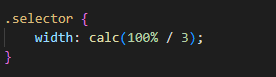
**Практическая работа №7 «Функция calc() в CSS. Свойство filter, функции фильтров. Свойство backdrop-filter, эффект glassmorphism. Свойство cursor»**

## 13 Функция calc()

**Функция calc()** - функция, позволяющая производить математические вычисления прямо в CSS.

Как самый простой пример использования функции calc() можно привести частую ситуацию в верстке, когда необходимо разделить контент на три колонки, ширина каждой из которых равна трети от 100%. Изначально указывалось значение 100% / 3 = 33,33333…%; при округлении, соответственно, получается 33%, но теперь можно использовать calc() и браузер сам рассчитает ширину колонок (листинг 13.1).

Листинг 13.1 – Добавление функции calc()

****

**Примечание:** при использовании этой функции следует учитывать, что на каждую операцию браузер тратит некоторую долю секунды и часть оперативной памяти. Если расчётов будет много, то это потенциально повлияет на скорость загрузки страницы. Используйте calc() с умом.

## Синтаксис функции calc()

В круглых скобках можно писать любые математические операции с любыми единицами измерения, доступными в вебе (*%, px, rem, em, vw, vh, vmin* и т.д.). Доступны четыре стандартных операнда:

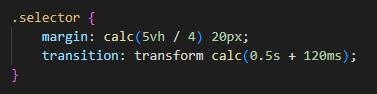
* **+** - сложение;
* **-** - вычитание;
* **/** - деление;
* **\***- умножение.

**Операторы сложения и вычитания** обязательно с двух сторон **должны отбиваться пробелом.** Иначе браузер воспримет их как часть числа. Хоть **операторы деления и умножения** не требуют такой строгости к себе, принято и **их тоже отбивать** **пробелами** для удобства чтения.

**Внутри скобок может быть больше одного вычисления,** можно группировать операции при помощи скобок. Но не стоит увлекаться: чем короче вычисление, тем проще потом его прочитать и понять, что в нем прописано.

**Функцию calc() можно использовать для любых свойств, значением которых должна быть цифра.** Причём если свойство предполагает составное значение, то можно указать функцию как часть этого значения (листинг 13.2):

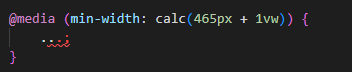
Листинг 13.2 – Функция calc()для CSS-свойства



**Внутри круглых скобок можно складывать только числовые значения.** Нельзя сложить число со строкой.

**Также неудачным местом применения этой функции являются медиавыражения.** Вот такая запись считается не валидной (листинг 13.3):

Листинг 13.3 – Не валидная функция calc()



## 14 Свойство filter

Свойство filter даёт возможность прямо в CSS применять к элементам различные графические эффекты. Например, перекрасить элемент в сепию, добавить ему необычную тень или использовать SVG в качестве фильтра.

Фильтр — очень мощный инструмент, он позволяет реализовать многие дизайнерские идеи при помощи одной строчки кода. Фильтры можно применить не только к картинкам, но и к любым непустым элементам.

В использовании фильтров удобно то, что исходная картинка или элемент никак не изменяются сами по себе. Меняется только их визуальное отображение в браузере. А это значит, что вы сможете использовать их повторно в других ситуациях с другими фильтрами или совсем без них.

Можно не только задавать статичные визуальные эффекты, но и анимировать их.

## Функции фильтров

Список функций, с помощью которых можно задать фильтры картинкам.

Функции используются в качестве значений для свойств filter и backdrop-filter.

## Функция blur()

**Примеряет размытие Гаусса к изображению.** Значение в скобках указывает сколько пикселей сливаются друг с другом. Чем больше значение, тем больше размытие. Можно указать положительное значение в любых единицах измерения, кроме процентов (листинг 14.1 и рисунок 14.1).

Листинг 14.1 – Функция blur()

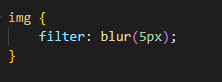


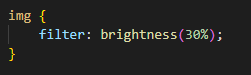


Рисунок 14.1 – Применение функции blur() к изображению

## Функция brightness()

**Меняет яркость изображения.** В скобках можно указать любое значение от 0% и выше. Значение 0% сделает изображение полностью чёрным. Значение 100% вернёт изображению исходную яркость. Значение больше 100% усилит яркость картинки. Значением может быть целое или дробное число без единиц измерения (листинг 14.2 и рисунок 14.2).

Листинг 14.2 – Функция brightness()



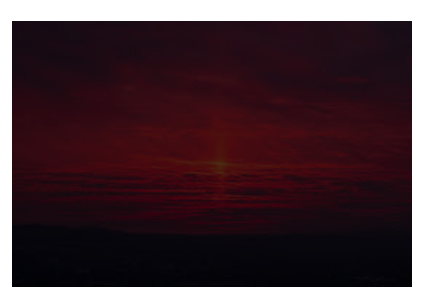


Рисунок 14.2 – Применение функции brightness() к изображению

## Функция contrast()

**Меняет контраст изображения.** В скобках можно указать любое значение от 0% и выше. Значение 0% сделает изображение полностью чёрным. Значение 100% вернёт изображению исходный контраст. Значение больше 100% усилит исходный контраст. Значением может быть целое или дробное число без единиц измерения (листинг 14.3 и рисунок 14.3).

Листинг 14.3 – Функция contrast()

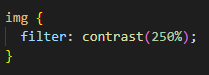




Рисунок 14.3 – Применение функции contrast() к изображению

## Функция drop-shadow()

**Задаёт тень для картинки.** Тень располагается снаружи элемента (листинг 14.4 и рисунок 14.4).

Эта функция очень похожа на box-shadow по допустимым значениям и результату. Разница лишь в том, что нельзя указывать ключевое слово inset.

Листинг 14.4 – Функция drop-shadow()

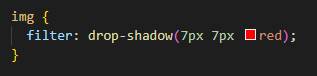




Рисунок 14.4 – Применение функции drop-shadow() к прямоугольному изображению

С помощью drop-shadow() можно создать тень по форме самого изображения (рисунок 14.5). Функция учитывает альфа-канал картинки и способна отбрасывать не только прямоугольную тень, как в случае с box-shadow.

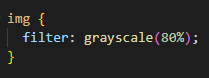


Рисунок 14.5 – Применение функции drop-shadow() к изображению .png

## Функция grayscale()

**Делает изображение чёрно-белым.** В скобках можно указать значение от 0% до 100%. Значение 100% сделает изображение полностью чёрно-белым. Значение 0% вернёт изображению исходные цвета. Значением может быть целое или дробное число без единиц измерения (листинг 14.5 и рисунок 14.6).

Листинг 14.5 – Функция grayscale()



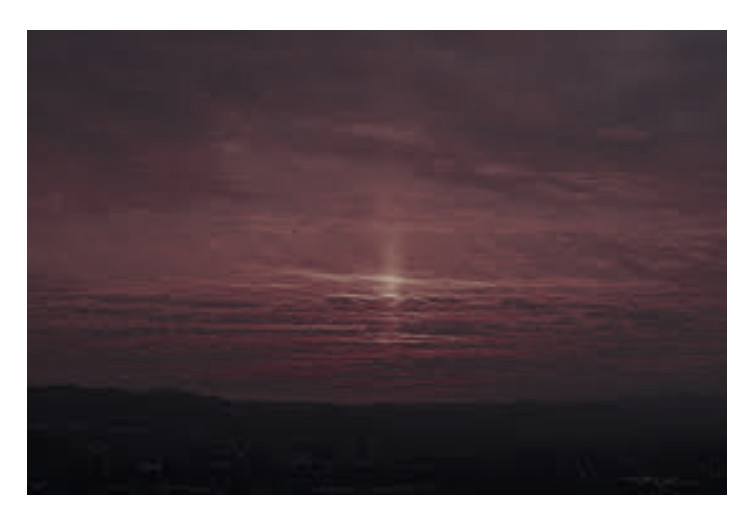


Рисунок 14.6 – Применение функции grayscale() к изображению

## Функция hue-rotate()

**Меняет цвета изображения за счёт поворота цветового круга.** Угол поворота указывается в скобках функции. Можно указывать угол в градусах deg или в поворотах turn (листинг 14.6 и рисунок 14.7).

Листинг 14.6 – Функция hue-rotate()

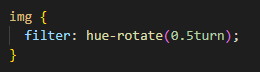




Рисунок 14.7 – Применение функции hue-rotate() к изображению

## Функция invert()

**Инвертирует цвета изображения**, как бы выворачивает их, превращая в противоположные. В результате получается что-то вроде негатива. Можно указать процент инверсии от 0% до 100%. При 100% цвета на картинке полностью инвертированы. Отрицательные значения или значения больше 100% не допускаются (листинг 14.7 и рисунок 14.8).

Листинг 14.7 – Функция invert()

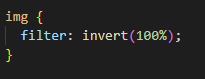




Рисунок 14.8 – Применение функции invert() к изображению

## Функция opacity()

**Меняет прозрачность изображения.** Можно указать процент прозрачности от 0% до 100%. 0% делает картинку полностью прозрачной. 100% не меняет прозрачность изображения. Отрицательные значения или значения больше 100% не допускаются (листинг 14.8 и рисунок 14.9).

Очень похоже на работу свойства opacity с той разницей, что для фильтра браузер, как правило, применяет аппаратное ускорение для улучшения производительности.

Листинг 14.8 – Функция opacity()

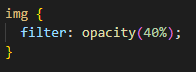




Рисунок 14.9 – Применение функции opacity()к изображению

## Функция saturate()

**Меняет насыщенность цветов изображения.** Значение 0% полностью убирает насыщенность цветов. Значение 100% не изменяет исходное изображение. Допускаются значения больше 100% что приводит к перенасыщенности. Нельзя указать отрицательное значение (листинг 14.9 и рисунок 14.10).

Листинг 14.9 – Функция saturate()

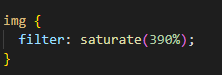


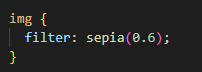


Рисунок 14.10 – Применение функции saturate()к изображению

## Функция sepia()

**Меняет цвета изображения на сепию** — коричневые оттенки. Значение 100% полностью преобразует изображение в сепию. Значение 0% не изменяет исходное изображение. Отрицательные значения или значения больше 100% не допускаются. Можно использовать целое или дробное число без единиц измерения в качестве значения (листинг 14.10 и рисунок 14.11).

Листинг 14.10 – Функция sepia()



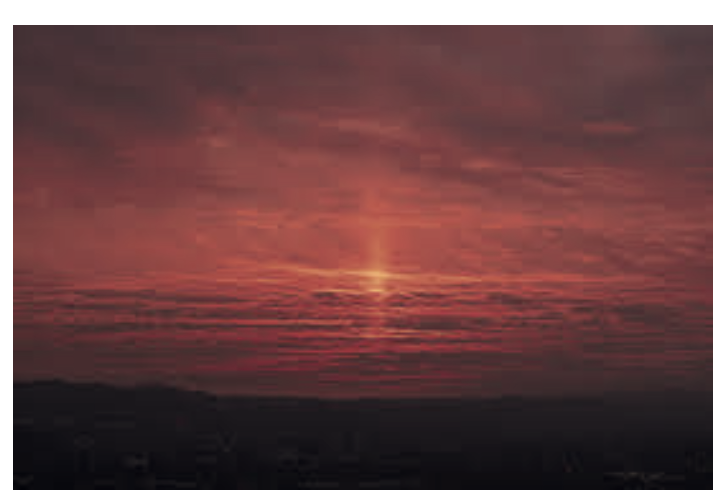


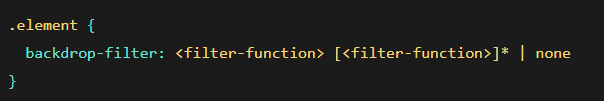
Рисунок 14.11 – Применение функции sepia()к изображению

## 15 Свойство backdrop-filter

Свойство backdrop-filter позволяет применить фильтры типа размытия, изменения контраста или обесцвечивания к фоновому изображению, находящемуся за элементом. Поскольку оно действует по отношению ко всему, что находится позади элемента, его необходимо сделать, по крайней мере, частично прозрачным.

**Синтаксис** (листинг 15.1):

Листинг 15.1 – Синтаксис свойства backdrop-filter



Можно использовать любую (или сразу несколько функций через пробел) из нижепредставленных:

* blur()
* brightness()
* contrast()
* drop-shadow()
* grayscale()
* hue-rotate()
* invert()
* opacity()
* saturate()
* sepia()
* url() – для применения SVG-фильтров

## Эффект glassmorphism

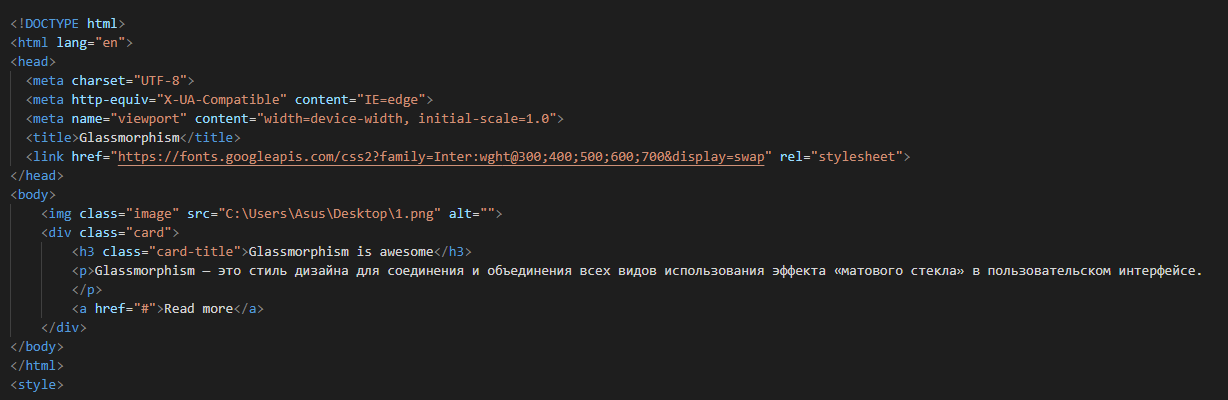
**Glassmorphism** — новая тенденция, которая активно набирает популярность на таких сервисах, как Dribbble и Behance.

Его характерными чертами являются:

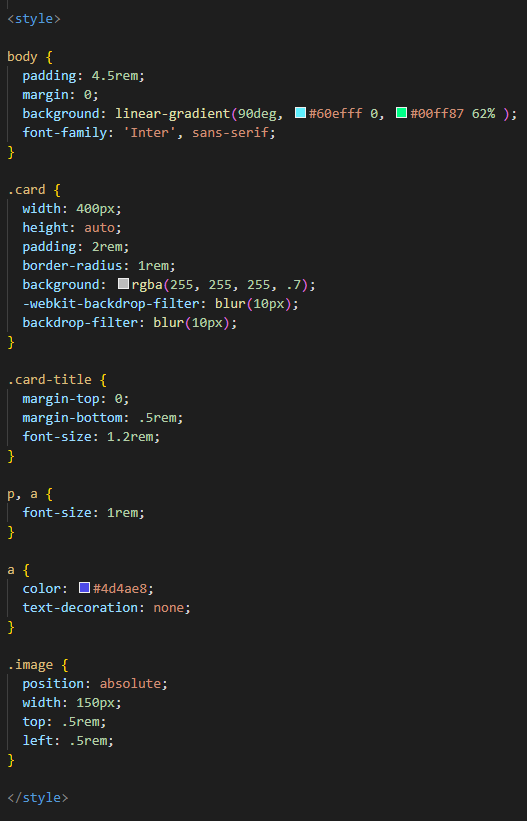
* прозрачность (эффект матового стекла);
* яркие или пастельные цвета;
* светлые границы элементов.

Для создания эффекта стекломорфизма следует использовать backdrop-filter: blur()(листинги 15.2, 15.3 и рисунок 15.1).

Листинг 15.2 – HTML-код для создания эффекта glassmorphism



Листинг 15.3 – CSS-код для создания эффекта glassmorphism



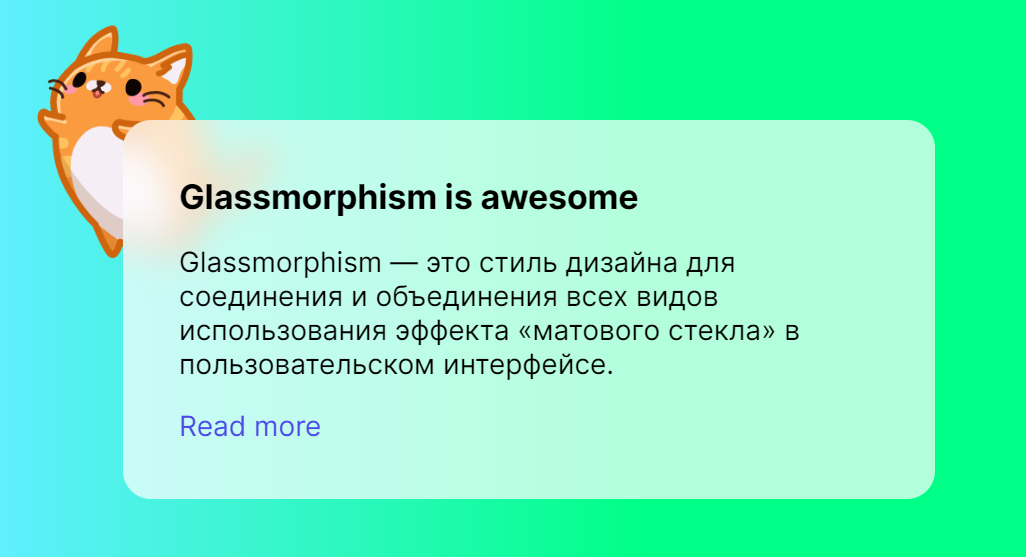


Рисунок 15.1 – Эффект glassmorphism

## 16 Свойство cursor

С помощью этого свойства можно указать, какой внешний вид будет у курсора, когда пользователь наведёт мышку на элемент.

**Основные значения (более популярные) свойства** cursor:

* auto – браузер сам решит какой курсор показывать в зависимости от того, на какой элемент он наведён. Например, для ссылки это будет рука с пальчиком, для поля ввода чёрточка, а для обычного блока — стандартная стрелка.
* default – обычная, привычная стрелочка.
* none – курсора совсем нет.
* pointer – курсор, который обычно появляется над нажимаемыми элементами типа ссылок.
* text – курсор в виде чёрточки с засечками сверху и снизу. Обычно показывается там, где текст может быть выбран, выделен.

**Остальные, менее популярные значения свойства** cursor:

* context-menu – курсор контекстного меню (обычно вызывается правой кнопкой мыши).
* help – доступен вспомогательный информационный контент.
* progress – программа в фоне выполняет какие-то действия, но пользователь всё ещё может с ней взаимодействовать.
* wait – программа не отвечает, занята обработкой какой-то операции.
* cell – можно выбрать одну или несколько ячеек таблицы.
* crosshair – курсор-крестик, обычно используется, чтобы показать, что на изображении можно выбрать какую-то область.
* vertical-text – практически как text, но вертикально.
* alias – загнутая стрелочка, так обозначают ссылки, клик по которым уведёт с текущего сайта.
* copy – содержимое можно скопировать.
* move – содержимое можно подвигать.
* no-drop – в эту область нельзя перетащить файл.
* not-allowed – действие не будет выполнено.
* grab – содержимое можно схватить, чтобы перетащить.
* grabbing – содержимое было схвачено для перетаскивания.
* all-scroll – содержимое может быть проскроллено в любом направлении.
* col-resize – колонку / ячейку таблицы можно изменить в размерах по горизонтали.
* row-resize – строку в таблице или другой элемент можно изменить в размерах по вертикали.
* zoom-in – содержимое можно приблизить, увеличить.
* zoom-out – содержимое можно отдалить, уменьшить.

Кроме значений, заданных при помощи ключевых слов, можно указывать ссылку на картинку, которая будет показана вместо курсора (листинг 16.1).

**Если вы решили сделать кастомный курсор, то обязательно укажите через запятую одно из стандартных значений.** Это нужно на случай, если браузер не смог загрузить или отобразить картинку. Тогда будет показан тот курсор, который вы указали в конце значения.

Можно указывать несколько картинок подряд через запятую (url(cat.svg), url(cat.png),…), тогда будет показана первая из доступных.

Листинг 16.1 – Установка курсора в виде картинки



Наглядное представление всех значений свойства cursor можно посмотреть по ссылке [Cursor Property Cheatsheet]: <https://codepen.io/GuillaumeChabot/pen/zrJbvM>

Также ниже представлен рисунок с некоторыми значениями свойства cursor (рисунок 16.1):

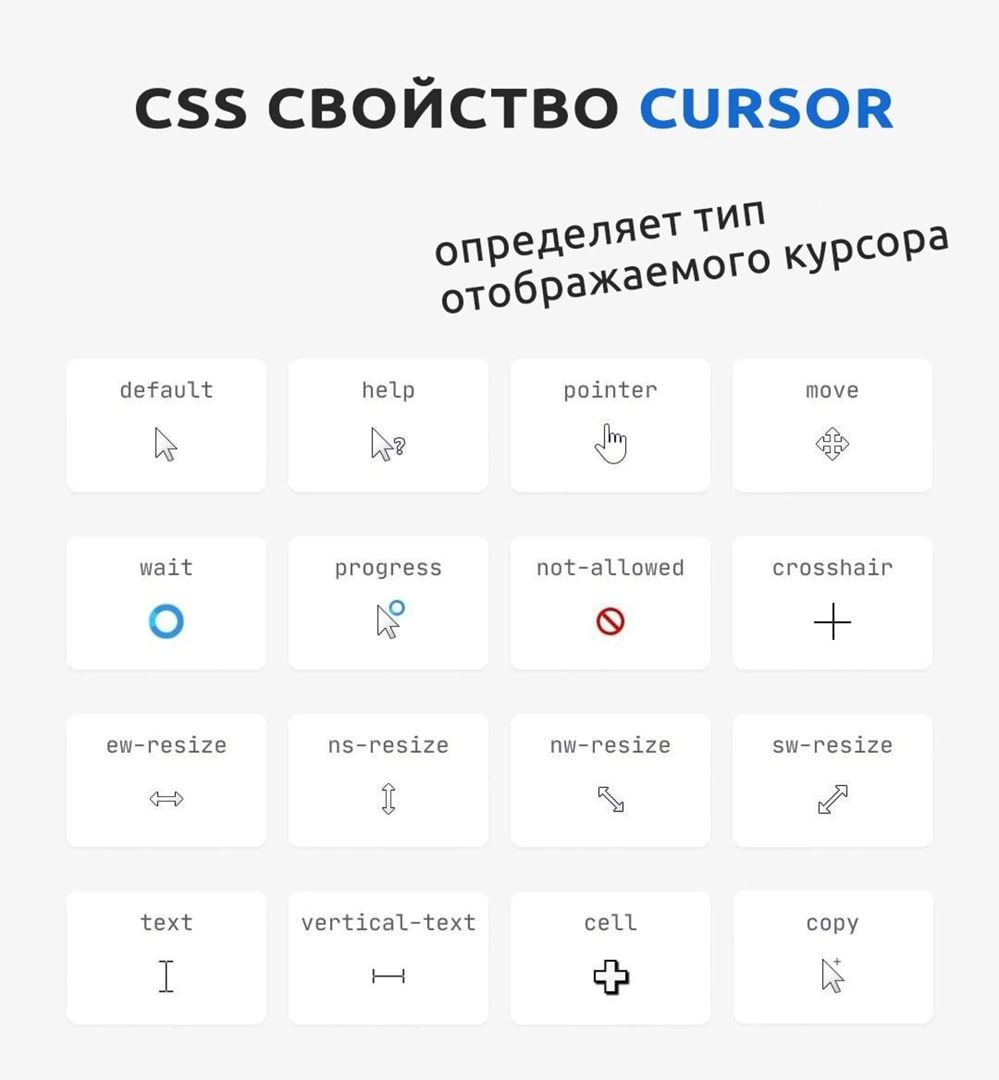


Рисунок 16 – Некоторые значения свойства cursor

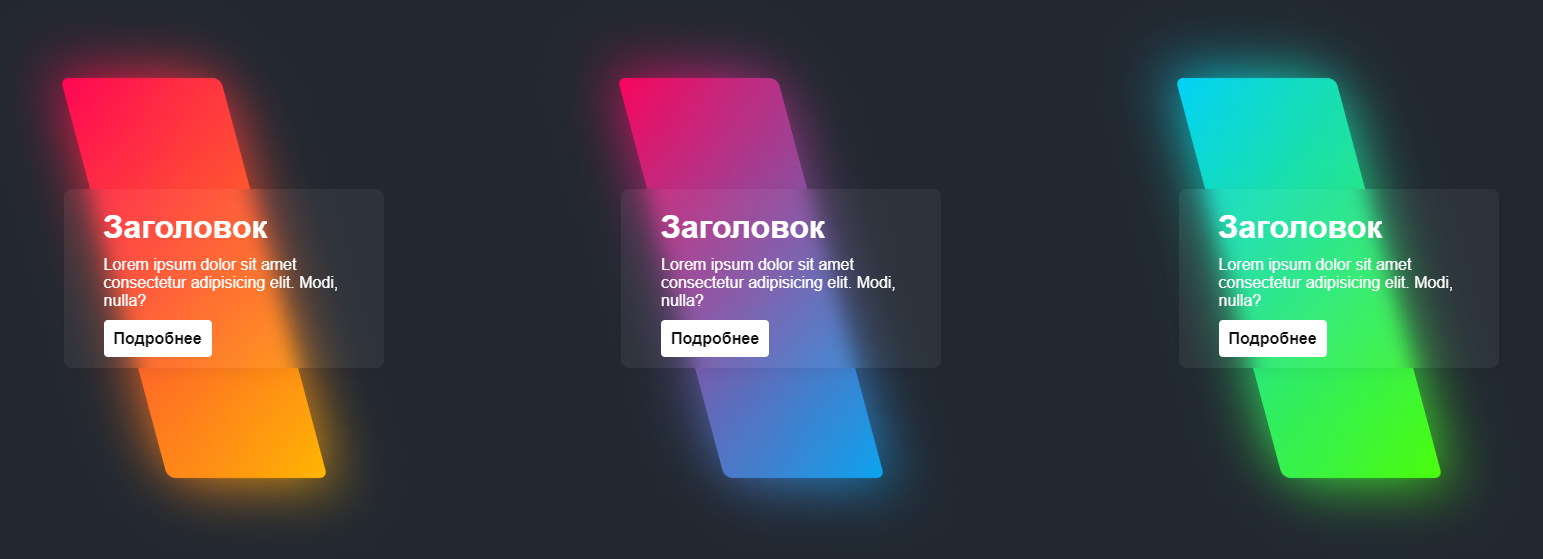
## Практическое задание

**Примечание.** В **каждом** элементе должны быть использованы CSS-переменные и адаптивность (медиа-запросы, свойство flex-wrap или относительные единицы измерения: *%, rem, em, vw, vh* и т.д.). Контент страницы всегда должен пропорционально умещаться на экране устройств с шириной от 320px до 2560px.

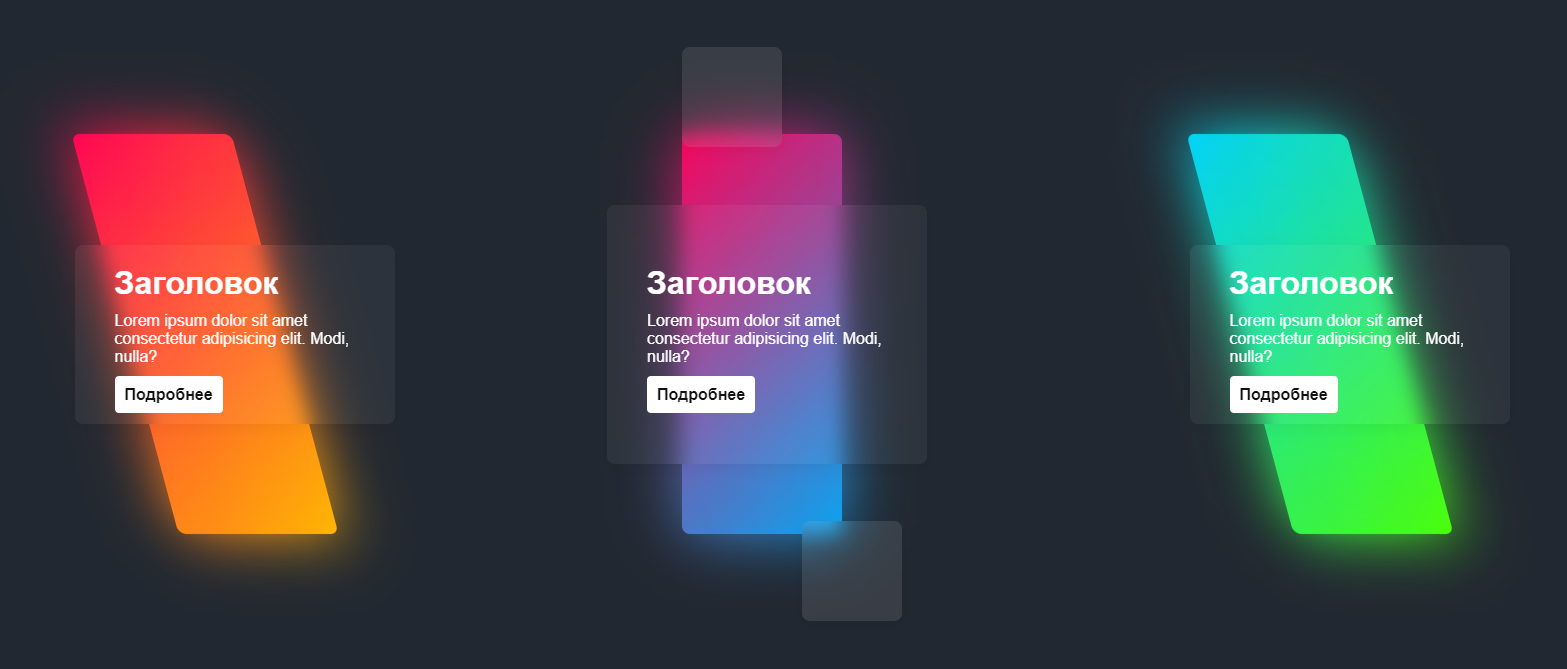
**1.** Создать три карточки по тематике своего сайта в стиле glassmorphism с градиентным элементом сзади. Добавить плавную покадровую анимацию (@keyframes) при наведении, как на фото ниже. Меньшие по размеру квадраты должны перемещаться.

**Обязательно**: использовать псевдоэлементы ::before и ::after.

До наведения:



После наведения (**включается покадровая анимация** @keyframes):





**2.** Создать 5 кнопок социальных сетей (например, в футере) с плавной анимацией: при наведении сверху должен появляться градиент и закрывать весь предыдущий задний фон. **Обязательно**: использовать псевдоэлемент(-ы).

Пример:

До наведения:



После наведения:



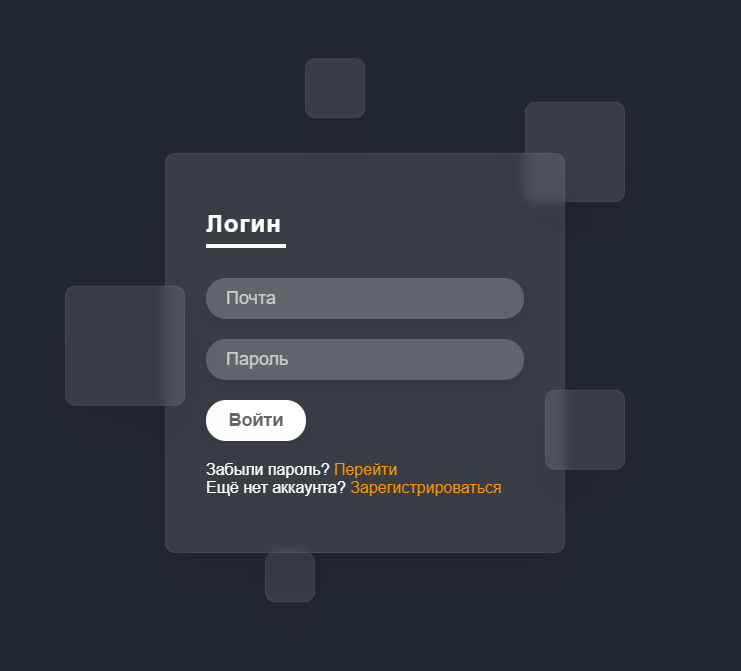


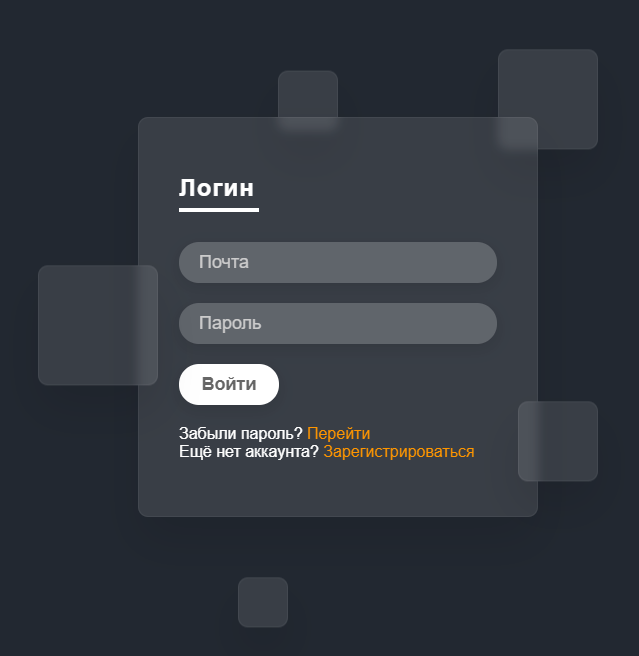


**3.** Создать форму регистрации (или доработать те формы, которые вы создавали на одной из предыдущих практических работ) в стиле glassmorphism с плавной покадровой анимацией. Меньшие по размеру квадраты должны быть анимированы и перемещаться в случайном порядке (используйте переменные и функцию calc()).

**Обязательно**: использовать псевдоэлемент(-ы), функции свойства transform, функцию calc(), переменные.

Пример:

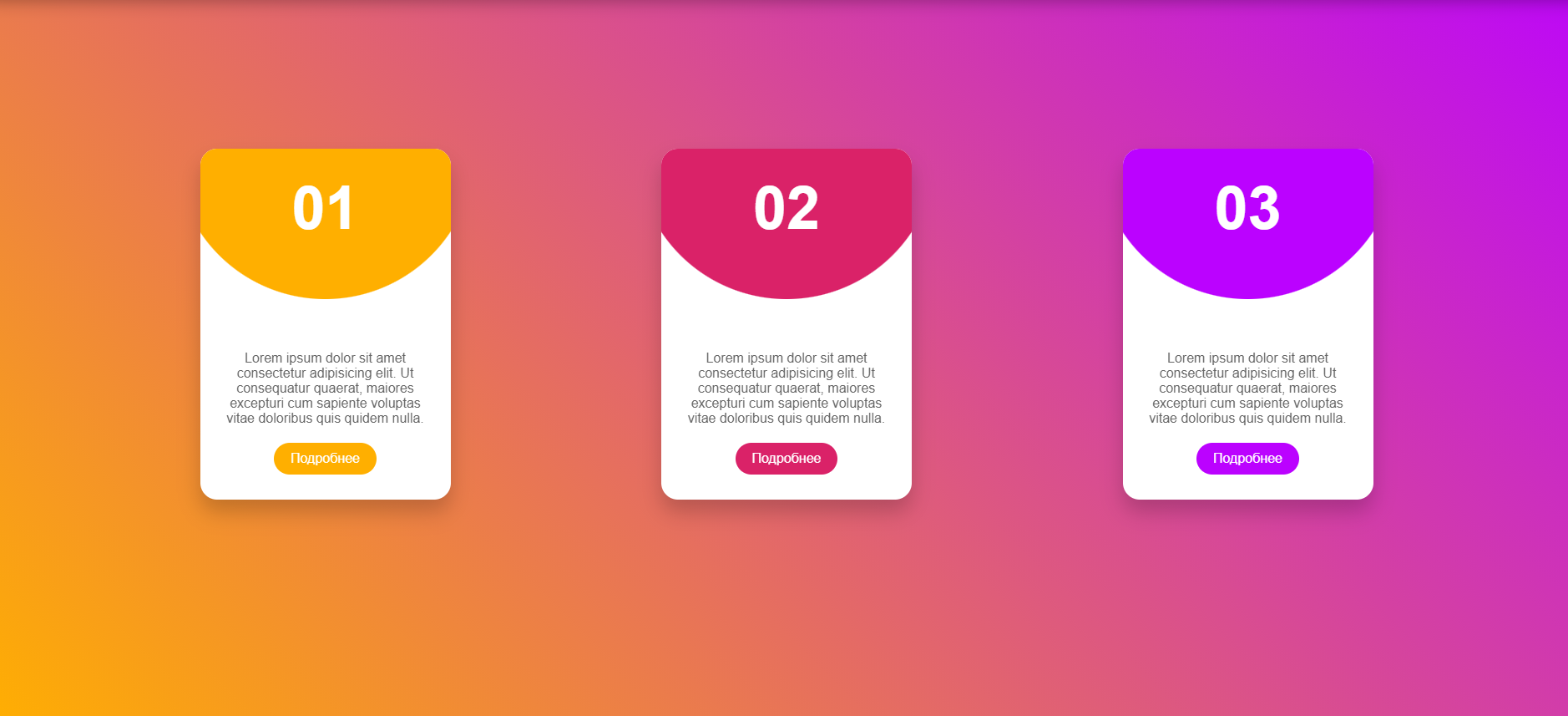




**4.** Создать три карточки по тематике своего сайта. Использовать свойства модуля flexbox, свойства transition, transform и filter. Добавить плавную анимацию при наведении.

Пример:

До наведения:



После наведения:



**5.** Создать три карточки по тематике своего сайта. Использовать свойства transition, transform и псевдоэлемент ::before. Добавить плавную покадровую анимацию при наведении, как на фото ниже: выбранная карточка становится более яркой и вдоль её границ "протекают" лучи.

Пример:

До наведения:



После наведения:

