**Выдвигающееся описание, шаг 1**

В заданиях этой серии мы создадим интерфейсный блок, похожий на прошлые, но воспользуемся в этот раз другой техникой.

Суть данной техники заключается в том, чтобы не создавать в разметке дополнительные блоки для описания, а хранить тексты в пользовательских data- атрибутах элементов. Пользовательские атрибуты, начинающиеся с data-, могут быть добавлены любому HTML-элементу для хранения текстовой информации.

Особой прелестью такого подхода является то, что мы можем обращаться к текстовому значению атрибута HTML-элемента непосредственно из CSS. Делается это с помощью свойства content, задающего генерируемое содержимое псевдоэлементов и функции attr(), которая применяется для добавления значения атрибута HTML-элемента в стилевое свойство.

Например, если у нас есть в разметке элемент <p data-text="hello">world</p>, то мы можем получить надпись «hello world», взяв «hello» из атрибута data-text в CSS и отобразив его в псевдоэлементе ::before:

p::before {

content: attr(data-text) " ";

// С помощью пустой строки в конце добавится пробел

}

В нашем случае давайте хранить тексты в атрибутах ссылки: заголовок блока в тексте атрибута data-title, а описательную часть — в атрибуте data-description. Соответственно заголовок будем выводить в качестве контента псевдоэлемента ::before, а описание — содержимым псевдоэлемента ::after.

# Закруглённые внутрь углы, шаг 1

С помощью свойства border-radius мы можем легко сглаживать края блоков и даже сделать блок визуально полностью круглым. Но, к сожалению, border-radius может сделать только выпуклые углы, а вогнутые — нет.

В этой серии заданий мы построим блок с вогнутыми углами.

Для этого, помимо ранее известных свойств, в текущей серии заданий мы воспользуемся свойством clip.

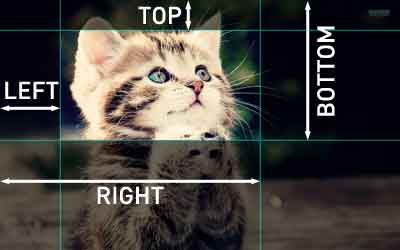
Свойство clip определяет прямоугольную область элемента, в которой будет показано его содержимое. Все, что не поместится в эту область, будет невидимым. Свойство clip работает только для абсолютно спозиционированных элементов.

Синтаксис свойства clip с функцией, вырезающей прямоугольную область:

clip: rect(верхняя-координата, правая-координата, нижняя-координата, левая-координата);

В качестве значений используется расстояние от края элемента до области вырезки, которое задаётся в единицах CSS — px, em... Если край области нужно оставить без изменений, устанавливается значение auto.

Нагляднее о том, как именно применяются координаты, показано на картинке:



Для построения блока с вогнутыми краями создадим поочерёдно для каждого угла свой маскирующий блок.

Начнём с нижнего левого угла.

Мы будем задавать номера ссылок автоматически с помощью CSS. Для этого используем следующие свойства:

* counter-reset, которое позволяет создать переменную-счётчик;
* counter-increment, которое позволяет увеличивать значение счётчика;
* функцию counter(), которая позволяет передавать значение счётчика свойству content у псевдоэлементов.

В исходном CSS для .breadcrumbs уже добавлено свойство counter-reset: flag. То есть наш счётчик уже создан и называется flag.

Чтобы увеличивать счётчик, нужно добавить свойство counter-increment: flag к каждому элементу, который будет нумероваться. А чтобы считывать значение счётчика и передавать его в псевдоэлементы, нужно задать им свойство content: counter(flag).

Логично и изменение нумерации, и её отображение производить в самом ::before у ссылок:

.breadcrumbs a::before {

content: counter(flag);

counter-increment: flag;

...

}

Ну и, конечно же, добавим базовое оформление для номеров ссылок.