволяет включить зациклить видео по кругу.

Атрибут ***poster*** позволяет установить картинку, которую будет видеть пользователь, пока не запустит воспроизведение или пока видео загружается.

*<video src="video.mp4" controls muted autoplay loop poster=”poster.png”></video>*

Для подключения нескольких видео используется одиночный тег ***source***. В первую очередь нужно подключать более современные типы файлов, которые указываются в атрибуте ***type***. Браузер возьмет первый файл, который сможет проиграть, и не будет скачивать остальные файлы.

*<video controls muted loop poster=”poster.png”>*

*<source src="video.webm" type="video/webm" />*

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

*</video>*

Свойство ***object-fit*** позволяет задать как видео будет подстраиваться под заданные размеры.

*object-fit: fill; // по умолчанию, вписывается без сохранения пропорций*

*object-fit: contain; // вписывается под заданные размеры без нарушения пропорций*

*object-fit: cover; // вписывается без нарушения пропорций, обрезая все лишнее*

*object-fit: none; // отображается без изменений пропорций или размеров*

*object-fit: scale-down; // браузер сам выбирает между none и contain*

Свойство ***object-position*** позволяет выровнять изображение по осям *X* и *Y* после подгонки по размерам.

*object-position: top right; // сначала по X, потом по Y*

Рассмотрим пример подключения своего аудио без использования сторонних сервисов. Для этого нам нужны теги ***audio***. Они имеют такие же атрибуты как у тегов ***video***.

*<audio>*

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

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

*</audio>*

Как сделать свой видеоплеер можно почитать [тут](https://habr.com/ru/company/microsoft/blog/127295/).

**При резиновой верстке** элементы растягиваются: пропорционально масштабируются под размеры окна, но не меняют свое положение и внешний вид. Размеры задаются в относительные единицы измерения *%*, *vw*, *vh*.

**При адаптивной верстке** внешний вид сайта меняется, а элементы перестраиваются в зависимости от размера браузера. Пишутся *CSS*-правила под разные условия (размеры окна).

**При отзывчивой верстке** совмещается резиновая и адаптивная верстка. Благодаря резиновой верстке переход из одного состояния в другое происходит плавно, а благодаря адаптивной верстке в определенных точках перестраивается внешний вид.

При любом из подходов для сайта верстают несколько состояний: для телефона, планшета, ноутбука, десктопа. Используются подходы ***desktop first*** и ***mobile first***.

**Векторная графика** – это изображения формата *svg*, которые состоят из линий. Их можно растягивать без потери качества.

**Растровая графика** – это изображения формата *jpeg*, *png*, которые состоят из пикселей. При растягивании качество теряется, становятся видны пиксели.

В изображениях в формате *png* присутствует альфа-канал (может быть прозрачный фон или элементы).

Формат *gif* используется для коротких зацикленных анимаций. Много весит.

Для ускорения загрузок изображений особое внимание нужно уделить их оптимизации:

* Использовать подходящий формат. Для простых картинок это *png*, а не *jpeg*.
* Использовать нужный размер. Не нужно отдавать картинку размером 2000х2000 в блок 500х500.
* Использовать для разных разрешений экранов разные изображения.
* Оптимизировать вес изображения через сжатие в онлайн-инструментах [TinyJPG](https://tinyjpg.com/) или [Sqoosh](https://squoosh.app/).

Пока картинка грузится используют цветной фон при помощи *background-color*.

**Физический пиксель** – самая маленькая точка цвета, которая может быть отображена на устройстве.

**Пиксель в *CSS*** – физическая величина, которая определяет сколько места на экране будет занимать элемент. Может соответствовать х1, х2 и х3 физическому пикселю.

В зависимости от плотности пикселей на экране нужно подгружать картинки в разных размерах.