**Готовый кейс «С чистого листа»**

## **Базовая настройка проекта**

Вы получили [файл с заготовкой проекта](https://code.s3.yandex.net/web-developer/verstka/zipki/s-chistogo-lista-main.zip). Изображения и шрифты лежат в соответствующих папках.

Почти все проекты по вёрстке начинаются одинаково — с создания необходимых файлов и связывания их друг с другом. Создайте индексный HTML-файл и файл стилей в папке styles. Назовите файл стилей style.css. Это поможет автотестам при сдаче проекта работать корректно.

Затем создайте внутри папки fonts файл fonts.css . В нём настройте все шрифты во всех форматах. Вам понадобятся такие шрифты:

* EB Garamond со значениями толщины 500 (файлы с именами ebgaramondmedium) и 400 (файлы с именами ebgaramondregular).
* Inter со значениями толщины 400 (файлы с именами interregular) и 700 (файлы с именами interbold).

Создайте базовую структуру HTML-страницы и подключите ссылки на файлы стилей: сначала на файл со стилями шрифтов, затем — на файл со стилями проекта.

Ваша страница будет написана на русском языке, поэтому в атрибуте lang укажите значение ru. Ещё поменяйте title. Рекомендуем такой — С чистого листа.

## **Сбросьте стили, которые помешают в будущем**

Браузерные отступы вам не понадобятся. Предлагаем сбросить их на элементах, которые встречаются чаще других.

Некоторые верстальщики не любят сбрасывать стили глобально и писать аналоги reset.css (normalize.css), но в проекте мы это сделаем, чтобы меньше думать о браузерах и больше о дизайне.

Установите значение 0:

1. Свойству margin для body;
2. Свойствам margin и padding для h1, h2, h3, h4, h5, h6, p, ul, ol, blockquote — можете взять такой селектор.

Ещё вы можете задать body фон, чтобы отличать его от мобильной страницы. Мы выбрали фон #1e1e1e, вы можете выбрать свой.

## **Найдите паттерны и опишите их**

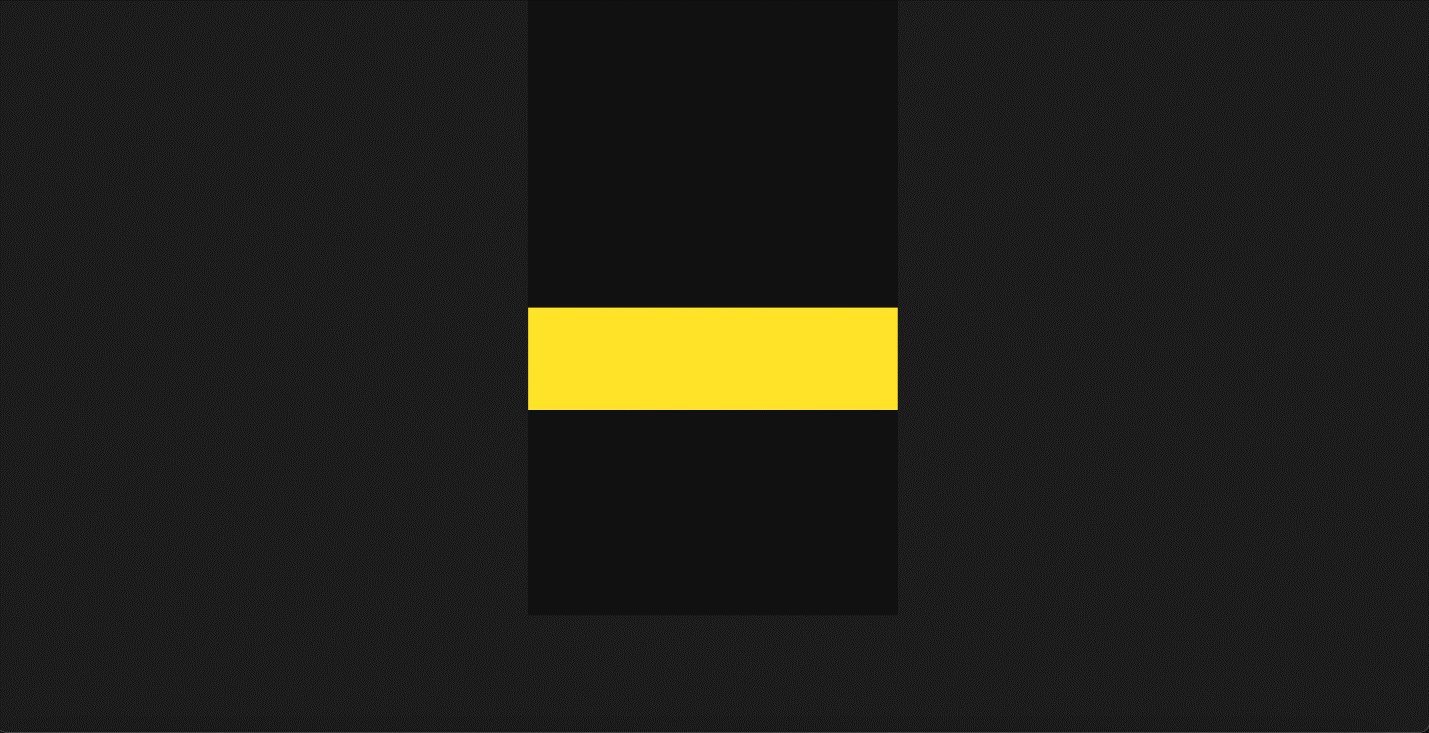
Посмотрите на [макет](https://www.figma.com/file/vYJfYCZUddsUQUCYUktcID/%231-С-чистого-листа/duplicate) и найдите повторяющиеся детали.

Вначале это непросто, но с опытом такое умение приходит. В этом проекте мы подскажем.

1. На странице два шрифта. По объему текста сильно преобладает Inter с толщиной 400. Можно смело назначить его как основной шрифт страницы для всего тела документа. Сделайте это. Уникальные шрифты будем добавлять каждому конкретному элементу. Не забывайте указывать альтернативный шрифт в дополнение к основному.
2. Страница состоит из шести секций: шапки, четырёх секций с контентом и подвала. Четыре секции с контентом можно считать основной зоной сайта. Вы можете написать семантическую HTML-разметку для перечисленных зон. Используйте теги header, main, section и footer.



1. Все секции одинаковой ширины — 360 пикселей. Ещё их стоит выровнять по горизонтальному центру, чтобы было удобнее смотреть. У них будут внутренние отступы, поэтому можно изменить box-sizing, чтобы не раздувать блоки отступами. Создайте в CSS селектор по классу content-section , опишите в нем правила ширины, автоматические поля и определите box-sizing. Этот класс можно присвоить всем элементам в HTML, кроме main.
2. Есть два вида секций: с чёрным фоном и белым шрифтом, с желтым фоном и чёрным шрифтом. Можем создать два класса-модификатора: content-section\_theme\_dark под первый вид и content-section\_theme\_bright под второй вид (имена не случайны: чёрный там не совсем чёрный, а жёлтый — не самый жёлтый). Этим модификаторам можно задать правила фона и цвета текста. Чтобы увидеть изменения на странице, можно задать секциям любую высоту, например 100 пикселей, и потом ее убрать. Расставьте эти классы нужным блокам в HTML и посмотрите результат.



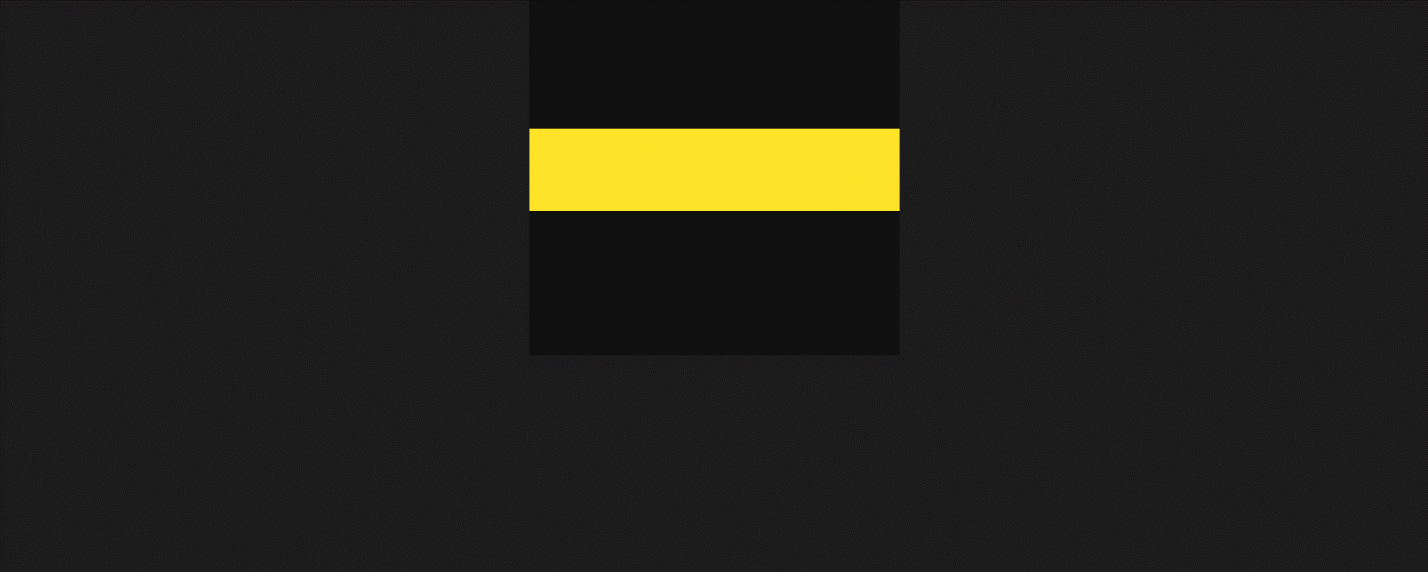
Получилась такая заготовка. Кажется, можно наполнять секции элементами.

## **Дайте секциям уникальные имена и настройте внутренние отступы**

Каждая из секций в чём-то уникальна, поэтому полезно использовать классы, которые подчеркнут это и определят персональные стили секций. Мы назвали их так: header; секции cover, intro, reasons и how-to; footer. Названия отражают содержание секций.

Через уникальные имена удобно задавать внутренние отступы. Хотя в их размерах и можно найти общие закономерности, пусть каждая секция отдельно управляет своим внутренним устройством. Например, для header заданы отступы в 24 пикселя сверху и снизу и по 20 пикселей по бокам.

Остальные отступы определите и расставьте сами. После этого секции будут видны и без указания высоты:



Секции уменьшились без правила min-height, но стали видны за счёт отступов

🌐 Дальше вам предстоит сверстать содержимое секций. Мы дадим подсказки по особенностям макета, но решения по вёрстке, измерениям, цветам и контенту вам необходимо забирать из Figma самостоятельно. Именно такая практическая работа поможет по-настоящему научиться верстать.

## **Шапка сайта**

В шапке два элемента — логотип и меню. Обычно на внутренних страницах логотип — это ссылка на главную страницу. На главной странице эту ссылку делают неактивной. Чтобы добиться нужного поведения для главной страницы, вложите изображение с логотипом внутрь тега ссылки, но уберите у ссылки атрибут href. Так ссылка станет контейнером без логики перехода.

Меню лучше оформить семантическим тегом навигации, но внутрь него вложить список. Это поможет скринридеру рассказать людям с особенностями зрения, сколько всего ссылок в меню и какая находится в фокусе (про фокус ещё расскажем). Не забывайте также указывать alt для изображений.

Задайте отдельный класс для выделения активной ссылки. Чтобы страница не перезагружалась по клику на ссылку, присвойте атрибуту href значение #0 (лайфхак с несуществующим идентификатором).

Рекомендуемые имена классов:

* header\_\_logo — ссылка-логотип;
* header\_\_logo-image — картинка логотипа;
* header\_\_menu — контейнер с меню;
* header\_\_menu-list — список пунктов меню;
* header\_\_menu-item — пункт меню;
* header\_\_menu-item\_active — активный пункт меню;
* header\_\_menu-link — ссылка в пункте меню.

Все остальные решения о стилизации принимать вам. Главное, чтобы страница совпадала с макетом.

## **Секция cover**

В HTML достаточно добавить один элемент — заголовок первого уровня. Внутри него нужно разбить текст на три строки. Это лучше сделать, заключив контент каждой строки в отдельный тег span.

По правилам вложенности внутрь заголовка нельзя положить, например, абзац. А вот строчный span вкладывать можно. Проверяйте правила вложенности с помощью сервиса [CAN I INCLUDE](https://caninclude.glitch.me/).

Чтобы каждый span оказался на своей строке, достаточно сделать его блочным через свойство display.

Выравнивание внутри строк удобно разметить отдельными классами, которые пригодятся для такого же выравнивания в других заголовках (в статьях есть выравнивание по правому краю).

Изображение нужно разместить на фоне блока cover. Поэкспериментируйте со свойствами background-repeat, background-size, background-position, чтобы расположить фон как на картинке. Попробуйте, например, указать размер фона в 95% auto. Блоку cover задайте минимальную высоту, чтобы он не схлопывался.

Рекомендуемые имена классов:

* cover\_\_title — заголовок;
* accent-span — все строки;
* text-center — строка с выравниванием текста по центру;
* text-right — строка с выравниванием текста по правому краю.

## **Секция intro**

Вам понадобятся два контейнера — один для всех элементов заголовка статьи, другой — для основного текста.

Для указания даты публикации используйте семантический тег time. Подробнее о нем [почитайте на Доке](https://doka.guide/html/time/). Даты обычно не переносят между строками, используйте для этого white-space: nowrap;.

В контейнере для текста не забывайте размечать содержимое абзацами. Не используйте перенос строк, когда абзацы разные.

Как обычно поработайте с флексами, шрифтами, цветами и отступами.

Рекомендуемые имена классов:

* intro\_\_heading — контейнер для элементов заголовка;
* intro\_\_title — заголовок;
* intro\_\_date — дата публикации;
* intro\_\_main-text — контейнер для текста.

## **Секция reasons**

Здесь три элемента — заголовок, абзац и список.

Для выравнивания строк в заголовке удобно использовать те же классы и приёмы, что в секции cover.

Внутри пункта списка нужен дополнительный span, чтобы выделить ключевые слова жирным начертанием.

Для заголовка используйте тег, описывающий корректную структуру: это всё одна статья, заголовок в секции reasons должен быть ниже уровнем, чем в секции intro.

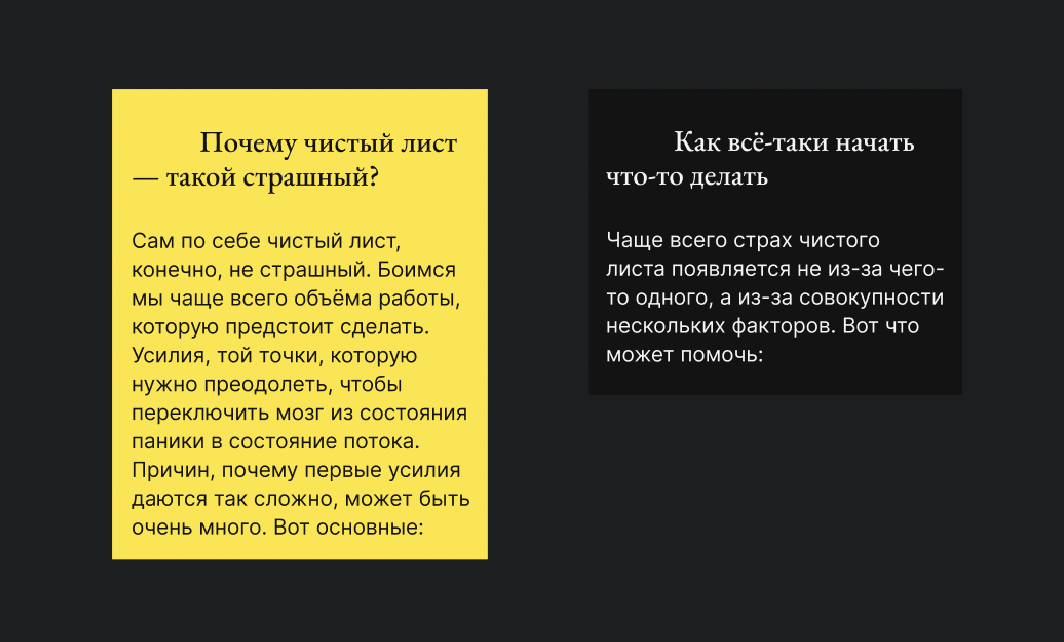
Когда будете верстать список, подумайте, как можно избежать лишнего отступа у последнего пункта (если используете нижние поля) или у первого (если верхние). Сейчас вы можете это сделать только через flex, но скоро изучите другие, более популярные приёмы.

Рекомендуемые имена классов:

* content-section\_\_subtitle — заголовок;
* accent-span, text-right — знакомы вам по секции cover;
* content-section\_\_introduction — абзац перед списком;
* content-section\_\_list — список;
* content-section\_\_list-item — пункт списка;
* content-section\_\_list-accent — выделенный жирным начертанием текст внутри списка.

## **Секция how-to**

Эта секция большая, но есть хорошие новости. Два элемента вы уже верстали:



Они одинаковые, отличаются только тексты и цвет, который наследуется от темы секции

Вы можете взять эти элементы из прошлой секции, заменив контент внутри. А вот над списком карточек придется потрудиться. Читайте буквально — это **список** карточек.

Для карточки используйте тег article, вложенный в пункт списка.

Карточки двух видов — серые и жёлтые. Для каждого из них полезно создать по два класса: один для общих стилей, второй — для цветовых модификаций. Общие стили описывайте в классе card, а стили для цветовых модификаций — классамcard\_theme\_dark и card\_theme\_brightсоответственно.

Вам потребуется два контейнера внутри карточки — под элементы, связанные с автором высказывания в карточке, и под текст внутри карточки. Нужно всё сделать так, чтобы можно было убрать первый контейнер и ничего не сломать. Внутри жёлтой карточки нет информации об авторе.

Внутри контейнера с информацией об авторе два элемента — изображение и имя автора, а внутри контейнера с текстом — заголовок и текст. Если абзацев несколько, обычно создают дополнительный контейнер-обёртку.

Присмотритесь к отступам. Когда в карточке нет картинки, у заголовка нет верхнего отступа. Для таких случаев можно использовать дополнительный класс, обозначающий карточку без фото и вложенный селектор для выбора заголовка в ней.

Рекомендуемые имена классов:

* content-section\_\_subtitle, accent-span, text-right, content-section\_\_introduction — эти классы вам уже знакомы;
* cards-list — список карточек;
* card — карточка;
* card\_text-only — карточка без фото
* card\_theme\_dark и card\_theme\_bright — цветовые модификаторы карточек;
* card\_\_author — контейнер с информацией об авторе;
* card\_\_author-photo — фото автора;
* card\_\_author-name — имя автора;
* card\_\_content — контейнер для контента карточки;
* card\_\_title — заголовок перед текстом карточки;
* card\_\_text — контейнер для текста карточки.

## **Подвал сайта**

В подвале сайта два элемента — абзац с копирайтом и список ссылок. После всего свёрстанного выше вы легко с ним справитесь.

Рекомендуемые имена классов:

* footer\_\_copy — блок с копирайтом;
* footer\_\_links — список ссылок.