

Что такое макет

Вспомним материал первого урока. Мы рассуждали на тему разработки сайтов и говорили, что в разработке участвует не только веб-разработчик, но ещё и веб-дизайнер. Он рисует внешний вид сайта, используя специальную программу, благодаря которой получается изображение сайта — макет.

Для чего необходимы макеты в вёрстке

Макет отличается от картинки сайта тем, что в нём собрана вся необходимая информация для вёрстки проекта: весь контент, размеры блоков, изображения, которые потребуются для создания сайта. Самое основное — в макете уже представлен CSS для будущего сайта. Простыми словами, благодаря макету вы сможете получить всё содержимое для вашего проекта, вы знаете, как выглядит сайт и какого результата вы должны добиться.

Какие виды макетов бывают

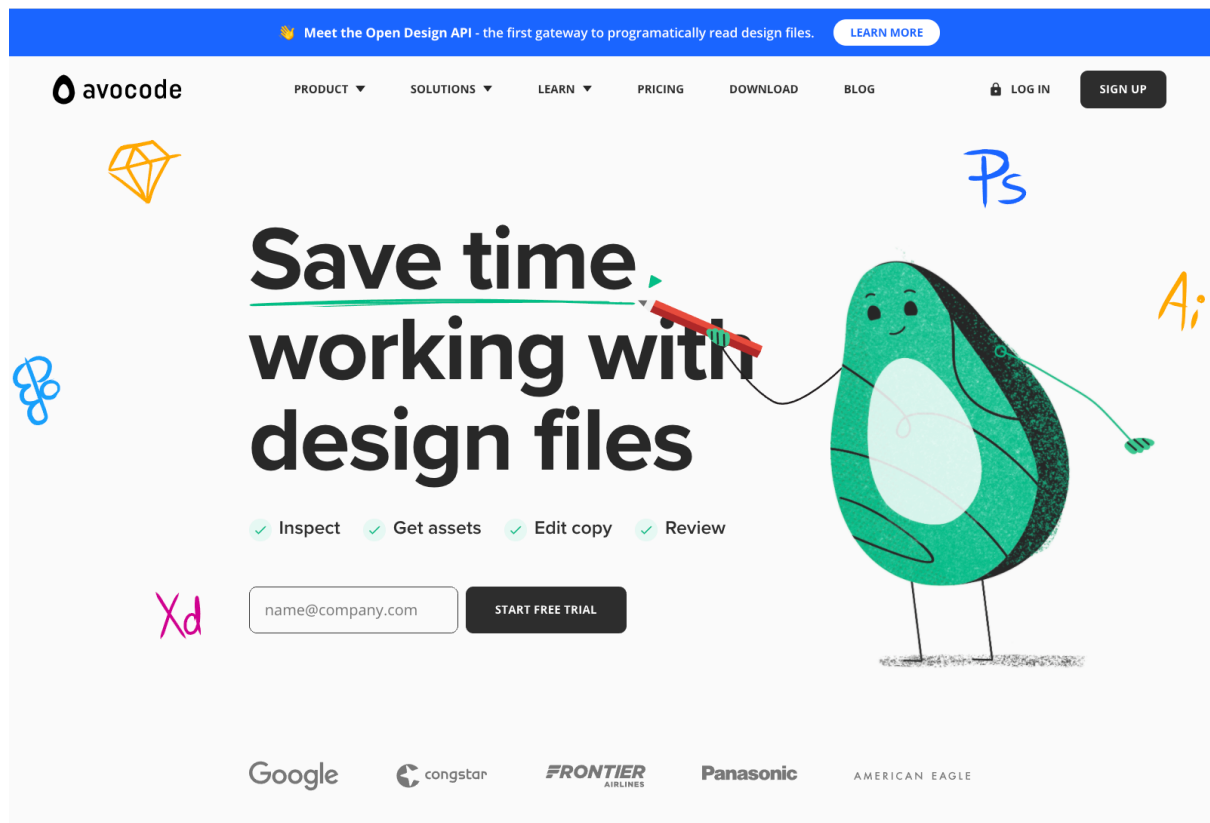
Графических редакторов много: Photoshop, Figma, Sketch, Illustrator, и это далеко не весь список. У верстальщика может возникнуть вопрос, какой же графический редактор изучать и нужно ли изучать их все. Если вы захотите изучить все эти редакторы, то вы, скорее всего, станете дизайнером, а не веб-разработчиком, поэтому нам необходимо искать альтернативные варианты работы с макетом.

Как открыть любой макет для верстки

Наша задача — работа с большинством графических редакторов, но без изучения каждого из них. Для этого нужен инструмент «визуальная вёрстка» — это конвертеры, которые смогут преобразовать макет сайта практически в любом формате в удобный для вёрстки внешний вид. Один из самых популярных вариантов работы с макетом — [Avocode](#).

В нём вы как раз сможете конвертировать Photoshop, Figma, Sketch, Illustrator, Adobe XD в правильный внешний вид именно для вёрстки проекта. Тут вы сможете извлечь все необходимые стили, получить буквально за пару кликов всю контентную часть и параметры блоков. Минус Avocode — это платный проект, поэтому предлагаем рассмотреть несколько альтернативных вариантов.

Avocode

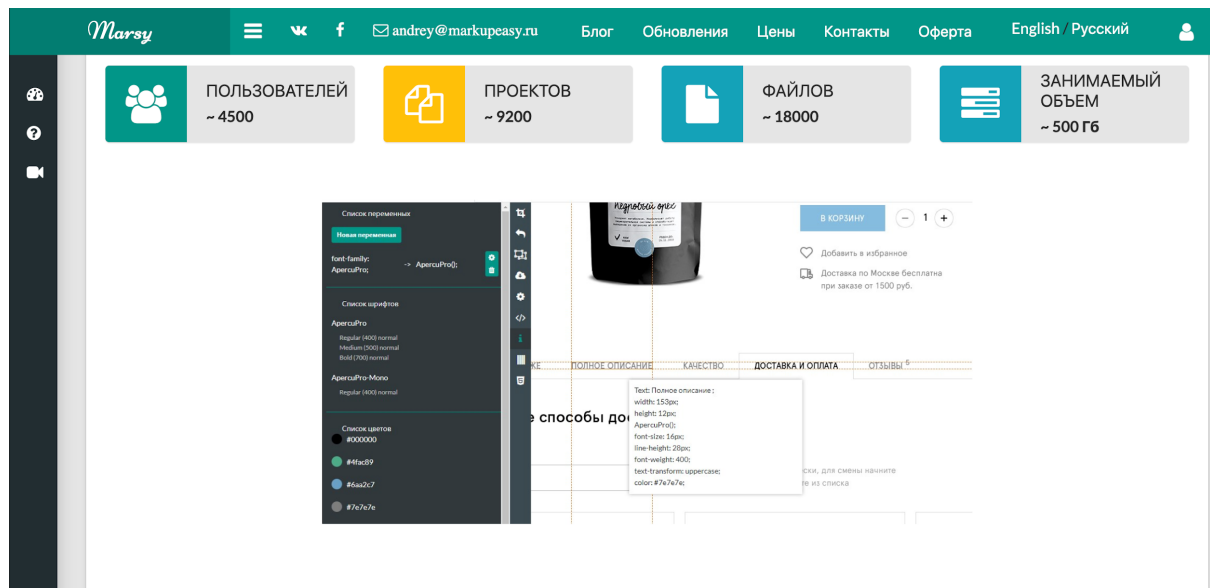


Плюсы

1. Возможность работы практически с любым графическим редактором.
2. Возможность скачивания изображений в SVG-формате.
3. Работа со слоями макета.
4. Настройка отображаемых стилей.
5. Веб-версия.

Минусы

1. 7-дневная бесплатная версия.
2. Отсутствует многопользовательский режим.




Плюсы

1. Бесплатная программа.
2. Работа со слоями макета.
3. Веб-версия.

Минусы

1. Работа только с форматом Photoshop-макетов.
2. Отсутствует многопользовательский режим.



Design Collaboration, Version Control & Handoff

Your entire design workflow in one place

[Try for Free](#)



VISA

♥ badoo



Bethesda

 Rakuten Viber

Trusted by thousands of
companies around the world



Sympli Versions:
Version Control for
Designers

Store and collaborate on Sketch mockups with
branches and unlimited version history.

  Coming soon

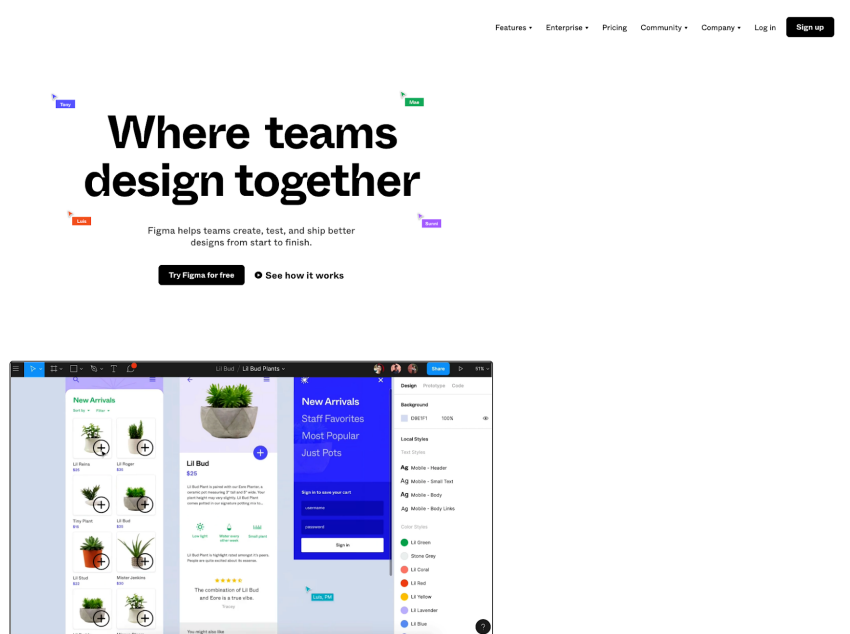
Плюсы

1. Бесплатная программа.
2. Работа со слоями макета.
3. Веб-версия.

Минусы

1. Для работы с программой потребуется установленный Photoshop (не обязательно последней версии).
2. Скачивание изображений без оптимизации.

Figma



Плюсы

1. Бесплатная программа.
2. Работа со слоями макета.
3. Веб-версия.
4. Многопользовательский режим.
5. Подходит как для верстальщика, так и для дизайнера.
6. Скачивание изображений в формате SVG.

Минусы

1. Возможна конвертация из других форматов, но могут быть расхождения внешнего вида.
2. Нет настройки отображаемых CSS-свойств

Figma — самый популярный и удобный инструмент для дизайнеров и верстальщиков. У него огромное количество плюсов и совсем незначительные минусы, поэтому именно на этом варианте работы с макетом мы с вами и остановимся.

Основные преимущества работы с Figma

Кросс-платформенность

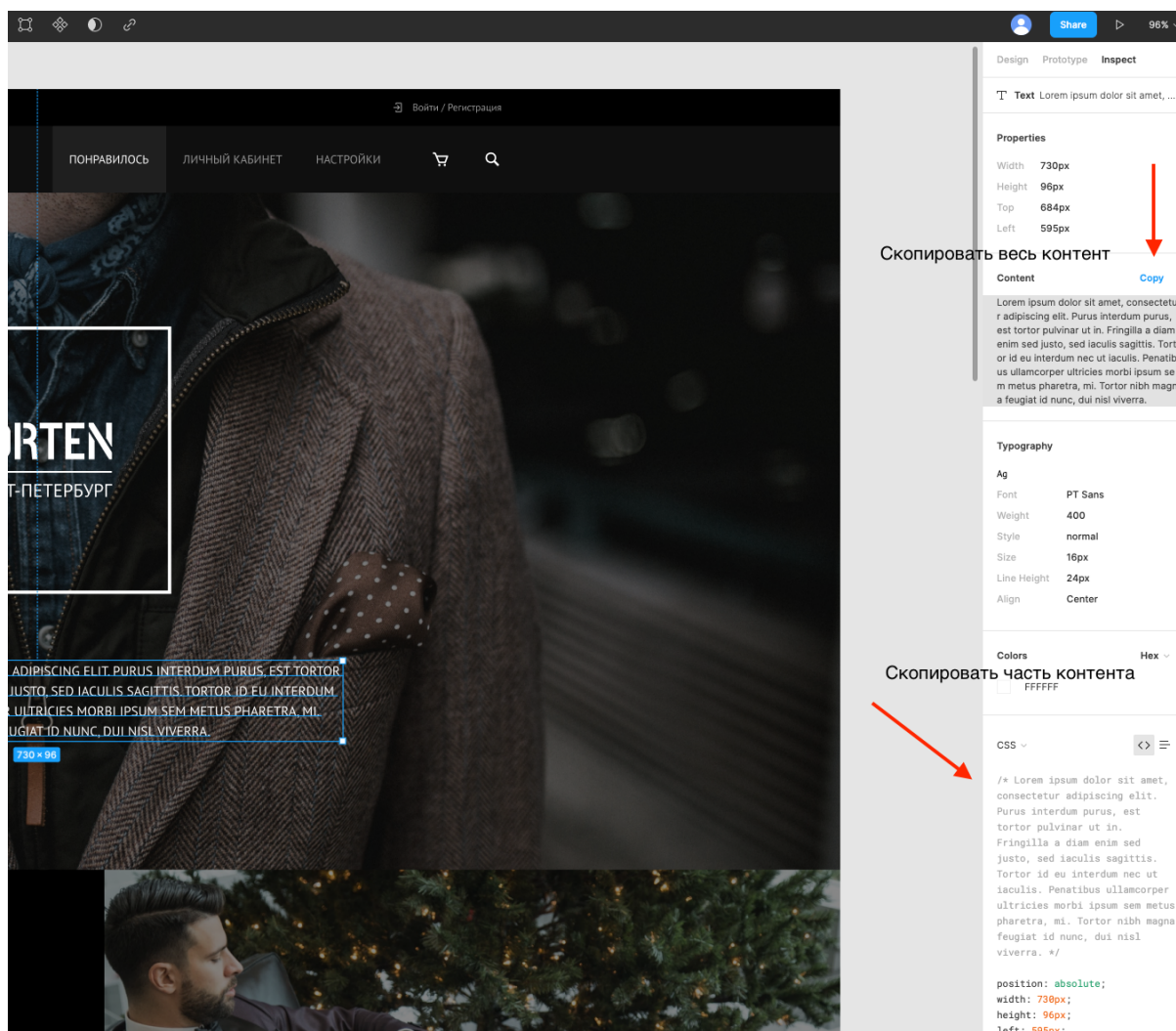
Доступ к макетам есть с любого устройства, даже с iPad. Достаточно запустить браузер и выйти в интернет. А это значит, что больше не нужно постоянно экспортировать макеты с результатом работы из редактора в PNG и JPG. Достаточно отправить ссылку на файл с настроенным доступом, а получателю не обязательно иметь аккаунт в Figma.

Как скопировать контент из макета

Основное отличие макета от изображения — мы можем извлечь из макета всю необходимую информацию. В Figma всё сделано максимально просто и в большинстве ситуаций интуитивно понятно.

Текстовая информация

При выборе блока с текстом у вас есть сразу три варианта копирования контента. Самый простой из них — использовать горячие клавиши Ctrl + C (Для MacOS command + C), и текст будет скопирован в буфер обмена, два альтернативных варианта вы можете увидеть на изображении ниже, они указаны красными стрелками.

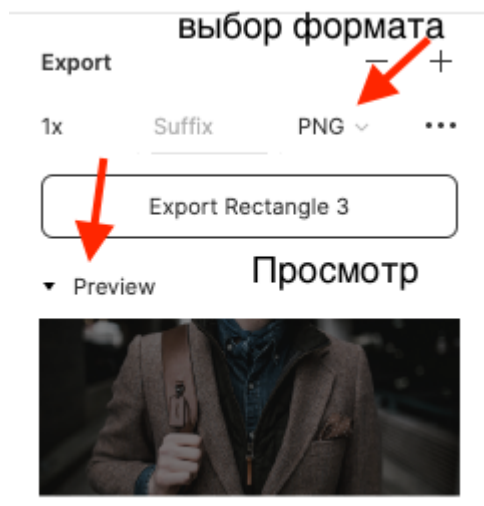


Извлечь изображение из макета

Чтобы скачать изображение из макета, нужно нажать на изображение. Возможно, оно находится под текстом, поэтому потребуется выделить изображение, дважды нажав на него. В панели управления справа вам необходимо найти кнопку Export.



Далее вы можете поменять формат изображения. Тут можно выбрать JPG для прямоугольного изображения, PNG для изображения с прозрачностью, GIF, если представлена GIF-анимация, или SVG, если изображение векторное. Также представлен пример:



Осталось только нажать на кнопку Export, и должна начаться загрузка изображения.

Как узнать расстояние между блоками

При выборе любого элемента в макете вы сможете увидеть, как автоматически рассчитываются расстояния до соседних элементов. Если вам необходимо узнать отступ от верхней границы блока, выделите этот блок и посмотрите расстояние до верхней границы.



Стили элементов

В макетах, созданных в графическом редакторе Figma, в панели управления, расположенной справа, собраны стили любого выбранного вами элемента в макете. При нажатии на любой блок вы получаете всю необходимую стилистику.

Внимание! В блоке стилей представлены параметры, которые чаще всего не нужно копировать, поэтому разберем несколько простых правил, которые значительно упростят работу с добавлением стилей.

Так как figma автоматически выставляет абсолютно все свойства как есть. В связи с этим возникает набор вещей, которые надо корректировать.

Figma укажет абсолютные координаты любого объекта как он указан в макете. Но если мы хотим, чтобы верстка была адаптивной, то такие координаты нас не устроят. Соответственно, не нужно копировать их из автоматически собранных стилей.

Стили, которые не нужно копировать из макета

Если у вас однострочный текст, этому блоку не нужно копировать свойства:

- `position: absolute;`
- `width: 96px;`
- `height: 18px;`
- `left: 998px;`
- `top: 75px;`

Абсолютное позиционирование для всех элементов — это дурной тон. Добавление ширины и высоты однострочному тексту вызовет проблемы при смене контента, параметры `left` и `top` относятся к `position`, поэтому мы его не используем.

Вторая частая ошибка — копирование параметров ширины и высоты каждому блоку:

- `width: 1440px;`
- `height: 96px;`

Если у вас есть блок родительского элемента, у которого уже определена ширина контента 1440px, то добавлять параметры ширины для дочерних элементов не обязательно, так как ширина блока по определению равна 100%.

С высотой блока немного сложнее. Вам необходимо задать себе вопрос, будет ли контент в этом блоке больше или меньше, так как если вы зададите этому блоку высоту, то текстовая информация, если её станет больше, не поместится и будет наезжать на соседние блоки. Те самые блочные элементы, которые мы рассмотрим чуть позже.

Основные теги вёрстки

В верстке сайта с помощью слоёв самый часто используемый HTML-тег — `<div>`. Он и формирует слой на веб-странице. Это блочный тег.

Второй тег, который используется в вёрстке, — строчный тег ``.

```
<div>Блочный элемент</div>
<span>Строчный элемент</span>
```

Элементы `span` используются для стилизации какой-то произвольной части текста. Например, текст зелёного цвета. Чтобы эту часть реализовать в вёрстке, требуется обернуть этот текст тегом `span`.

html	css
<pre><p class="text"> В этом параграфе текст зеленого цвета будет размером 18px и красного цвета </p></pre>	<pre>.green { color: green; }</pre>

Блочный элемент создает разрыв строки перед тегом и после него. Он образует прямоугольную область. Она занимает всю ширину веб-страницы (100%) или блока-родителя, если для него нет значения `width`.

Блочные элементы содержат элементы любого типа. Нельзя размещать блочные элементы внутри строчных, кроме элемента ``. Для блочных элементов задаются отступы. Свойства `width` и `height` устанавливают ширину и высоту области содержимого элемента. Фактическая ширина элемента складывается из ширины полей (внутренних отступов), границ и внешних отступов.

Простыми словами, всё на странице состоит из прямоугольников. Это и будут элементы `div`.

Строчные элементы не создают блоки, а отображаются на одной строке с содержимым рядом стоящих тегов. Это потомки блочных элементов. Они игнорируют верхние и нижние отступы.

Ширина и высота строчного элемента зависит только от его содержания. Задать его размеры с помощью CSS нельзя. Можно увеличить расстояние между соседними элементами по горизонтали с помощью горизонтальных полей и отступов.

Рассмотрим основные блочные и строчные элементы HTML.

Блочные элементы

- `<h1>, <h2>...<h6>`
- `...`
- `...`
- `<body>...</body>`
- `<div>...</div>`
- `<p>...</p>`
- `...`
- ...

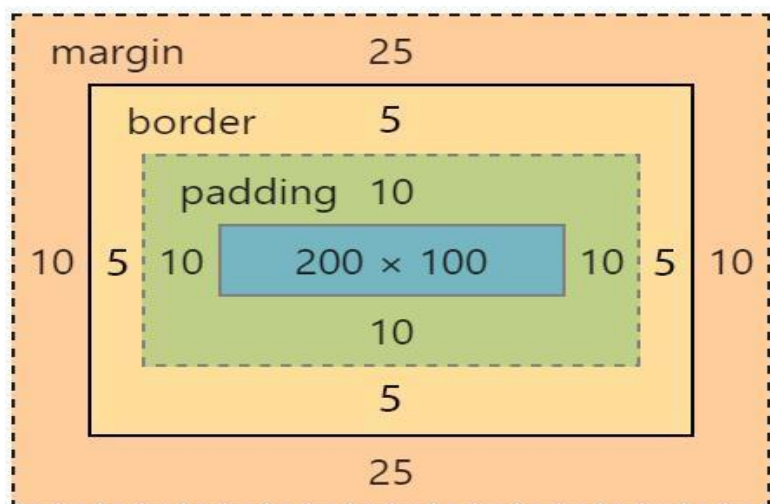
Строчные элементы

- `...`
- `<i>...</i>`
- `...`
- `...`
- `...`
- ``
- `<input>`
- ...

В этом списке тег картинки `` — замещаемый строчный элемент. При помощи замещаемых элементов указывается, что в этом месте должен быть какой-то сторонний объект, в нашем случае, картинка. Замещающему элементу задаётся ширина и высота. Но это всё равно строчный элемент.

Формирование блочной модели

Атрибуты `width` и `height` — неокончательные размеры элемента. Чтобы вычислить размеры, требуется учитывать следующие моменты.



Согласно схеме, напрашивается вывод, что ширина блока складывается из таких свойств:

```
margin-left +  
border-left +  
padding-left +  
width +  
padding-right +  
border-right +  
margin-right
```

Соответственно, высота из следующих:

```
margin-top +  
border-top +  
padding-top +  
height +  
padding-bottom +  
border-bottom +  
margin-bottom
```

Внутренний отступ или поле элемента (padding) добавляет отступы внутри элемента, между его основным содержимым и границей. Если элементу задать фон, он распространится и на поля элемента. Такой отступ не принимает отрицательных значений, в отличие от внешнего отступа.

Внешний отступ (margin) добавляет отступы за границами элемента, создавая тем самым промежутки между элементами. Они всегда остаются прозрачными. Через них виден фон родительского элемента. Значения padding и margin задаются в следующем порядке: верхнее, правое, нижнее и левое.

Граница или рамка элемента задается с помощью свойства `border`. Если цвет рамки не указывается, она принимает цвет основного содержимого элемента, например, текста. Если рамка имеет разрывы, то сквозь них проступает фон элемента.

Внешние, внутренние отступы и рамка элемента — необязательные свойства. По умолчанию их значение равно нулю. Но некоторые браузеры добавляют к ним положительные значения автоматически на основе собственных таблиц стилей. Как добавить обнуление отступов, чтобы во всех браузерах корректно отображался сайт, мы узнаем в следующих уроках.