

# CSS - FIGMA

FIGMA, pixel-perfect, mask(доп)

# Mask

CSS свойство `mask` используется для применения маски к элементам веб-страницы. Маска определяет, какая часть элемента будет видима, а какая скрыта, основываясь на содержимом маски. Маска может быть создана изображением, градиентом или SVG-графикой.

Иногда требуются вендорные префиксы, пример:

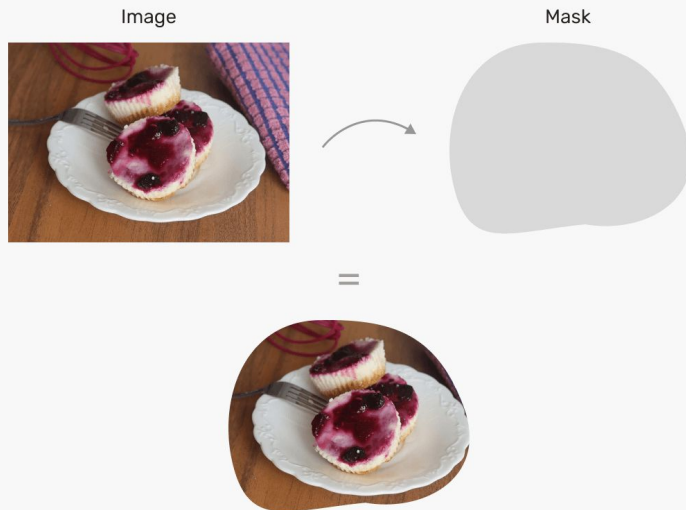
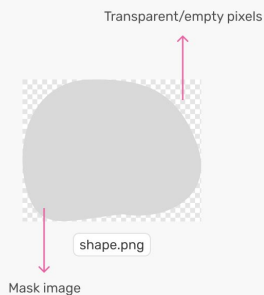
`-webkit-mask`

`-moz-mask`

# mask-image

Устанавливает изображение в качестве маски. Можно использовать как растровые, так и векторные изображения.

```
.masked-element {  
  mask-image: url('mask-image.png');  
}
```



# mask-mode - ТЫК

Свойство mask-mode определяет режим смешивания маски с содержимым элемента

Возможные значения: alpha, luminance, alpha luminance.

- alpha: В этом режиме маска использует только альфа-канал (прозрачность) своего изображения. При этом непрозрачные области маски скрывают содержимое элемента, а прозрачные области позволяют содержимому элемента проникать через маску. Это полезно, например, когда нужно применить маску изображения с полупрозрачными областями.
- luminance: В этом режиме маска использует яркость (luminance) своего изображения. При этом белые области маски полностью скрывают содержимое элемента, а черные области полностью отображают содержимое. Серые области маски приводят к частичному отображению содержимого элемента. Это полезно для создания эффектов маскирования на основе яркости изображения.
- alpha luminance: Этот режим объединяет оба предыдущих. Маска использует как альфа-канал, так и яркость своего изображения. Это дает более сложные возможности для маскирования, позволяя использовать как прозрачность, так и яркость для определения видимых областей содержимого элемента.

# mask-repeat - ТЫК

Свойство `mask-repeat` в CSS определяет, как должна повторяться маска внутри элемента, к которому она применена. Это свойство позволяет контролировать поведение маски, если она меньше или больше, чем элемент, к которому она применяется.

- `mask-repeat: repeat`: Это значение указывает, что маска должна повторяться как по горизонтали, так и по вертикали, чтобы полностью заполнить область элемента. Если маска меньше, чем элемент, она будет повторяться по всей области элемента.
- `mask-repeat: repeat-x`: Это значение указывает, что маска должна повторяться только по горизонтали, чтобы заполнить ширину элемента. Если маска меньше, чем элемент, она будет повторяться только по горизонтали.
- `mask-repeat: repeat-y`: Это значение указывает, что маска должна повторяться только по вертикали, чтобы заполнить высоту элемента. Если маска меньше, чем элемент, она будет повторяться только по вертикали.
- `mask-repeat: no-repeat`: Это значение указывает, что маска не должна повторяться. Если маска меньше, чем элемент, она будет отображена один раз, а остальная область элемента, не занятая маской, будет оставлена пустой или будет отображено содержимое элемента.

# mask-position - ТЫК

Свойство `mask-position` в CSS определяет позицию маски внутри элемента, к которому она применяется. Оно позволяет контролировать точное расположение маски относительно элемента.

Значения свойства `mask-position` могут быть заданы как ключевые слова, проценты или длины (в пикселях). Вот основные значения:

- Ключевые слова: `center`: Маска располагается по центру элемента. `top`, `right`, `bottom`, `left`: Маска располагается соответственно вверху, справа, внизу или слева относительно элемента.
- Проценты: Маска располагается относительно размеров элемента. Например, `50% 50%` означает, что маска будет находиться посередине элемента как по горизонтали, так и по вертикали.
- Длины: Маска располагается на заданных координатах относительно левого верхнего угла элемента. Например, `10px 20px` означает, что маска будет находиться в 10 пикселях от верхней границы элемента и в 20 пикселях от левой границы элемента.
- Можно также комбинировать значения для задания горизонтальной и вертикальной позиции маски. Например, `top center` располагает маску в верхней части и по центру элемента.

## mask-clip - ТЫК

Свойство `mask-clip` в CSS определяет область, в которой маска должна быть видима. Это позволяет управлять тем, как маска обрезается или ограничивается относительно границ элемента.

- `mask-clip: border-box`: Маска видна только внутри границ элемента. Это значит, что маска обрезается по границам элемента, включая границу, но не выходит за их пределы.
- `mask-clip: padding-box`: Маска видна только внутри области содержимого элемента, ограниченной его границами и отступами. Это значит, что маска обрезается по границам элемента, но не включает в себя границы.
- `mask-clip: content-box`: Маска видна только внутри области содержимого элемента, ограниченной его границами. Это значит, что маска обрезается по границам элемента и не включает в себя границы или отступы.

## mask-origin - ТЫК

Свойство `mask-origin` в CSS определяет точку отсчета для позиционирования маски относительно элемента, к которому она применяется. Оно указывает, какая часть элемента будет использоваться в качестве источника координат для позиционирования маски.

- `mask-origin: border-box`: Маска позиционируется относительно границ элемента. Это означает, что координаты маски измеряются от границ элемента.
- `mask-origin: padding-box`: Маска позиционируется относительно области содержимого элемента, ограниченной его границами и отступами. Это означает, что координаты маски измеряются от границ элемента, но не включают в себя границы.
- `mask-origin: content-box`: Маска позиционируется относительно области содержимого элемента, ограниченной его границами. Это означает, что координаты маски измеряются от границ элемента и не включают в себя границы или отступы.



## mask-size - [ТЫК](#)

Свойство mask-size в CSS определяет размер маски, применяемой к элементу. Оно позволяет установить размеры маски относительно элемента, к которому она применяется.

contain и cover: Они задают размеры маски так, чтобы она полностью вместила в элемент (contain) или полностью покрывала его (cover).

auto, которые автоматически определяют размеры маски.

Можно использовать проценты для определения размеров маски относительно размеров элемента.

Можно указать конкретные значения ширины и высоты маски, например, в пикселях.

# FIGMA

Figma - это мощный инструмент для дизайна интерфейсов, который предоставляет возможности для создания, прототипирования и совместной работы над дизайн-проектами в реальном времени. Он позволяет дизайнерам создавать макеты, интерфейсы и прототипы веб-сайтов и мобильных приложений.

# Плюсы Figma для front-end разработчика

- Совместная работа и командная синхронизация: Figma позволяет нескольким членам команды работать над проектом одновременно, делиться комментариями и видеть изменения в реальном времени. Это сильно упрощает совместную работу и улучшает коммуникацию в команде.
- Прототипирование: С помощью Figma можно создавать прототипы интерфейсов, что позволяет фронтенд-разработчикам быстро визуализировать и тестировать пользовательский опыт перед началом разработки.
- Универсальность и доступность: Figma доступен через веб-браузер и имеет приложения для различных операционных систем, что позволяет работать над проектами из любого места. Также он обеспечивает возможность экспорта дизайна в различные форматы, включая SVG, PNG и другие, что облегчает интеграцию с процессом разработки.
- Библиотеки и компоненты: Figma позволяет создавать библиотеки компонентов, которые могут использоваться повторно в различных проектах. Это сокращает время разработки и обеспечивает консистентность в дизайне.
- Интеграция с другими инструментами: Figma интегрируется с различными инструментами для разработки, такими как Zeplin, Avocode, Abstract и другими, что упрощает процесс передачи дизайна от дизайнера к разработчику.

# Минусы Figma для front-end разработчика

- Сложность изучения: Новичкам может потребоваться время, чтобы освоить все возможности Figma и научиться эффективно им пользоваться.
- Ограниченные возможности анимации: Хотя Figma предоставляет инструменты для создания прототипов, его возможности анимации могут быть ограниченными по сравнению с специализированными инструментами.
- Зависимость от интернет-соединения: Для работы в Figma требуется постоянное интернет-соединение, что может быть неудобно в случае непостоянной доступности к сети.

# Примеры figma

Landing страница портфолио front-end разработчика - [ТЫК](#)

Пример сайта для фотографа - [ТЫК](#)

Пример онлайн магазина - [ТЫК](#)

# Pixel Perfect

Pixel perfect (или пиксель-перфект) на фронтенде относится к методологии разработки, при которой веб-сайт или веб-приложение разрабатывается с такой точностью, чтобы его внешний вид и визуальное представление полностью соответствовали исходному дизайну в графическом редакторе или макете.

Ключевой целью пиксель-перфектной разработки является создание веб-страницы, которая выглядит идентично исходному макету в браузере, с учетом всех деталей и элементов дизайна, таких как размеры, расположение, цвета, шрифты и отступы.

# цели Pixel Perfect

- Точное соответствие дизайну: Каждый элемент веб-страницы должен быть размещен и стилизован таким образом, чтобы полностью соответствовать исходному дизайну.
- Использование точных размеров: Размеры элементов, шрифтов, отступов и других деталей должны быть точно установлены в соответствии с макетом.
- Поддержка разных браузеров и устройств: При пиксель-перфектной разработке необходимо учитывать совместимость с различными браузерами и устройствами, чтобы обеспечить одинаковый внешний вид и функциональность на всех платформах.
- Тестирование и проверка: Важно тщательно тестировать веб-сайт или приложение, чтобы убедиться, что он отображается корректно и выглядит идентично в разных условиях и на разных устройствах.

Плагин - [ТЫК](#)

Плагин для Google Chrome - [ТЫК](#)

# Ресурсы

- статья про mask - [ТЫК](#)
- w3 документация mask - [ТЫК](#)
- документация MDN по mask - [ТЫК](#)
- html book документация mask - [ТЫК](#)
- статья про pixel perfect - [ТЫК](#)