

Эффективный рендеринг SVG-стрелок: оптимизация для графов с сотнями объектов

Александр Мальцев
Yandex Crowd

О себе



Александр
Мальцев

Разработчик интерфейсов
в Yandex Crowd

- ▢ Разработчик интерфейсов в Яндексе **2+** года
- ▢ Интересуюсь статическим анализом кода и 2D-графикой в вебе

Задачи Yandex Crowd

Тестирование приложений

Текста и дизайн

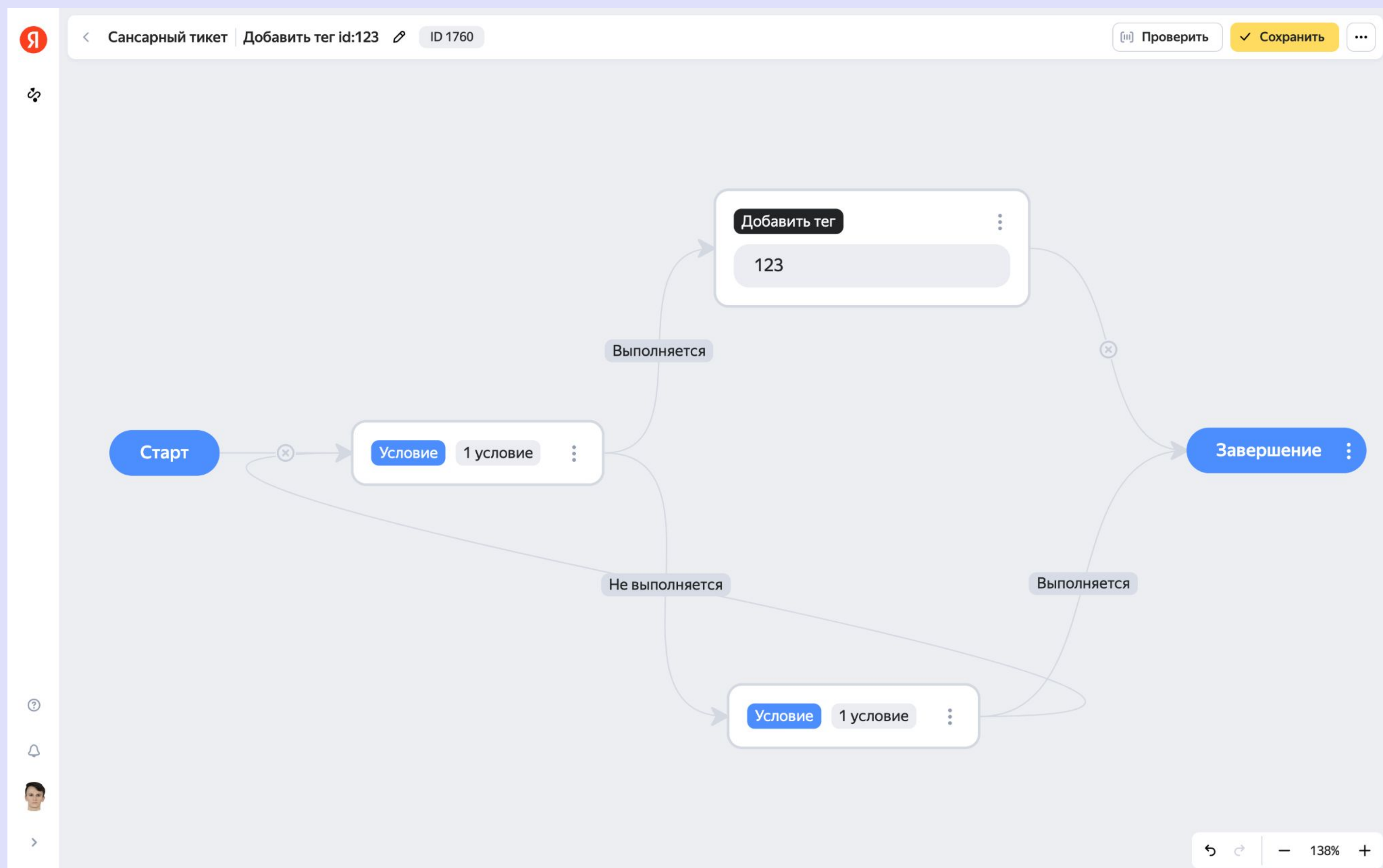
Разметка данных

Модерация контента

Поддержка пользователей

Локализация и документирование

Полевые задачи





Описание задачи

- ✓ Необходимо реализовать стрелки для однонаправленного графа
- ✓ Граф должен быть интерактивным
- ✓ Блоки могут иметь несколько входящих и исходящих связей
- ✓ Графы могут иметь большое количество элементов, 500+ вполне реальный случай
- ✓ Проект на React

Обзор существующих решений

Технология

Canvas

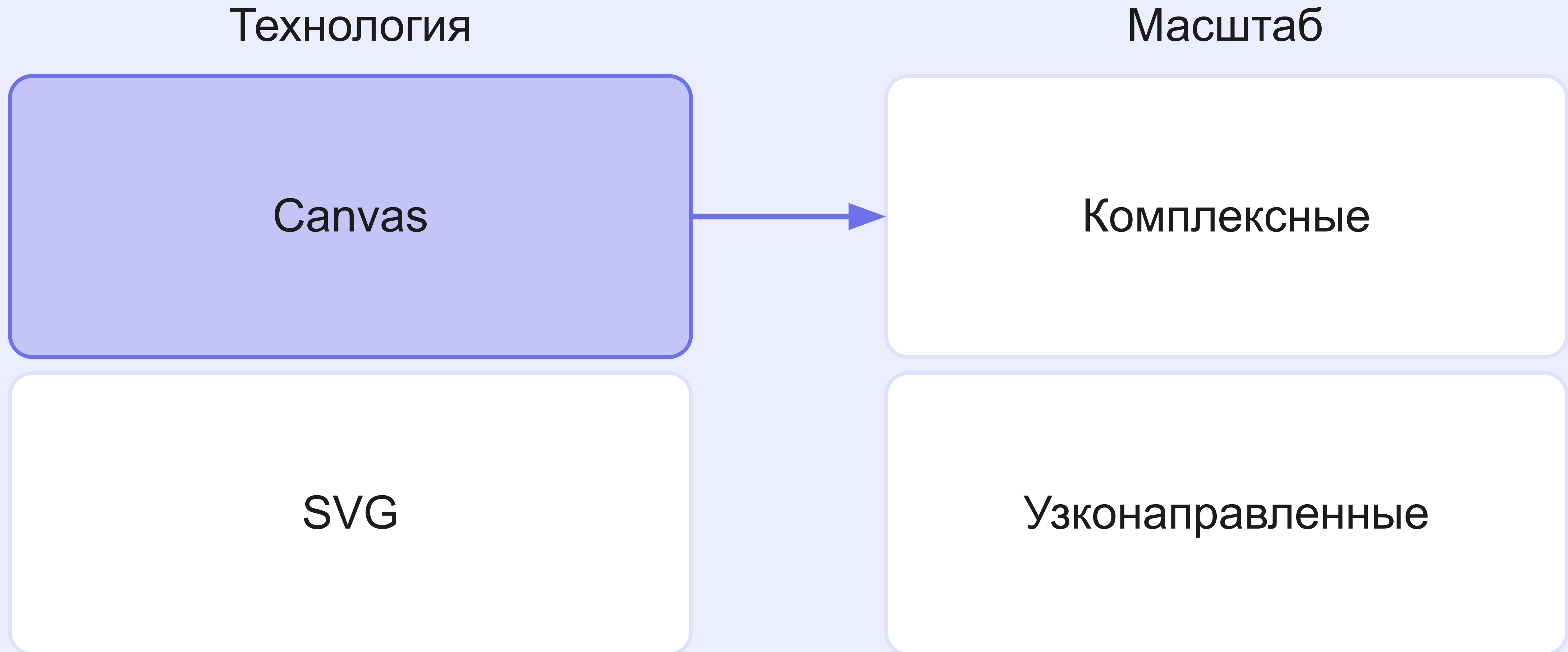
SVG

Масштаб

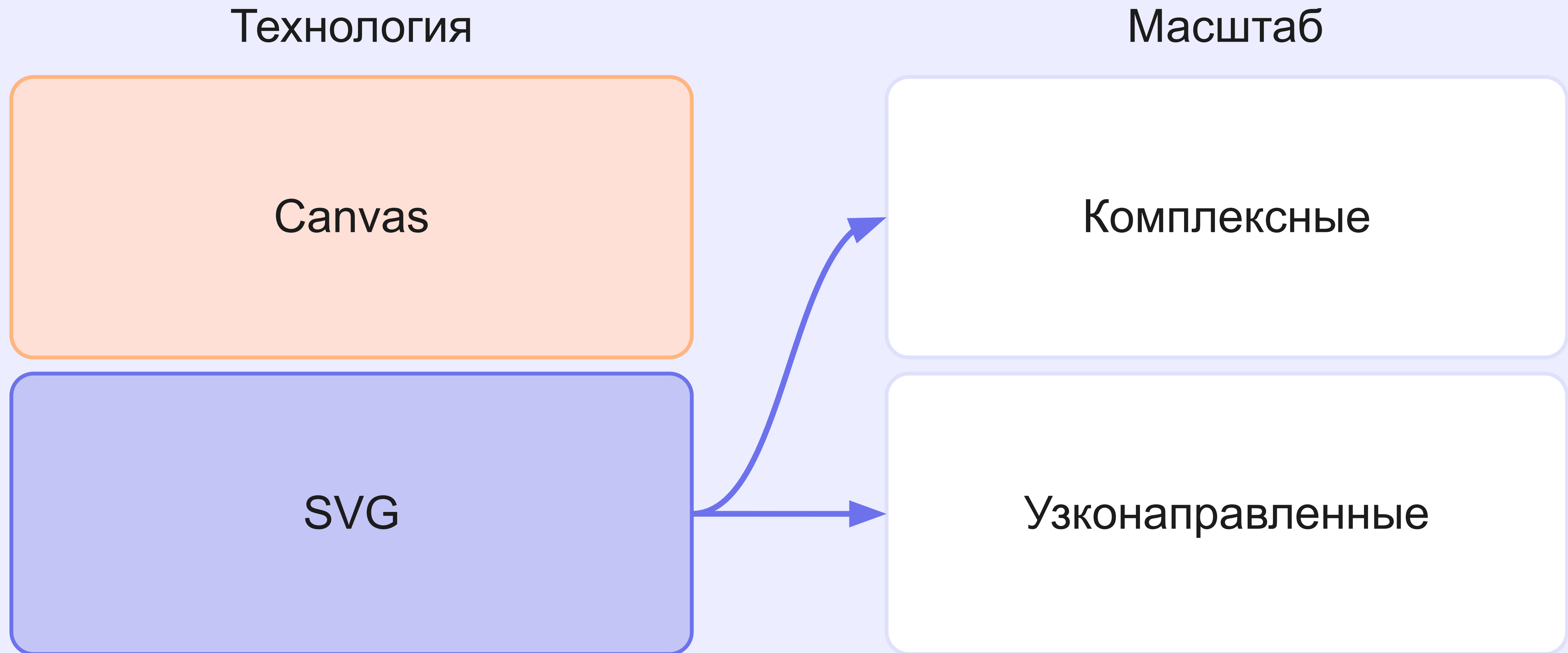
Комплексные

Узконаправленные

Обзор существующих решений



Обзор существующих решений



Обзор существующих решений

Технология

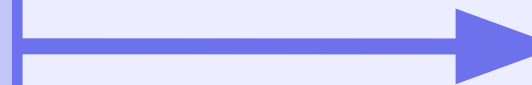
Масштаб

Canvas

Комплексные

SVG

Узконаправленные



SVG vs Canvas

- Меньше реализовывать самостоятельно
- Проще кастомизация
- Меньше вес



Преимущества SVG

- Могут возникнуть трудности с оптимизацией на большом количестве элементов



Недостатки SVG

Κορотно про SVG

SVG line, rect, circle, ellipse

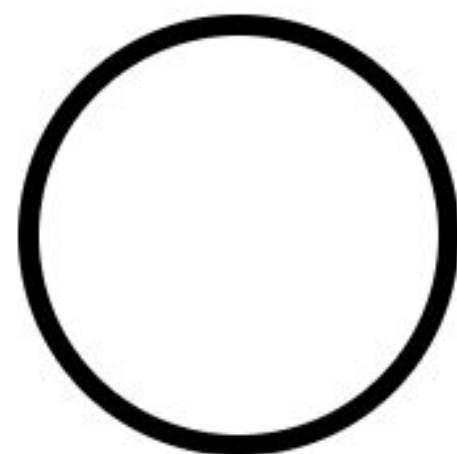
```
<line x1="10" y1="10" x2="90" y2="30" />
```



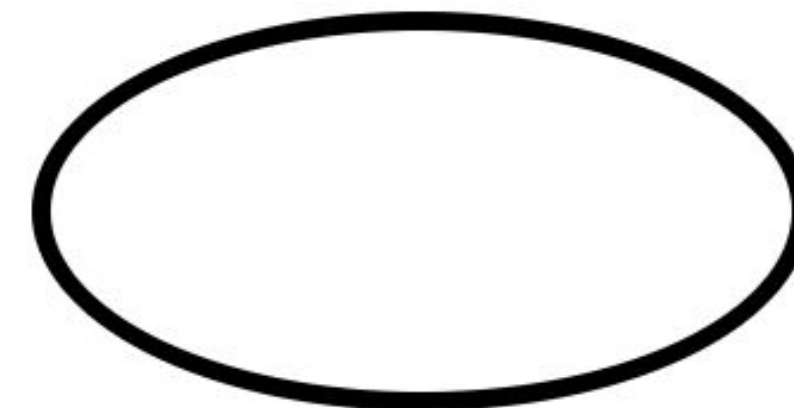
```
<rect x="10" y="10" width="90" height="30" />
```



```
<circle cx="50" cy="20" r="10" />
```

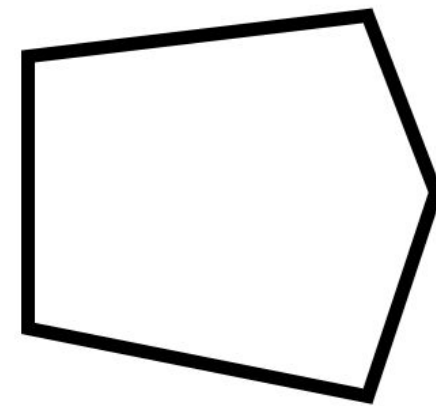


```
<ellipse cx="50" cy="20" rx="20" ry="10" />
```

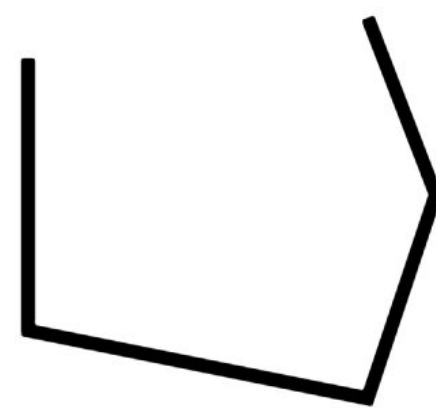


SVG polygon & polyline

```
<polygon points="5,5 5,25 30,30 35,15 30,2" />
```



```
<polyline points="5,5 5,25 30,30 35,15 30,2" />
```



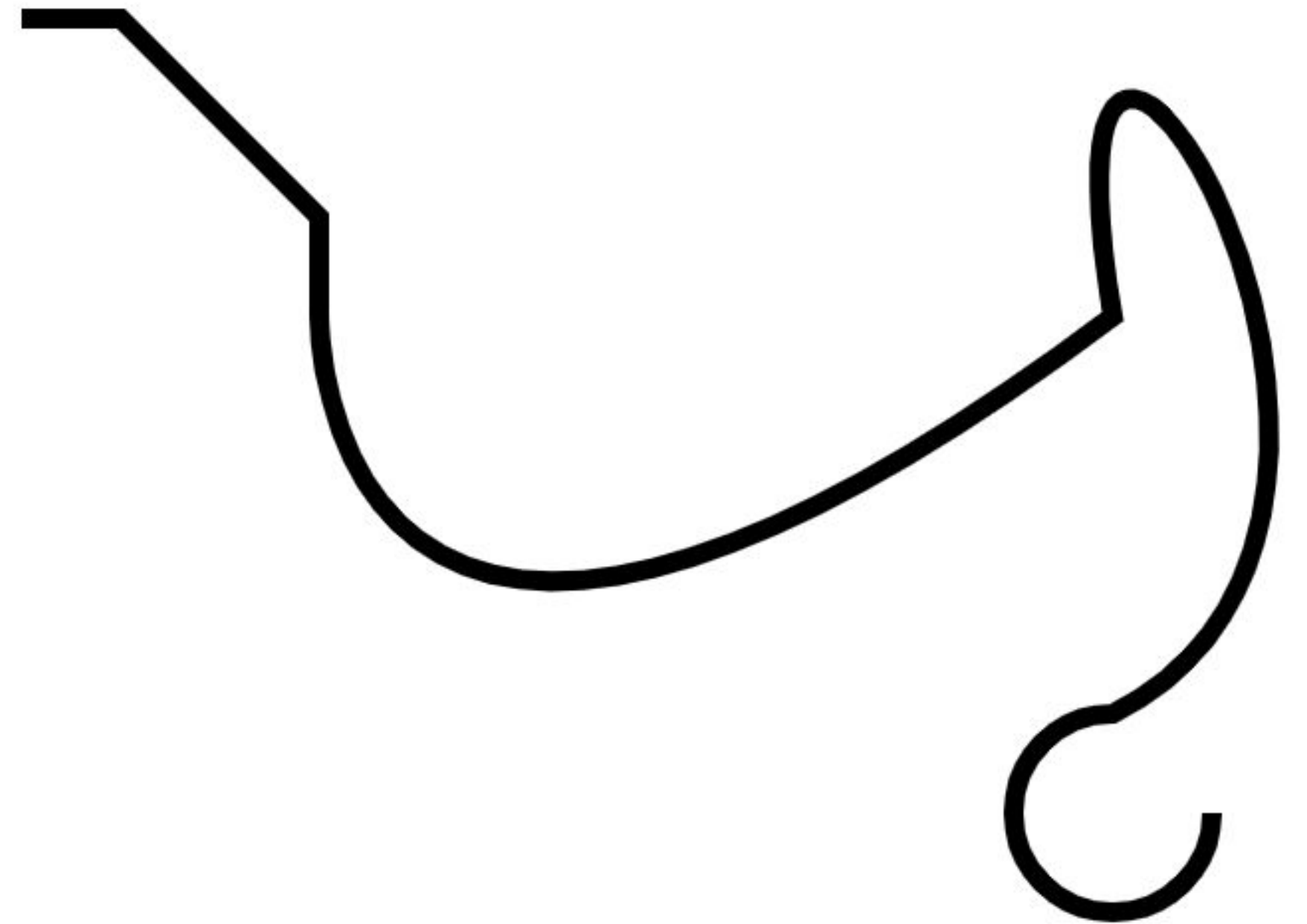
SVG text

```
<text x="20" y="20">PiterJS</text>
```

PiterJS

SVG path

```
<path d="M 5,5 h 5 l 10,10 v  
5 s 0,30 40,0 c -5,-30 20,10  
0,20 a 5 5 180 1 0 5,5" />
```



Синтаксис прямых линий и окружности

Команда	Действие	Параметры
m	move to	x,y
v	vertical line	y
h	horizontal line	x
l	line	x,y
a	arc	rx ry angle large-arc-flag sweep-flag dx dy

Синтаксис кривых Безье

Команда	Действие	Параметры
c	cubic bezier	x1,y1 x2,y2 x,y
s	smooth cubic bezier	x2,y2 x,y
q	quadratic bezier	x1,y1 x,y
t	smooth quadratic bezier	x,y

3 важных правила синтаксиса path

Начинается всегда с **M / m**

01

Заглавная буква команды —
абсолютные координаты

Строчная буква команды —
относительные координаты

02

Одна команда продолжает другую
(использует последнюю точку
предыдущей как свою первую)

Например, M 5,5 L 10,10 — переместить
в (5, 5), нарисовать прямую из (5, 5) в (10, 10)

03

SVG path

```
<path d="
```

```
M 5,5 ←
```

```
h 5
```

```
l 10,10
```

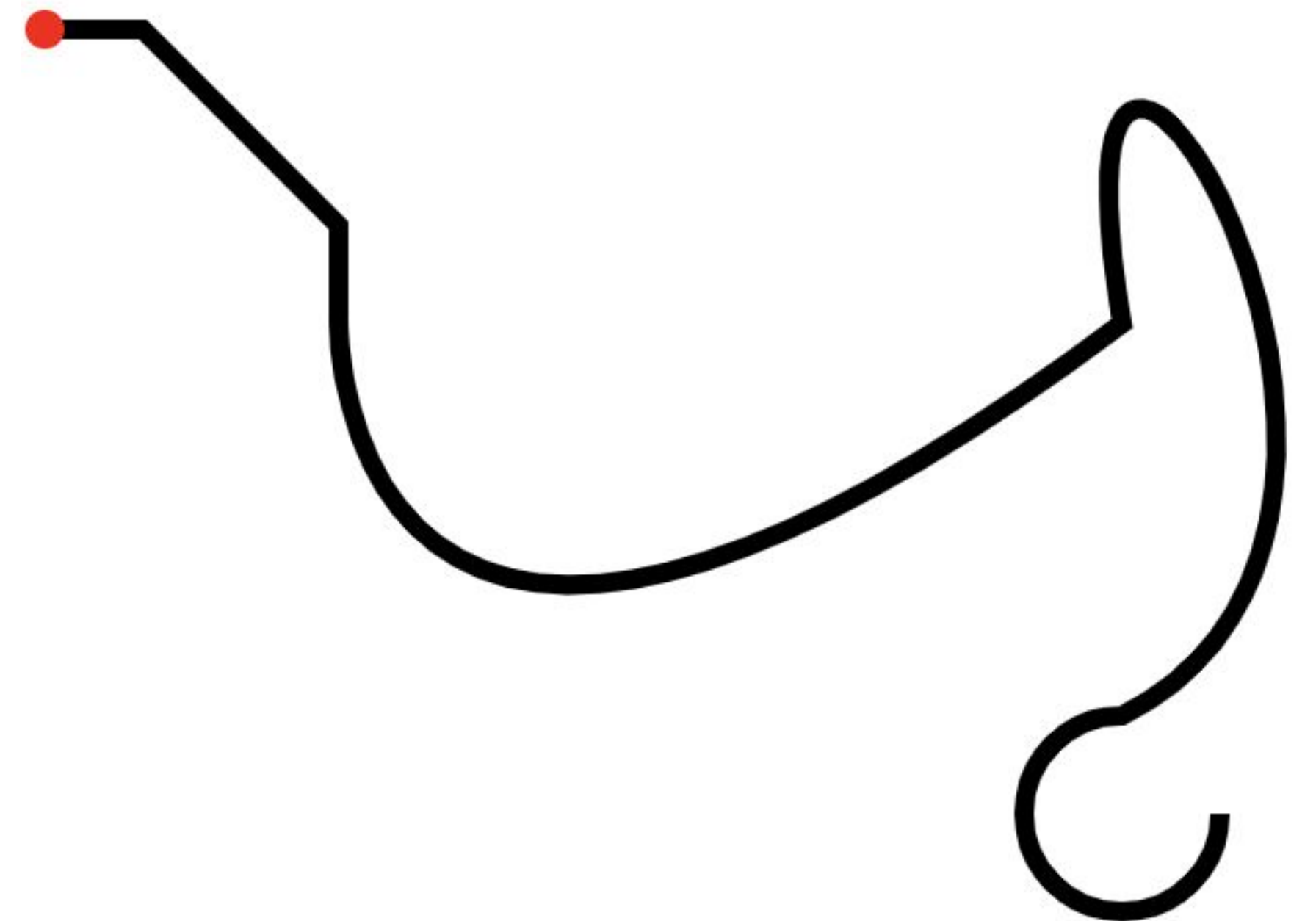
```
v 5
```

```
s 0,30 40,0
```

```
c -5,-30 20,10 0,20
```

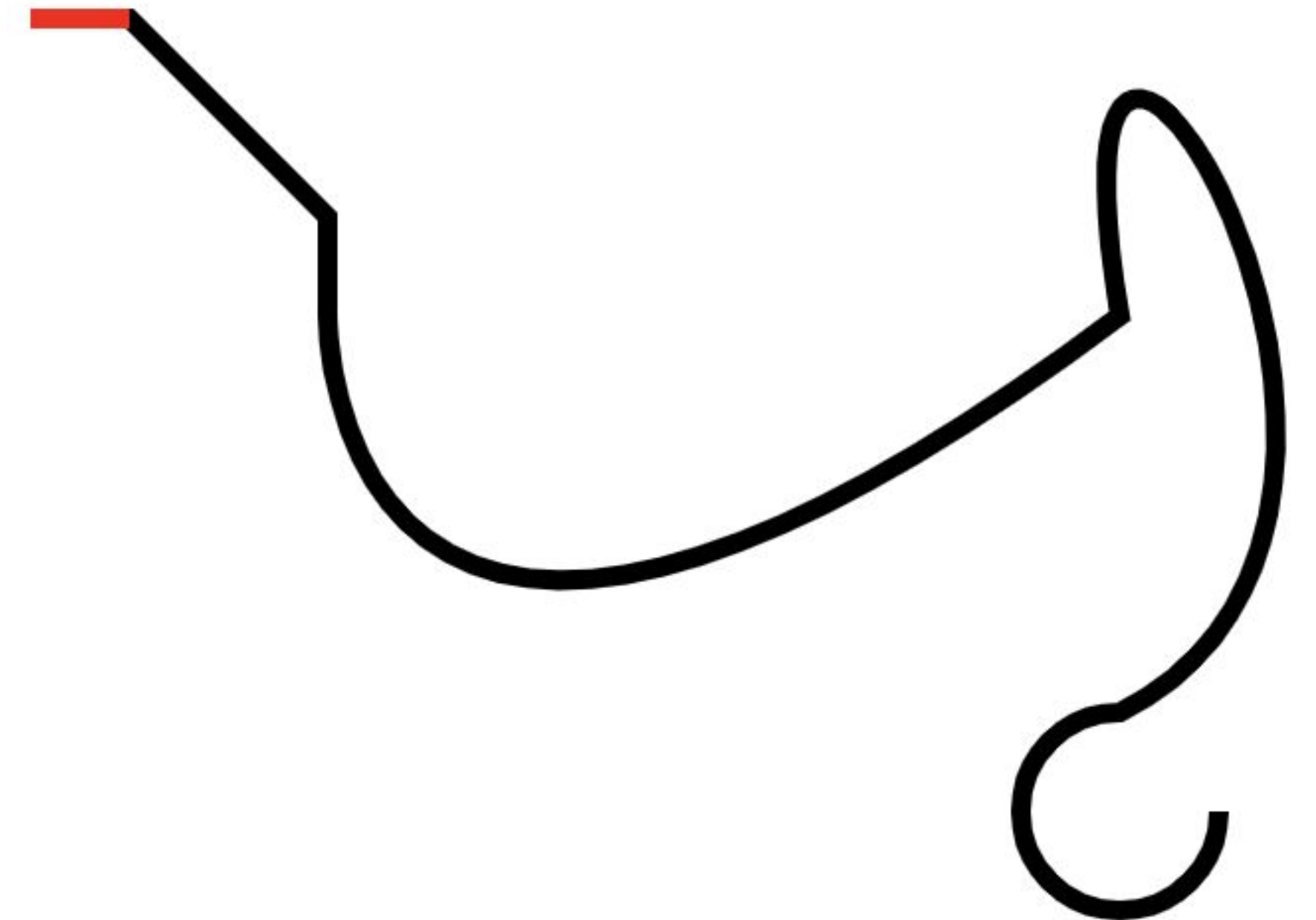
```
a 5 5 180 1 0 5,5"
```

```
/>
```



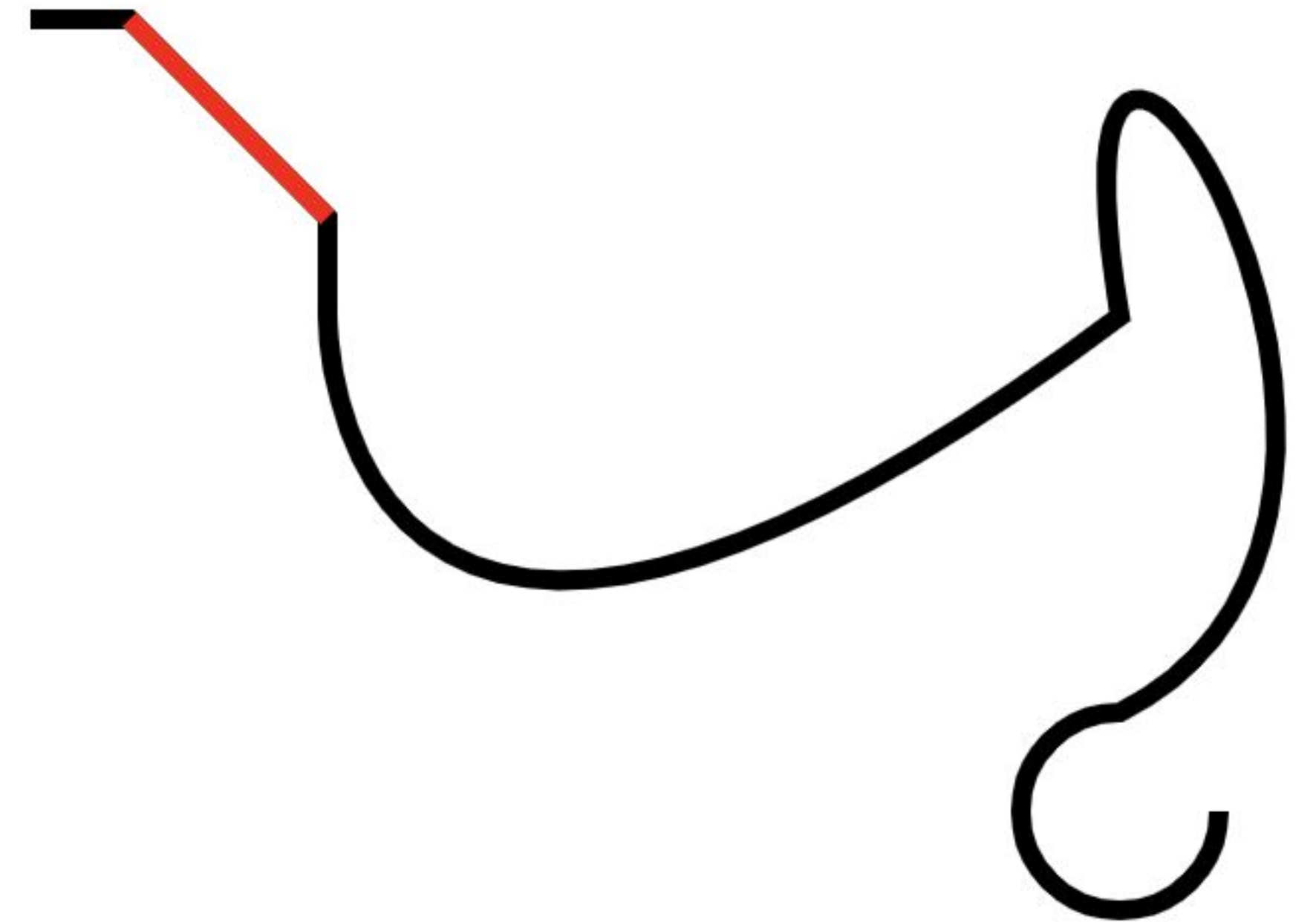
SVG path

```
<path d="
  M 5,5
  h 5 ←
  l 10,10
  v 5
  s 0,30 40,0
  c -5,-30 20,10 0,20
  a 5 5 180 1 0 5,5"
/>
```



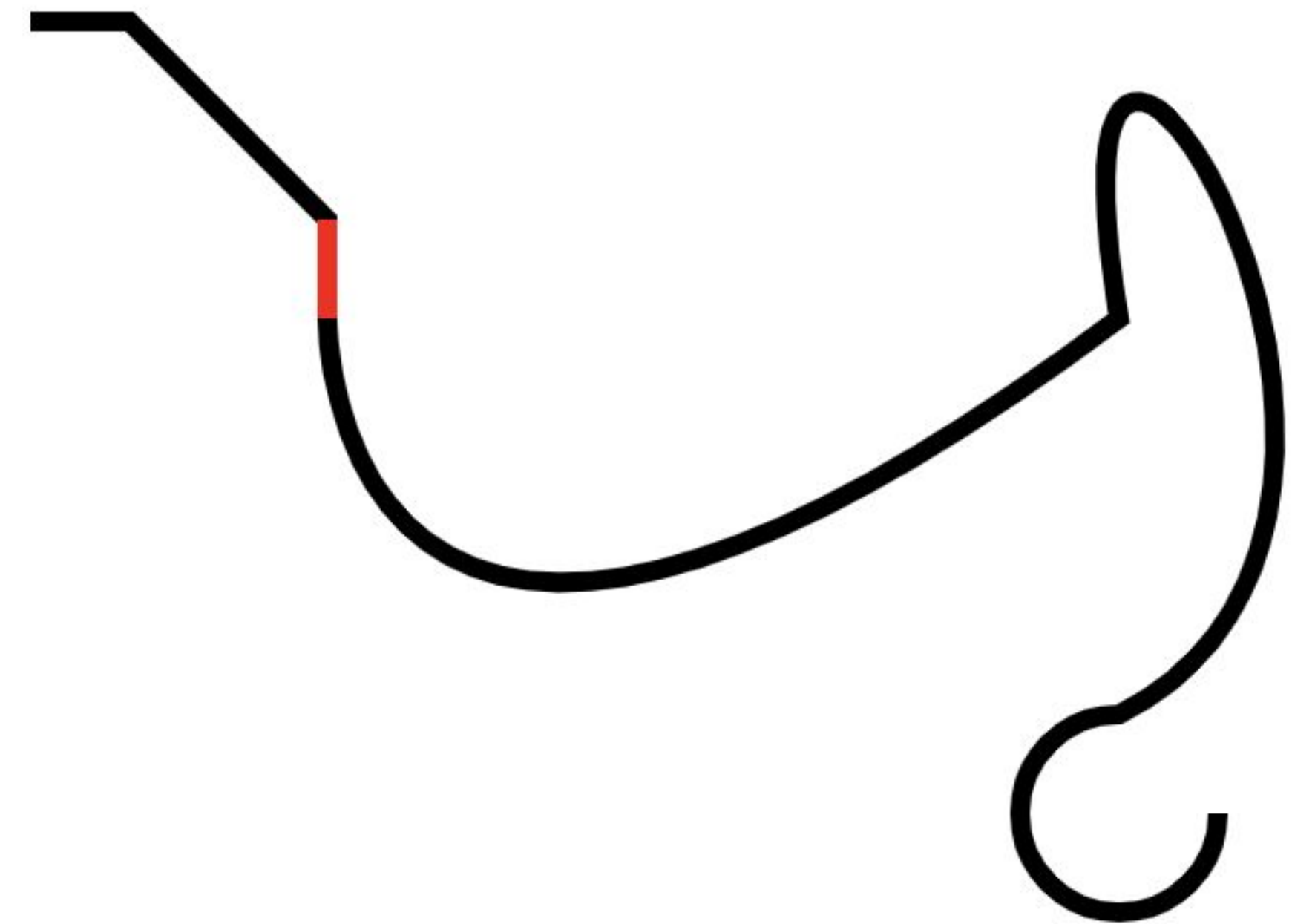
SVG path

```
<path d="
  M 5,5
  h 5
  l 10,10 ←
  v 5
  s 0,30 40,0
  c -5,-30 20,10 0,20
  a 5 5 180 1 0 5,5"
/>
```



SVG path

```
<path d="
  M 5,5
  h 5
  l 10,10
  v 5 ←
  s 0,30 40,0
  c -5,-30 20,10 0,20
  a 5 5 180 1 0 5,5"
/>
```



SVG path

```
<path d="
```

```
  M 5,5
```

```
  h 5
```

```
  l 10,10
```

```
  v 5
```

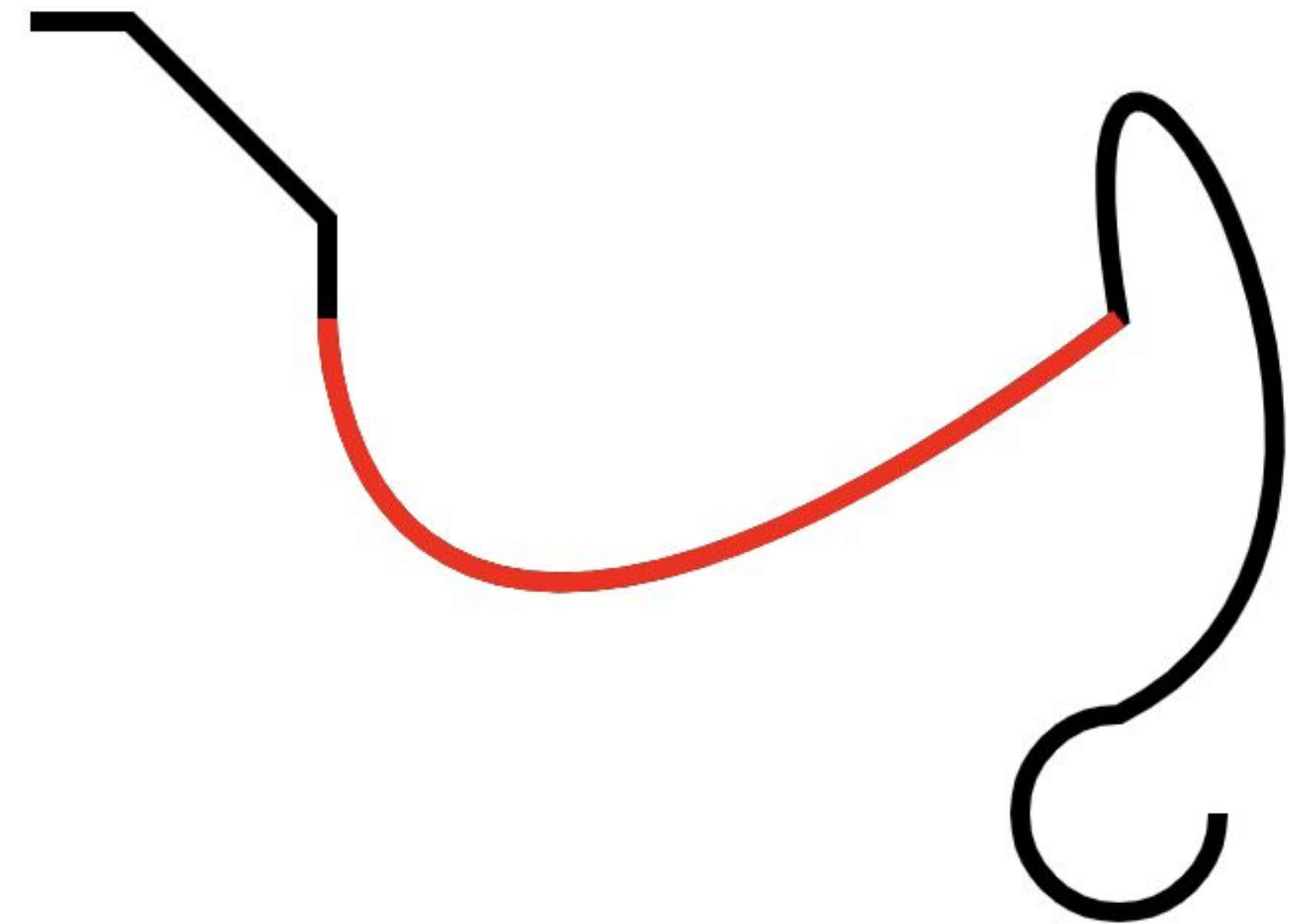
```
  s 0,30 40,0
```



```
  c -5,-30 20,10 0,20
```

```
  a 5 5 180 1 0 5,5"
```

```
/>
```



SVG path

```
<path d="
```

```
M 5,5
```

```
h 5
```

```
l 10,10
```

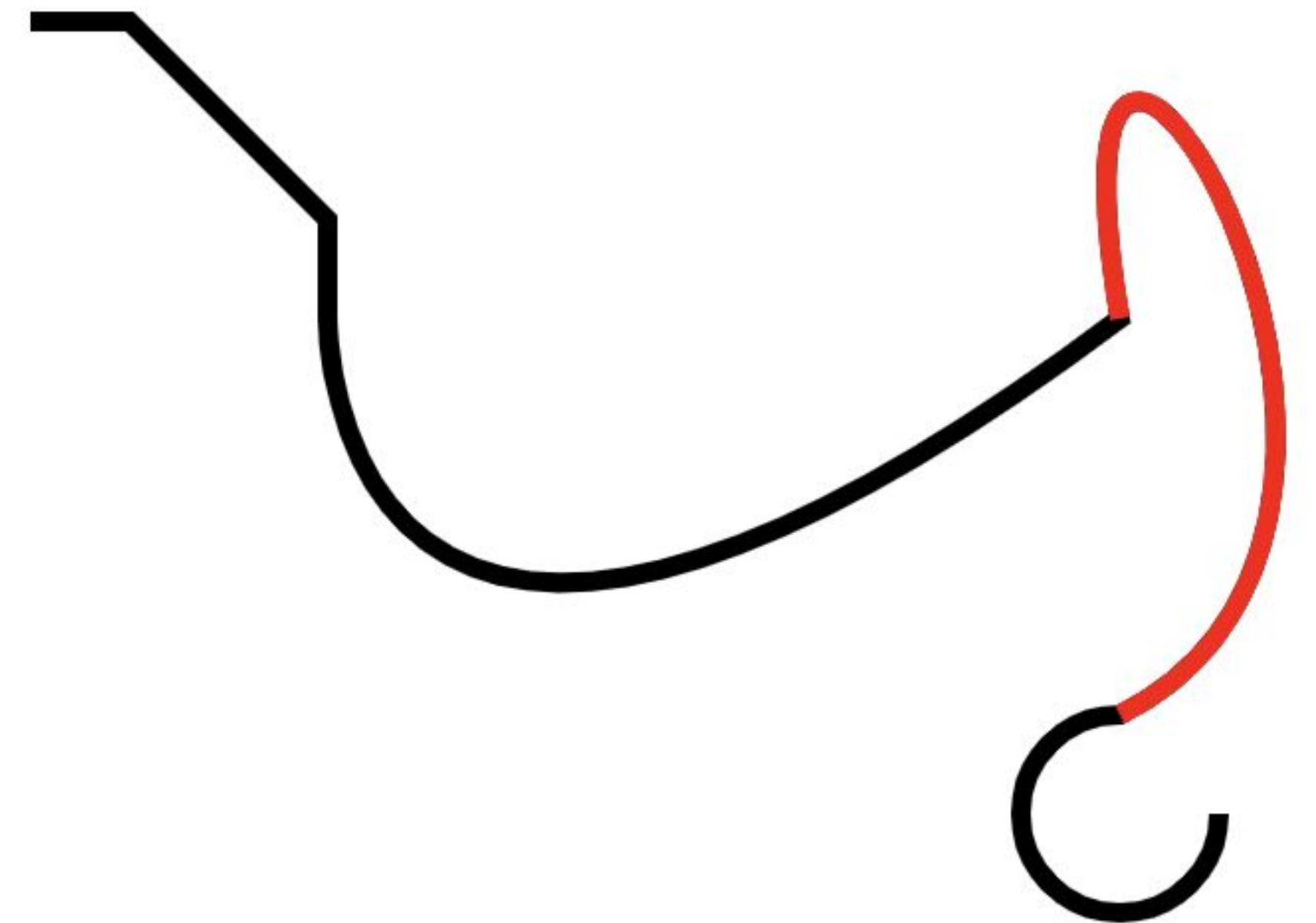
```
v 5
```

```
s 0,30 40,0
```

```
c -5,-30 20,10 0,20 ←
```

```
a 5 5 180 1 0 5,5"
```

```
/>
```



SVG path

```
<path d="
```

```
M 5,5
```

```
h 5
```

```
l 10,10
```

```
v 5
```

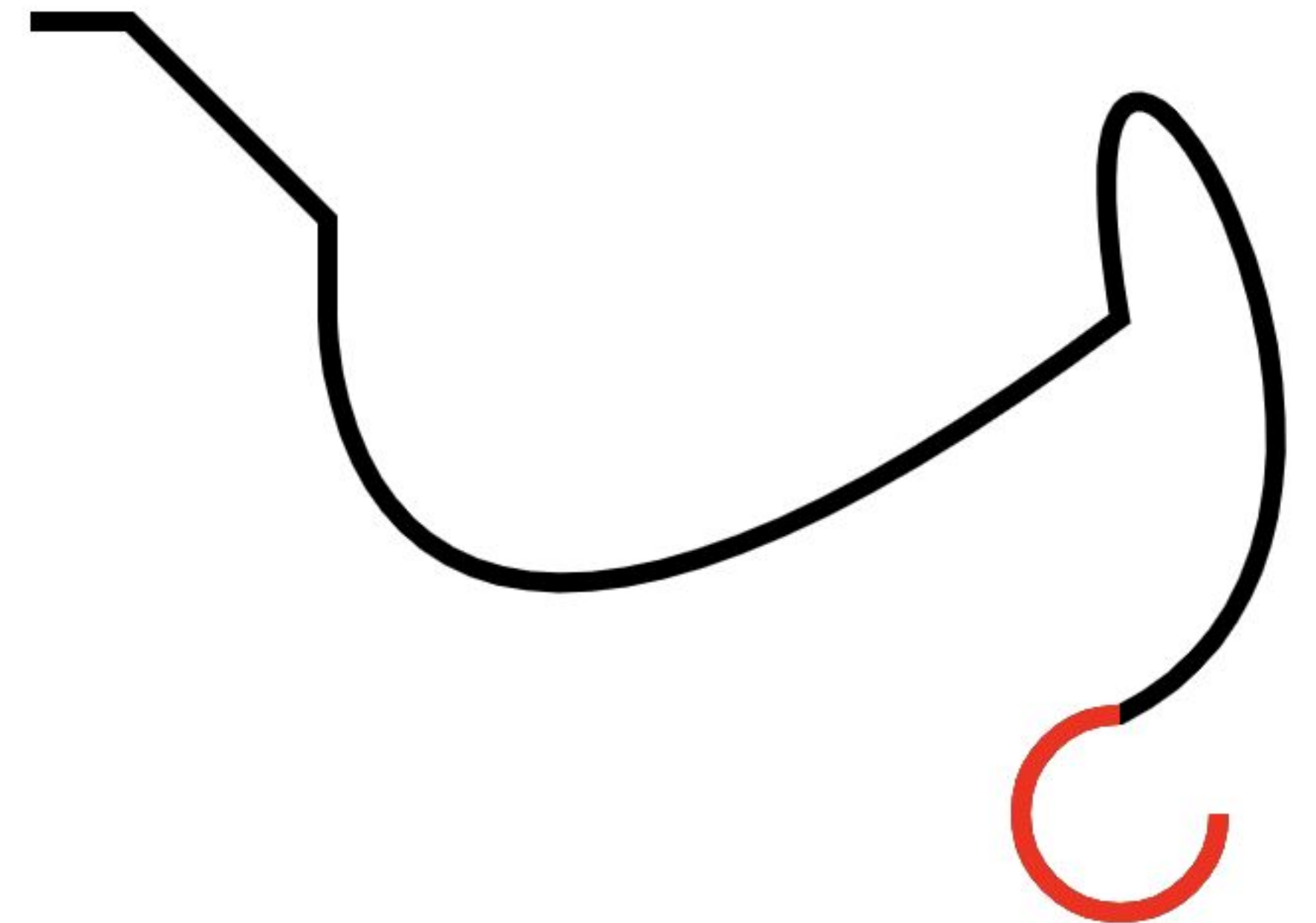
```
s 0,30 40,0
```

```
c -5,-30 20,10 0,20
```

```
a 5 5 180 1 0 5,5"
```



```
/>
```



Маркеры в SVG

Элемент `<marker>` определяет графический элемент, используемый для рисования стрелок или полимаркеров на элементах `<path>`, `<line>`, `<polyline>` или `<polygon>`

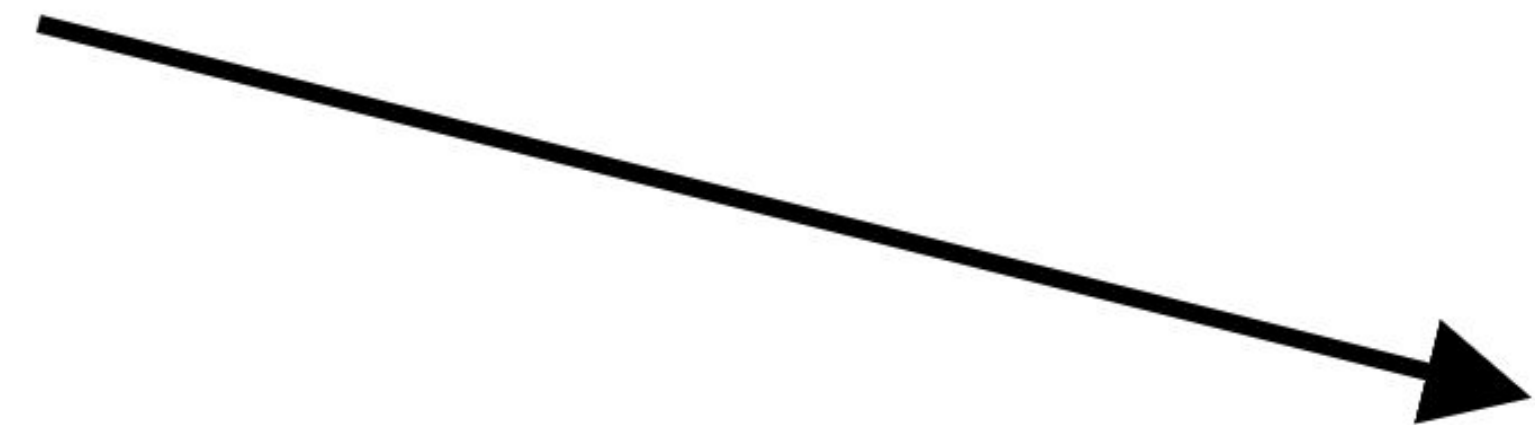
Маркеры можно прикреплять, используя свойства `marker-start`, `marker-mid` и `marker-end` *

* — Определение с MDN

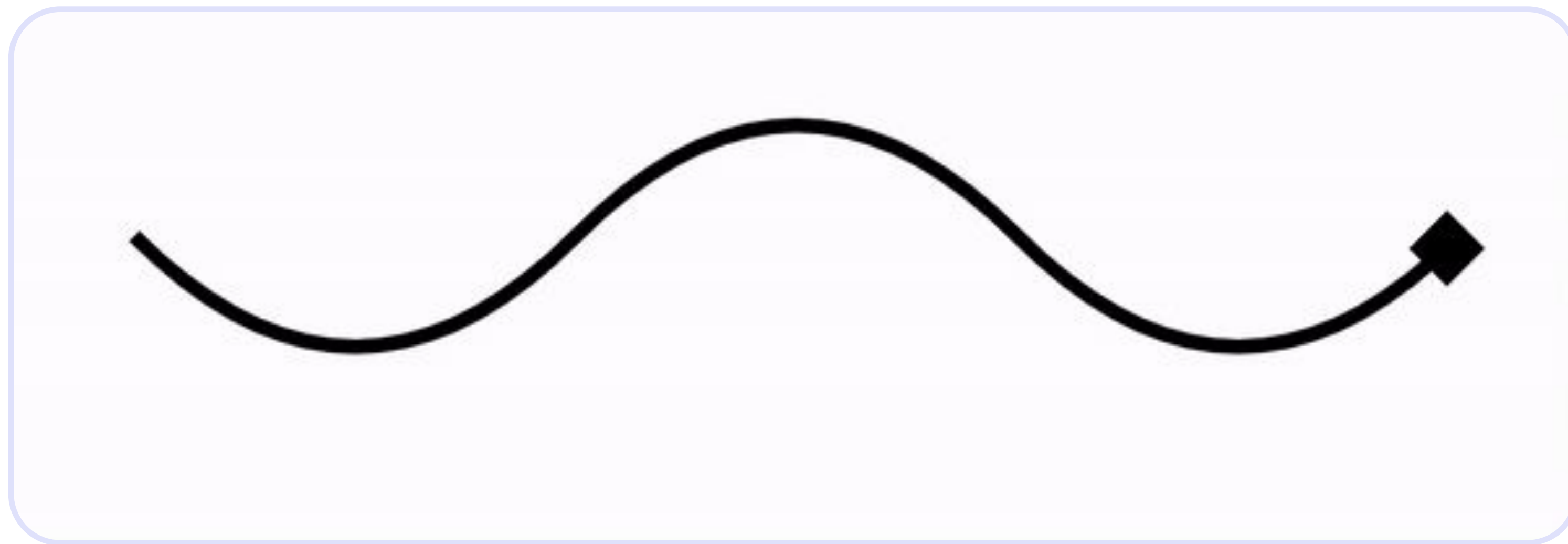


Маркеры в SVG

```
<marker
  id="arrow" viewBox="0 0 10 10" refX="5" refY="5"
  markerWidth="6" markerHeight="6" orient="auto"
>
  <path d="M 0 0 L 10 5 L 0 10 z" />
</marker>
<line
  x1="10" y1="10"
  x2="90" y2="30"
  stroke="black"
  marker-end="url(#arrow)"
/>
```



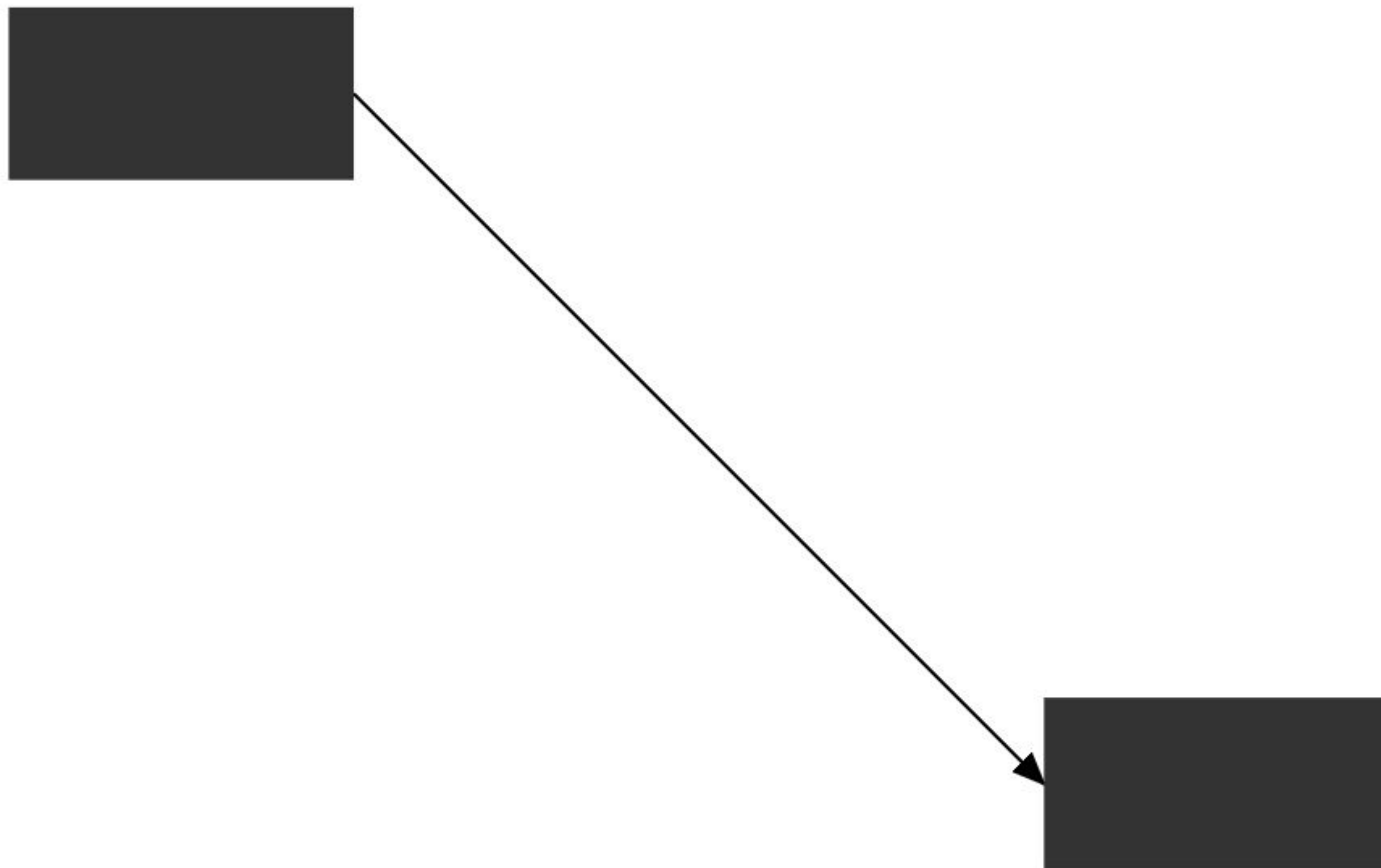
Маркеры в SVG



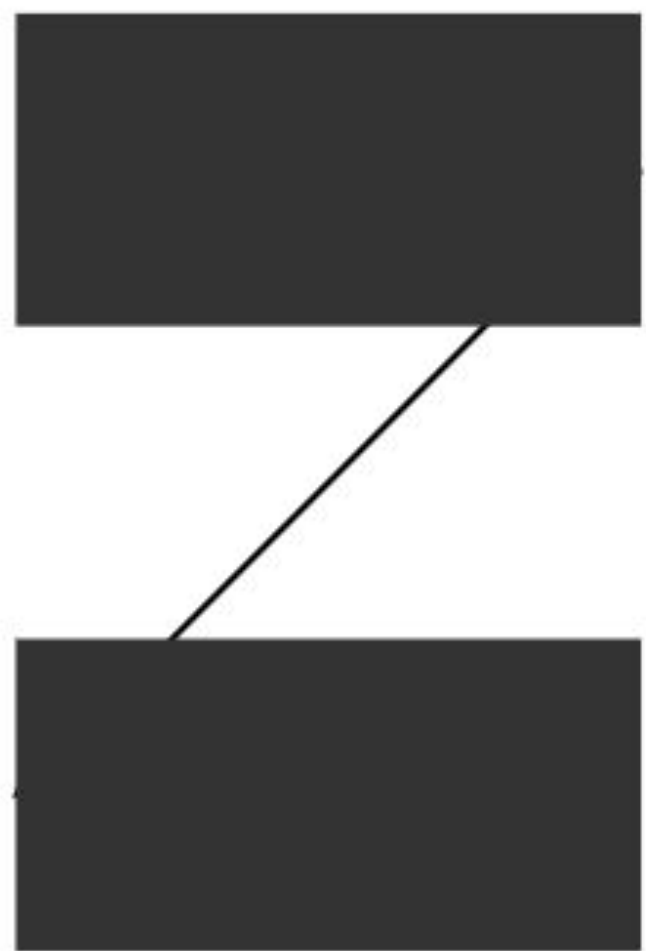
Маркеры можно анимировать

Форма линии

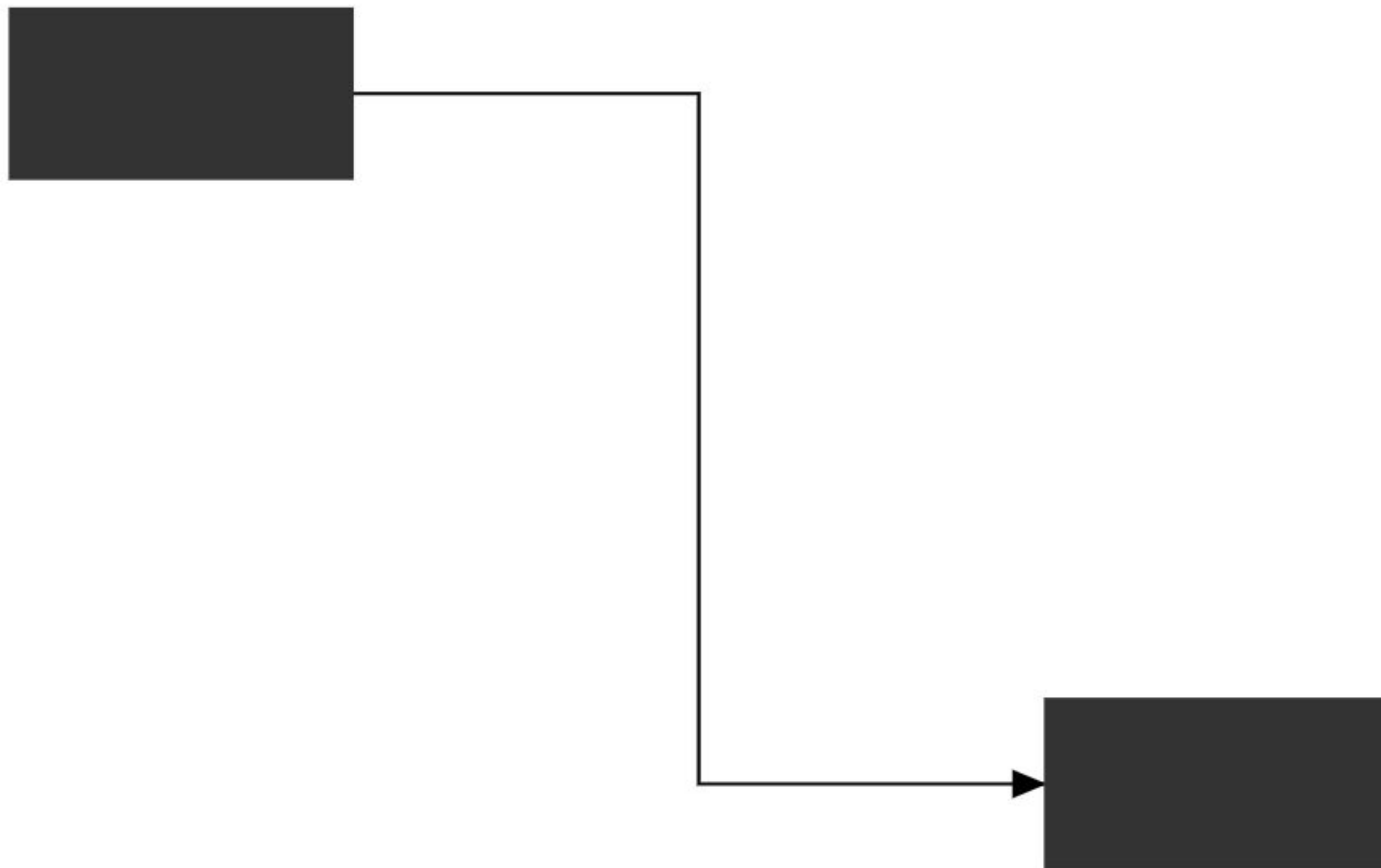
Как нарисовать «красивую» стрелку?



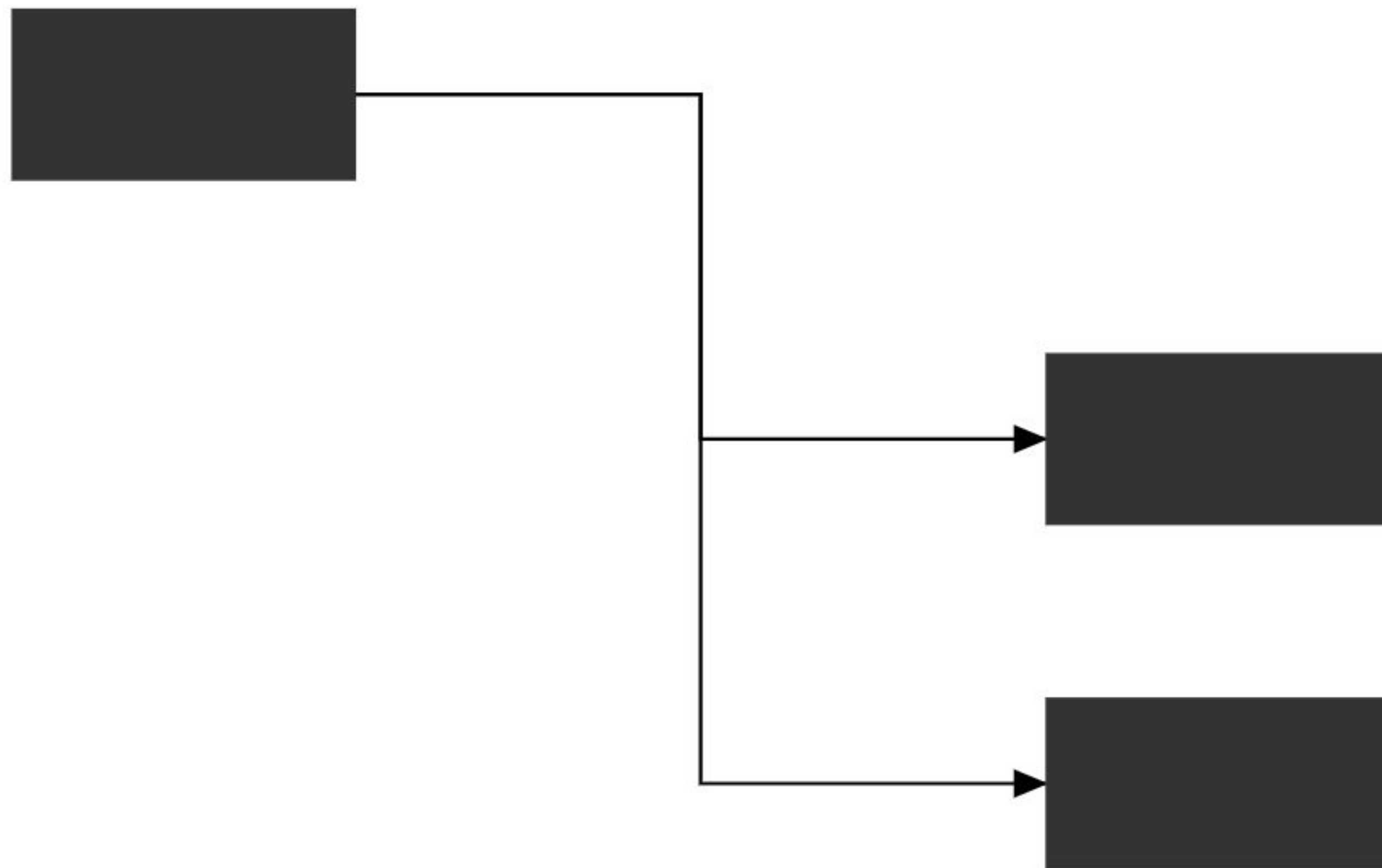
Как нарисовать «красивую» стрелку?



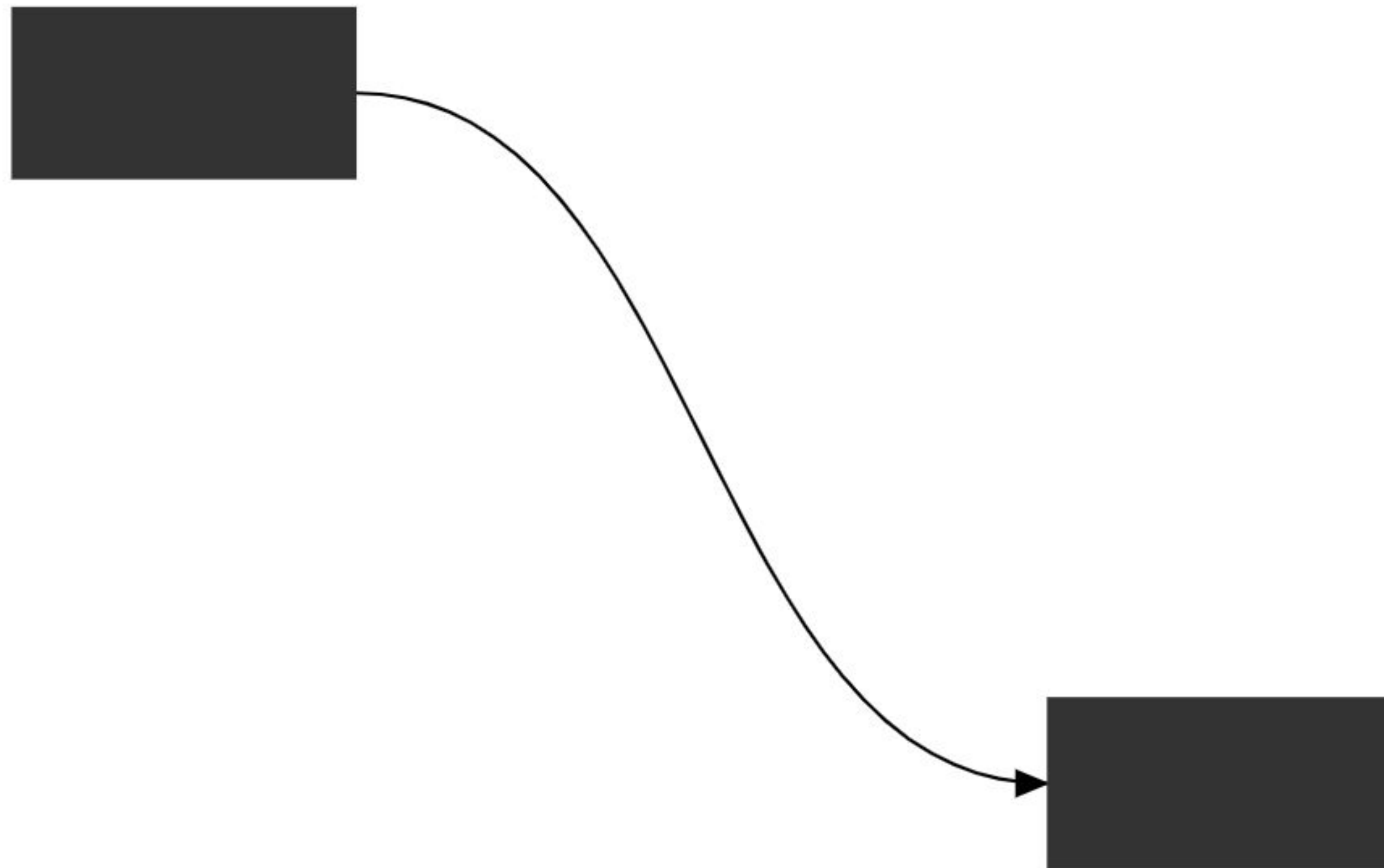
Как нарисовать «красивую» стрелку?



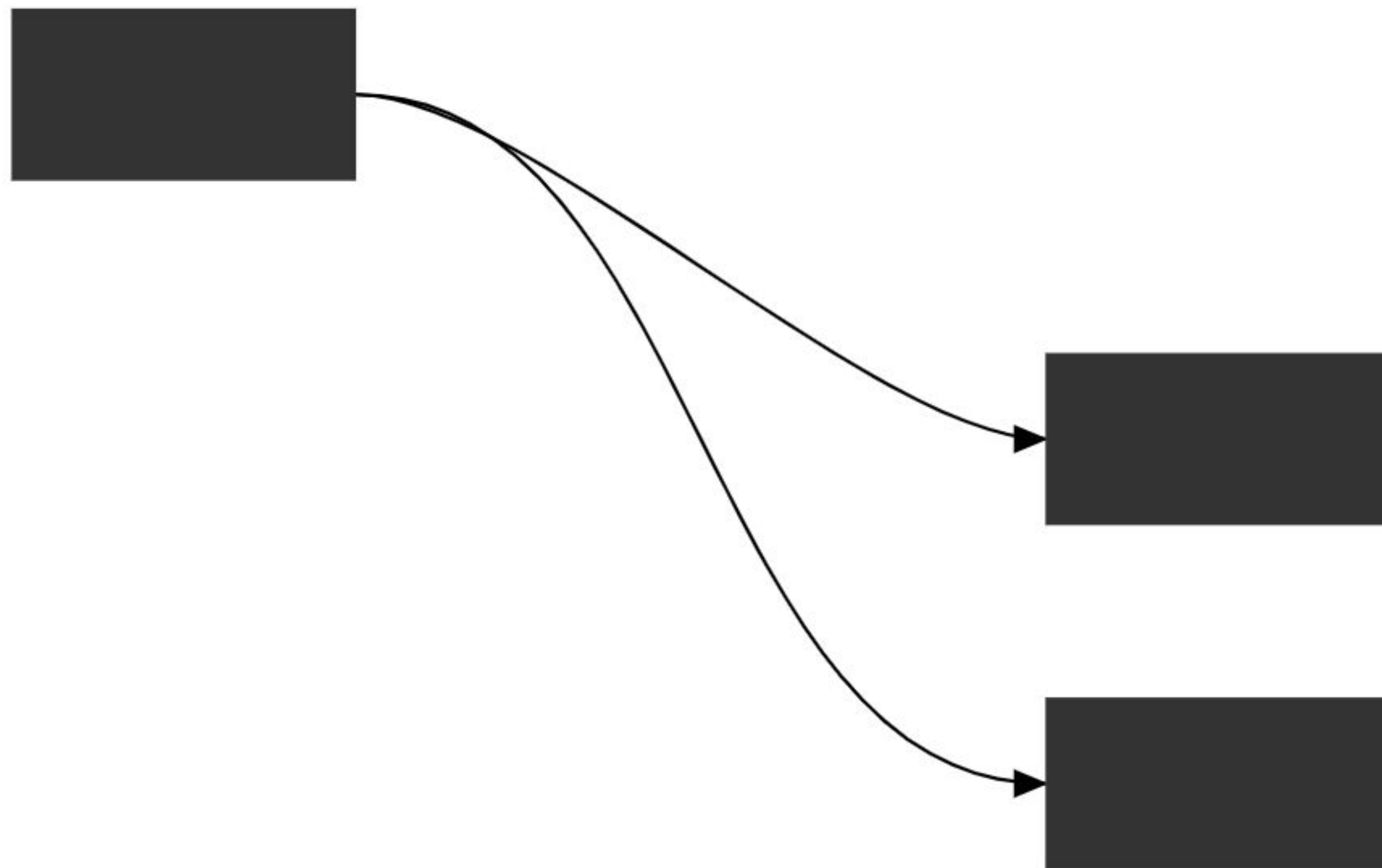
Как нарисовать «красивую» стрелку?



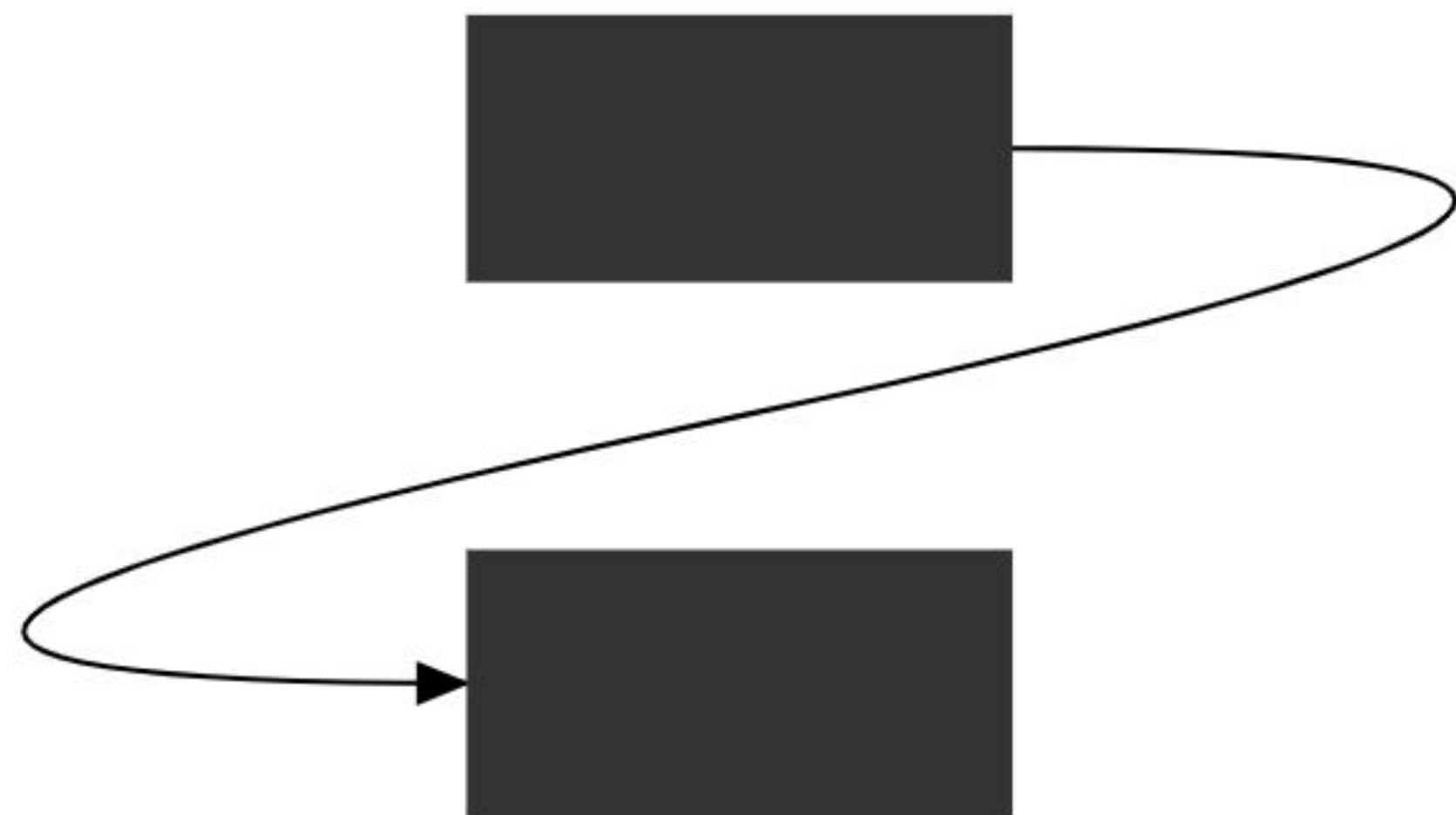
Как нарисовать «красивую» стрелку?



Как нарисовать «красивую» стрелку?



Как нарисовать «красивую» стрелку?



Что такое кривые Безье?

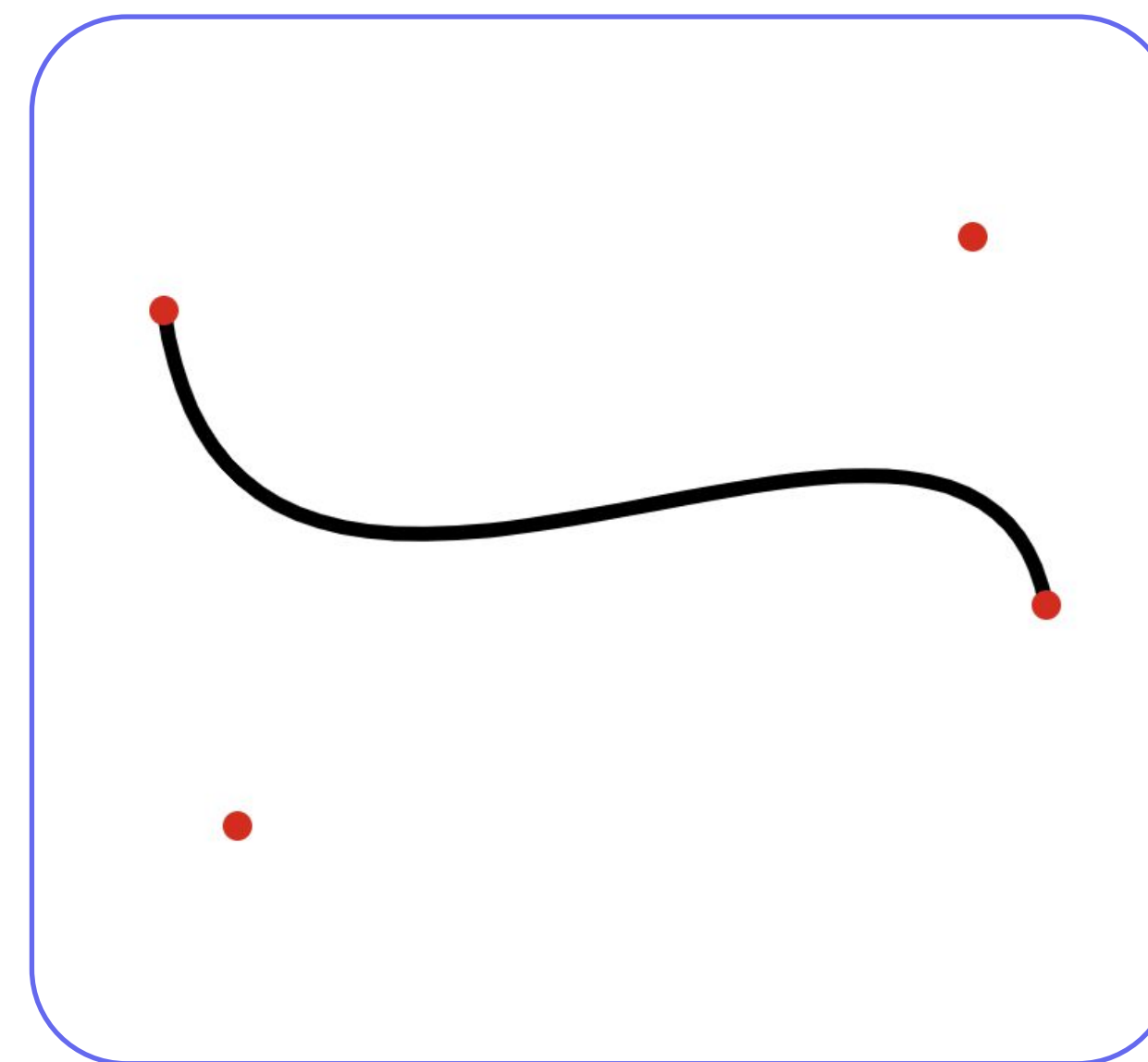
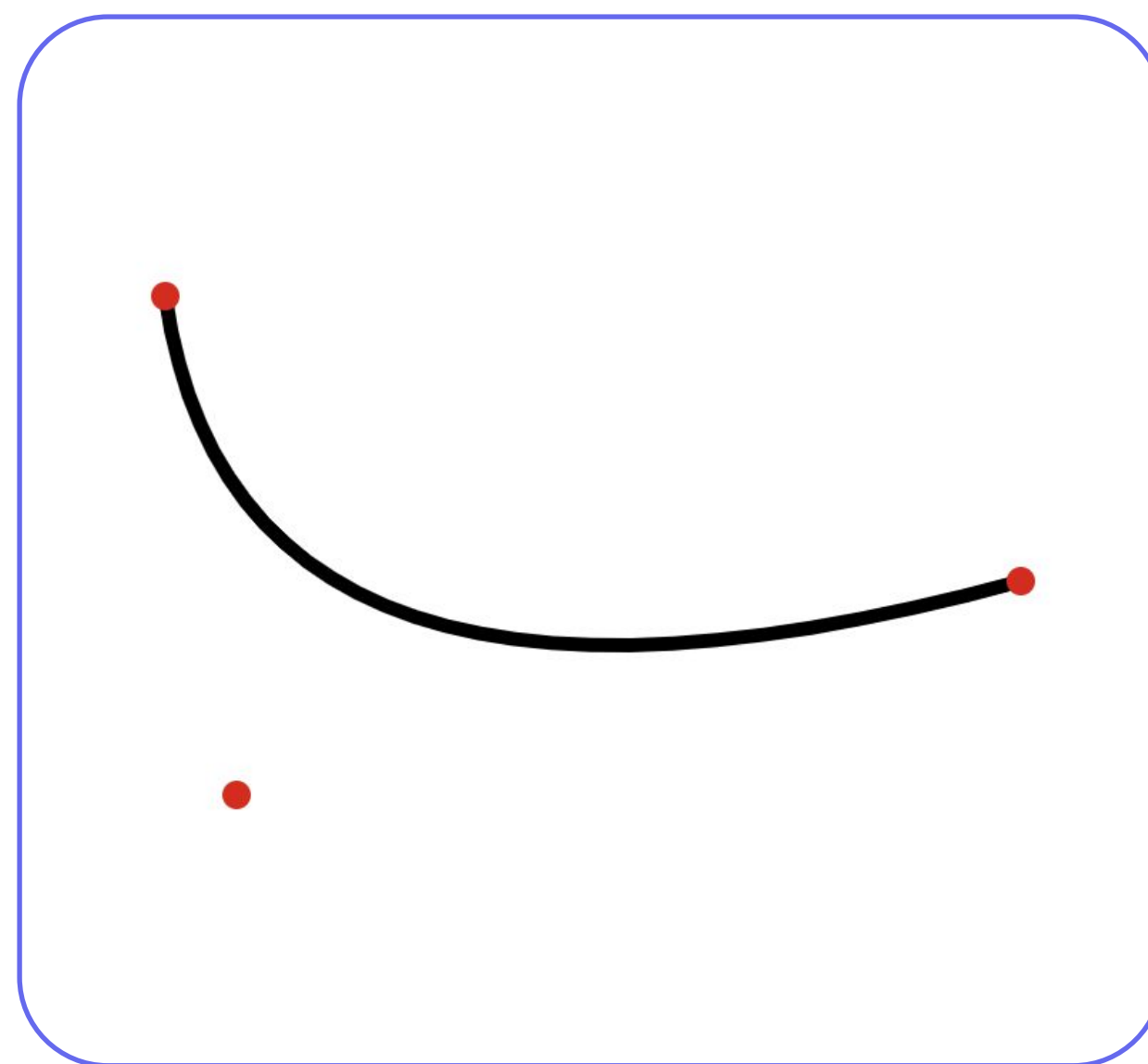
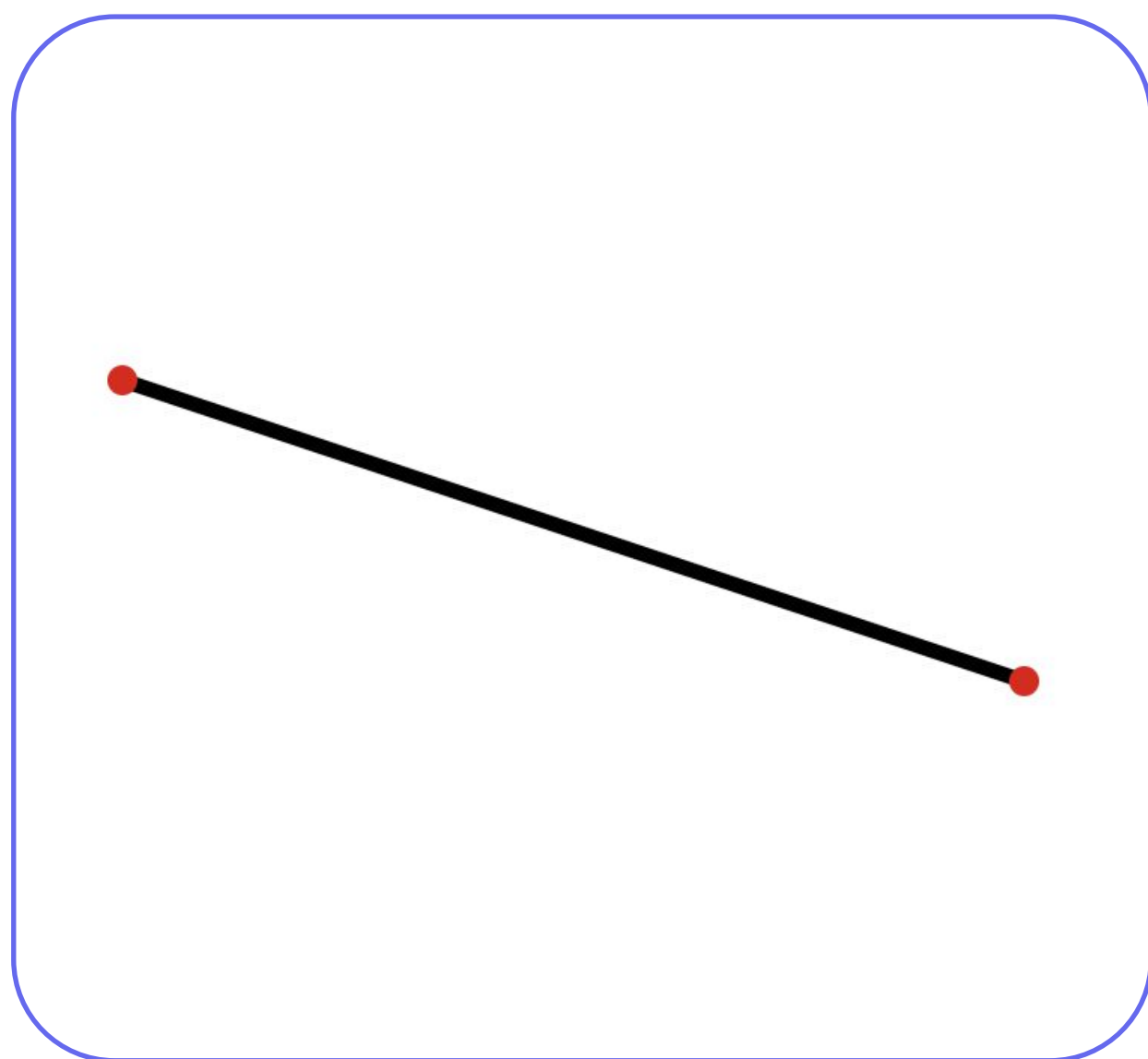
Кривая Безье — это математически описанная кривая, задаваемая **2+** опорными точками

Используется в:

- компьютерной графике
- шрифтах
- анимации



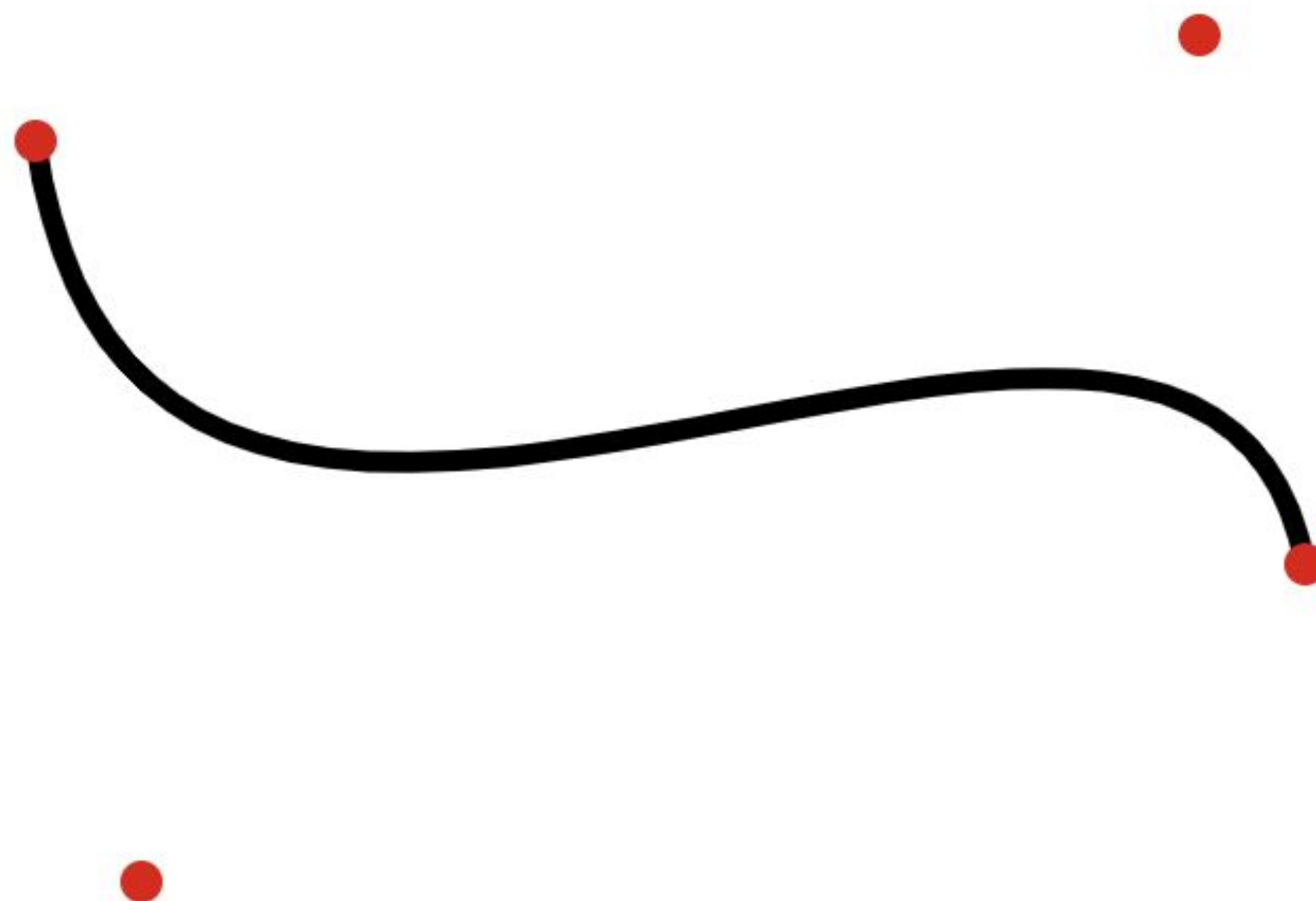
Виды кривых Безье



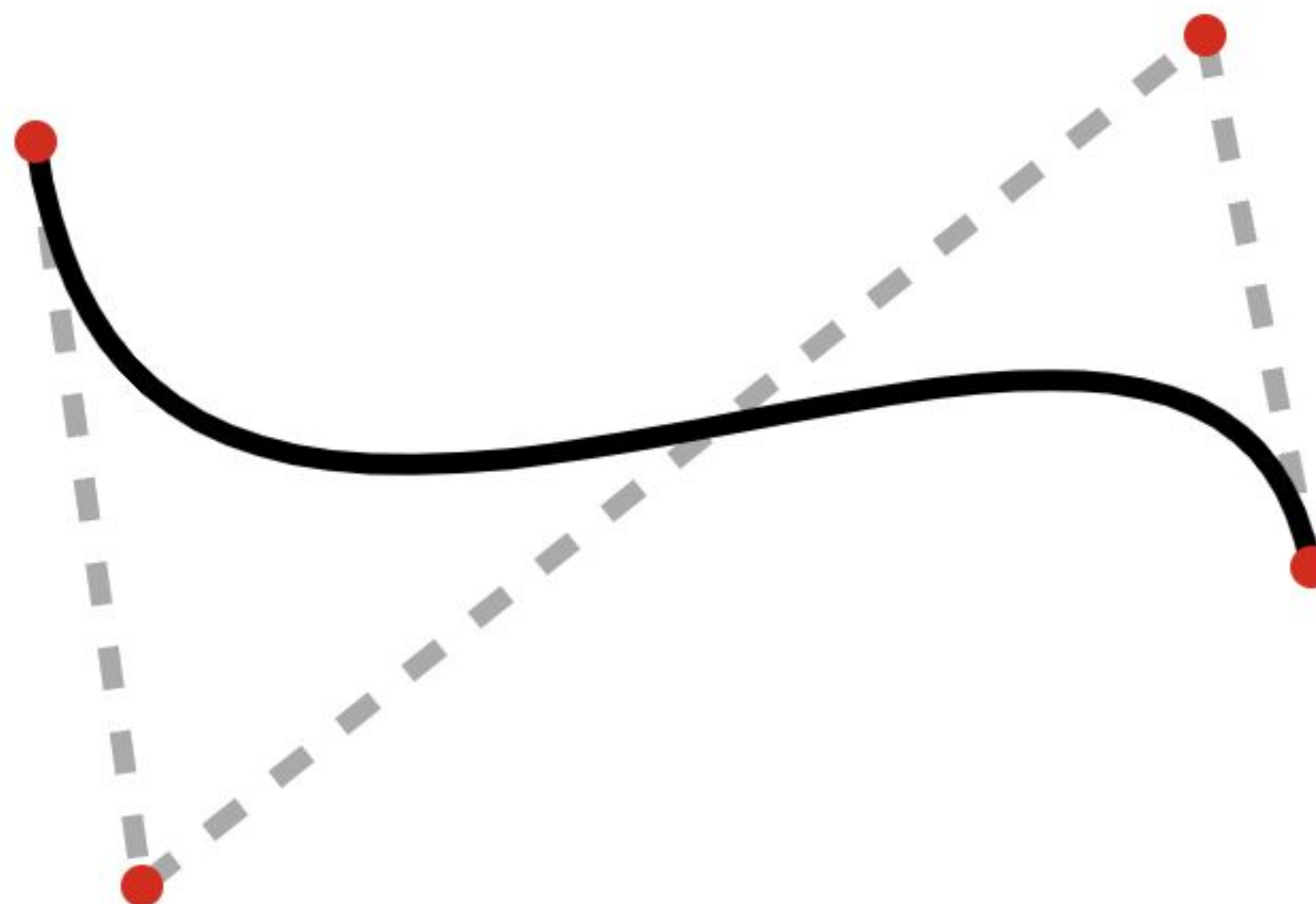
Уравнение кривой Безье

$$\mathbf{B}(t) = (1 - t)^3 \mathbf{P}_0 + 3t(1 - t)^2 \mathbf{P}_1 + 3t^2(1 - t) \mathbf{P}_2 + t^3 \mathbf{P}_3, \quad t \in [0, 1].$$

Алгоритм де Кастельжо

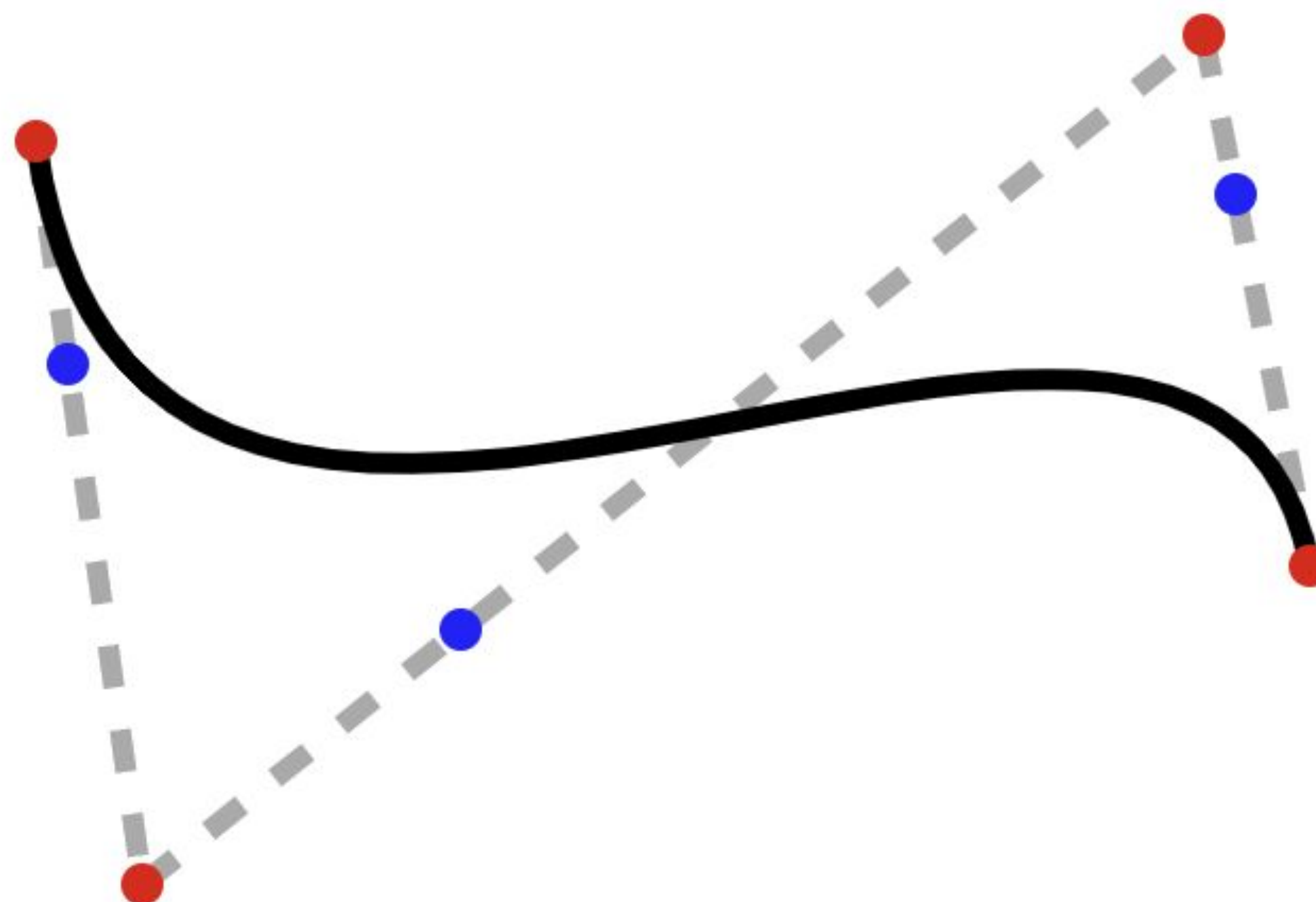


Алгоритм де Кастельжо



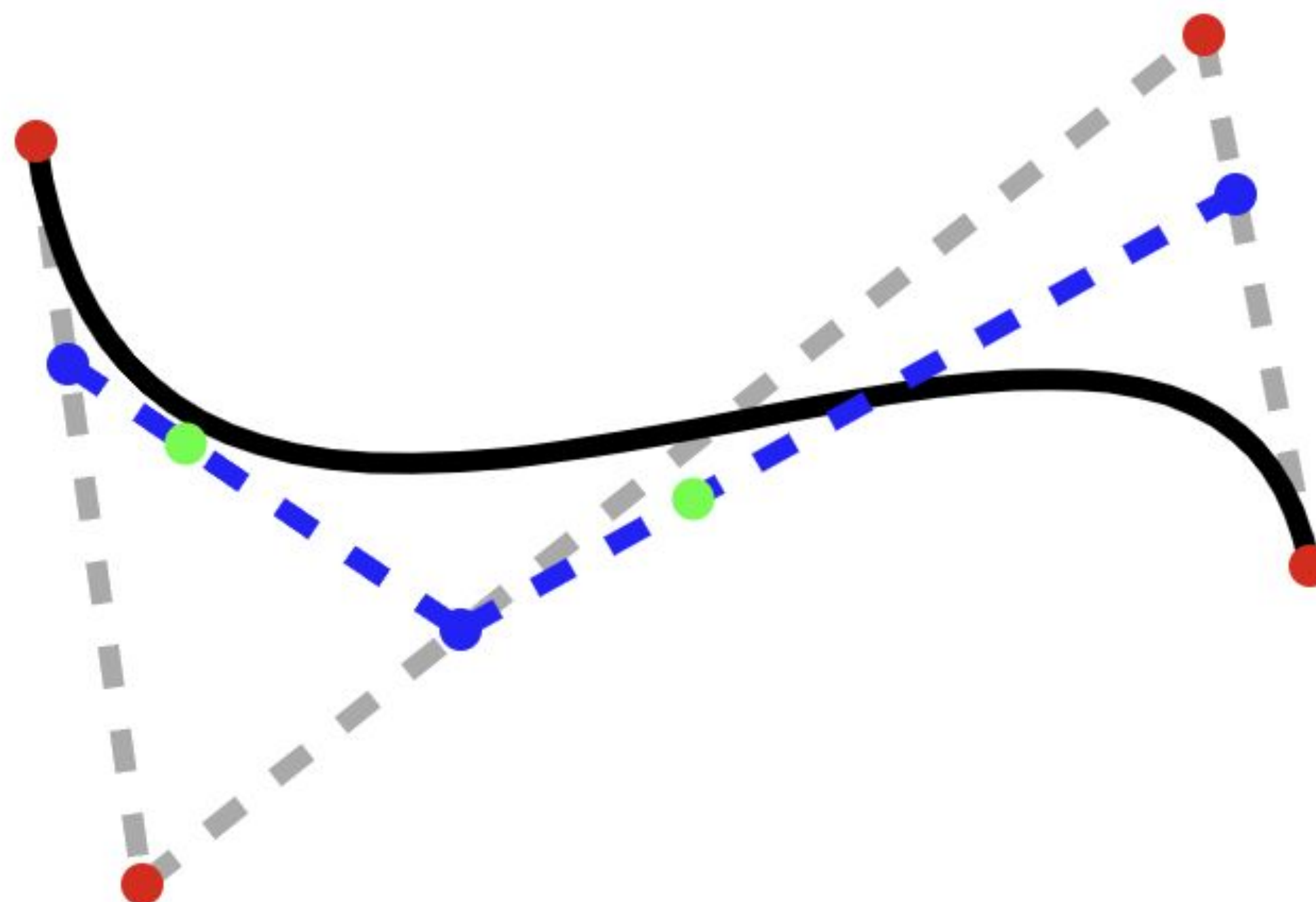
Алгоритм де Кастельжо

$t=0.3$



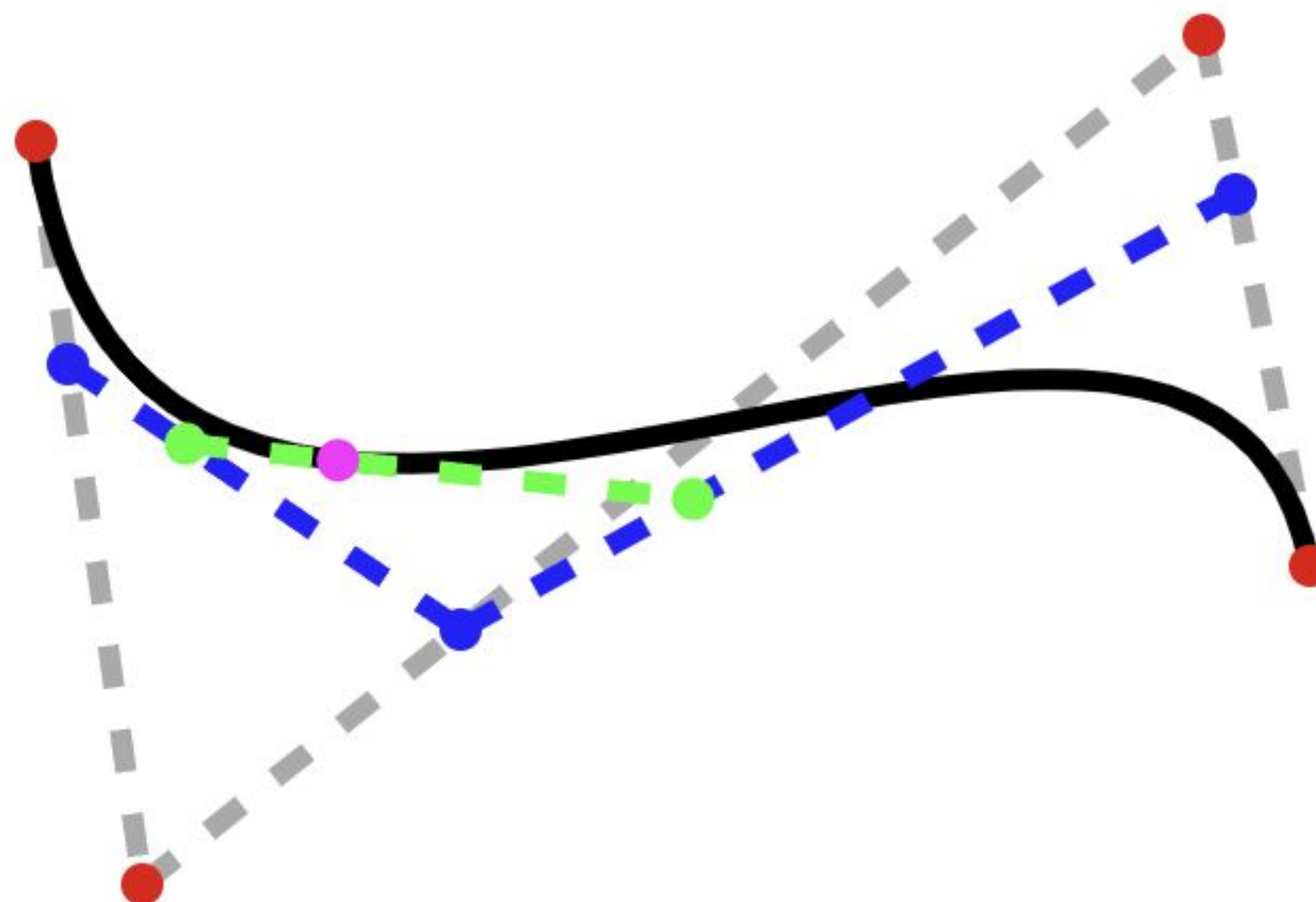
Алгоритм де Кастельжо

$t=0.3$



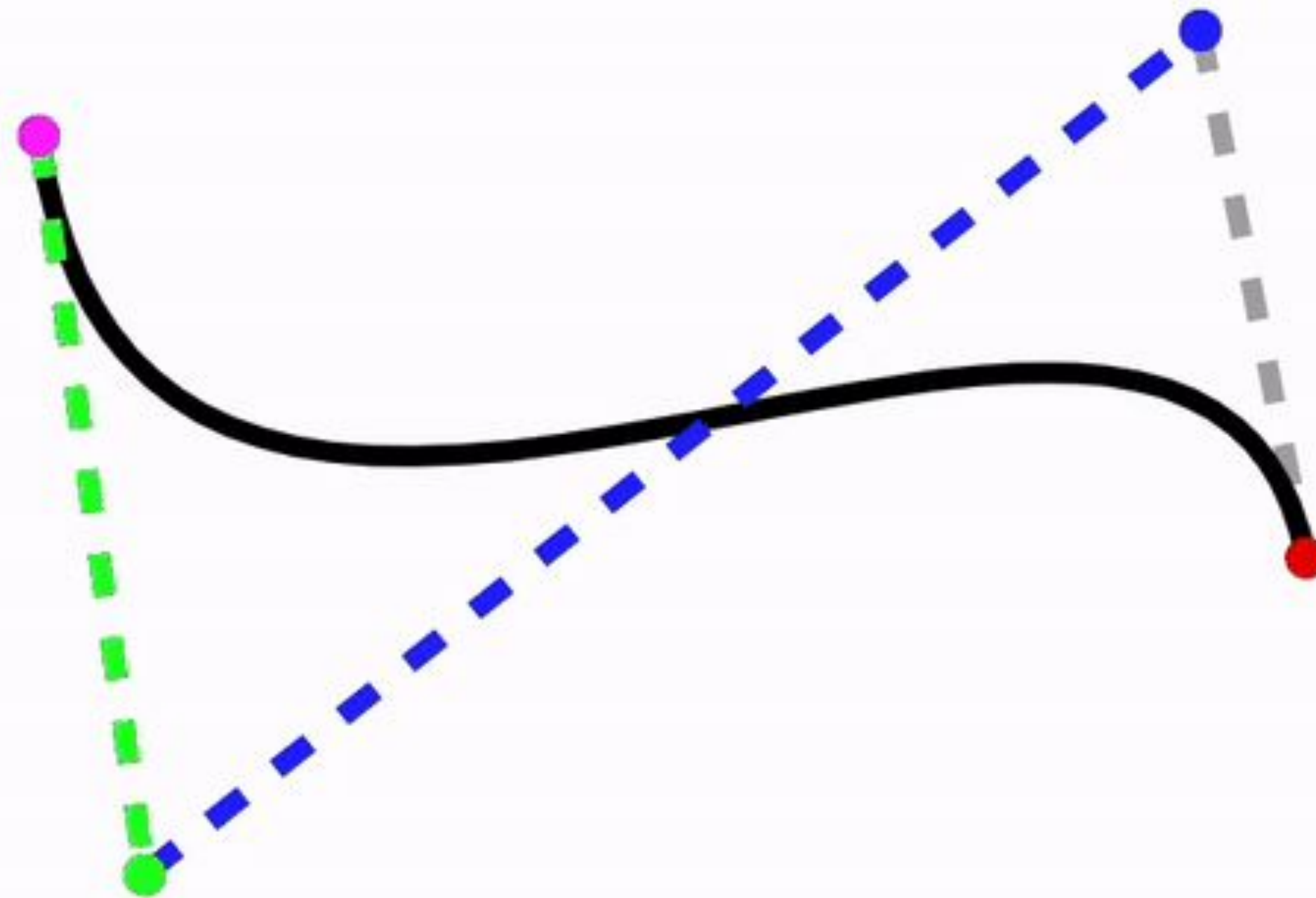
Алгоритм де Кастельжо

$t=0.3$

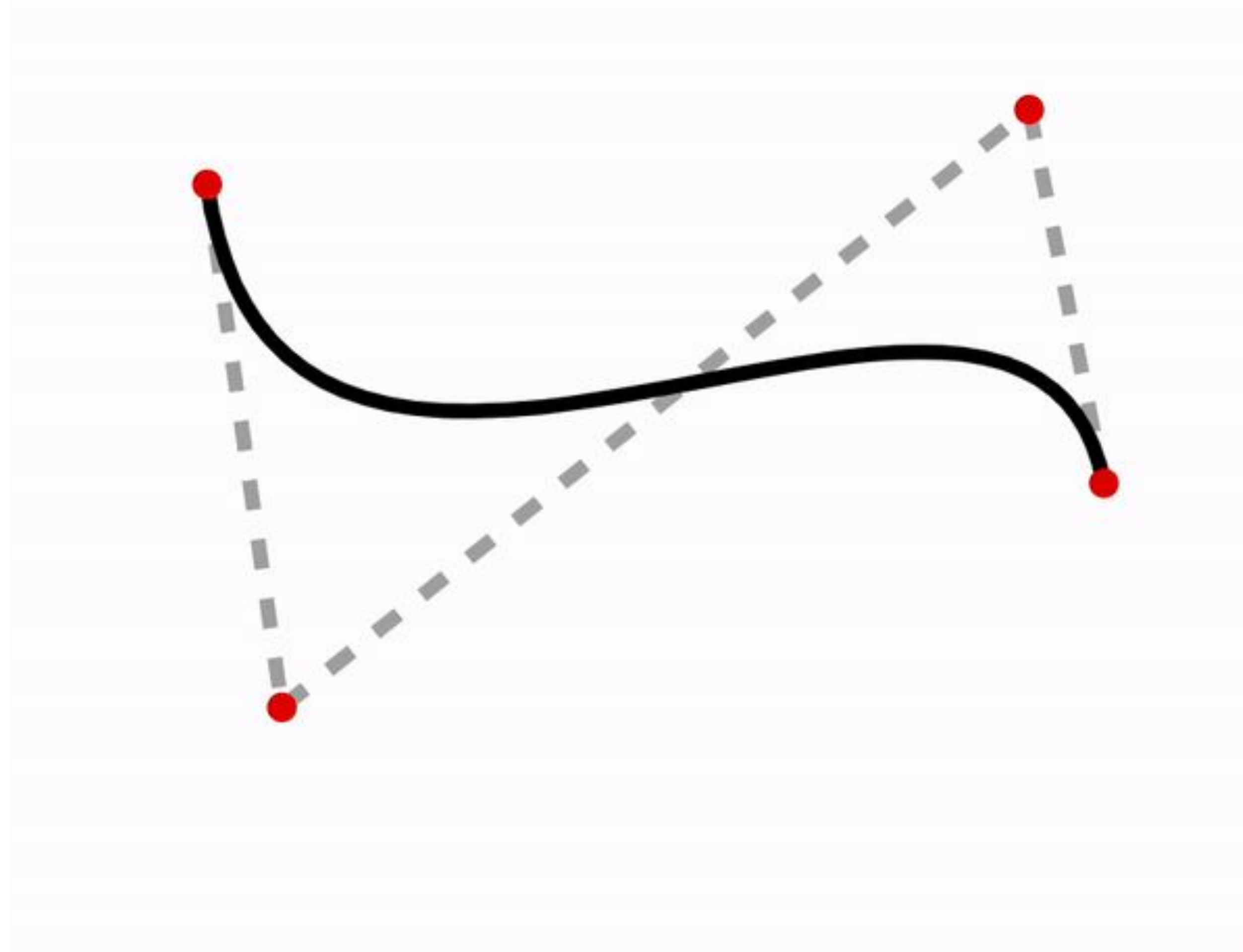


Алгоритм де Кастельжо

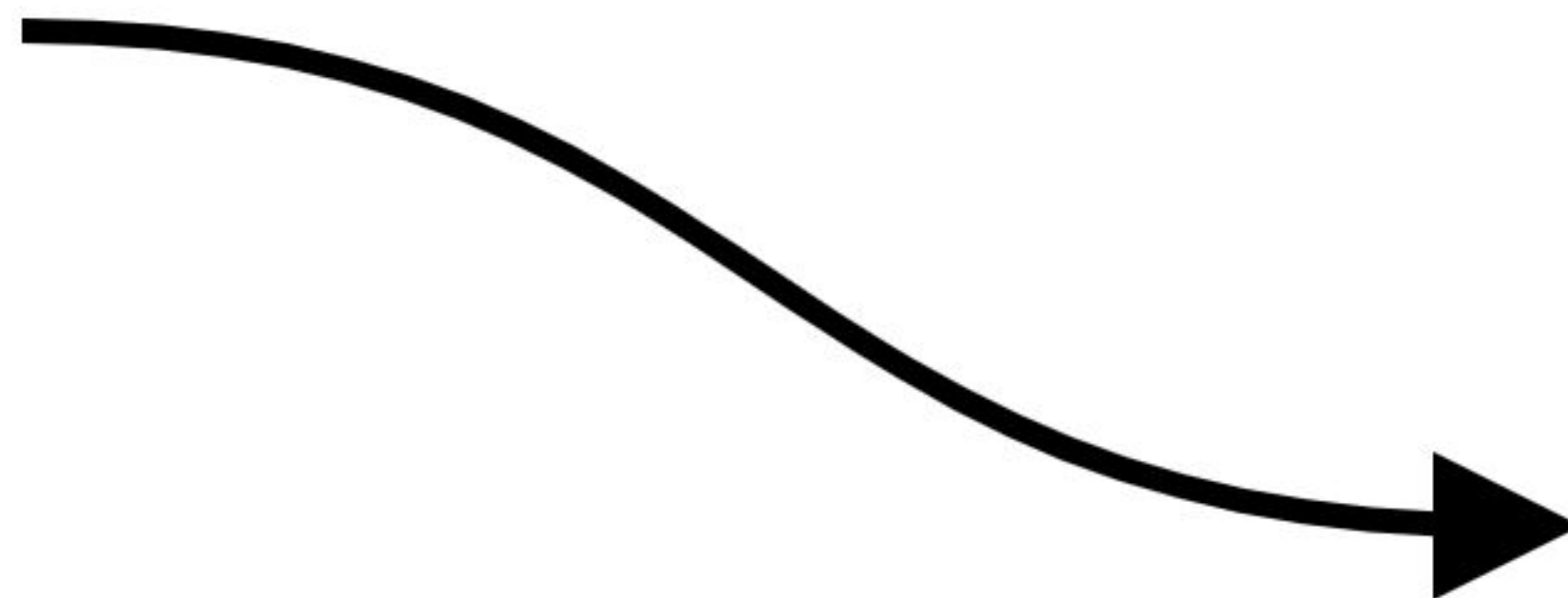
$$t = 0$$



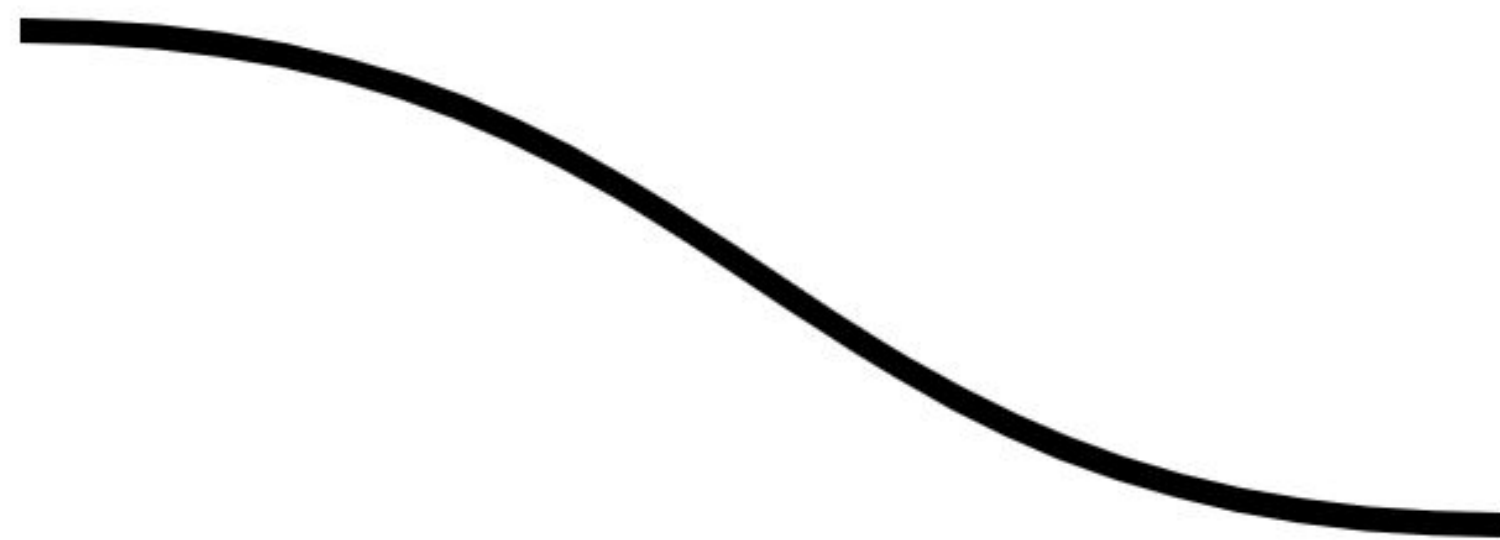
Как кривая Безье меняется
от положения опорных точек?



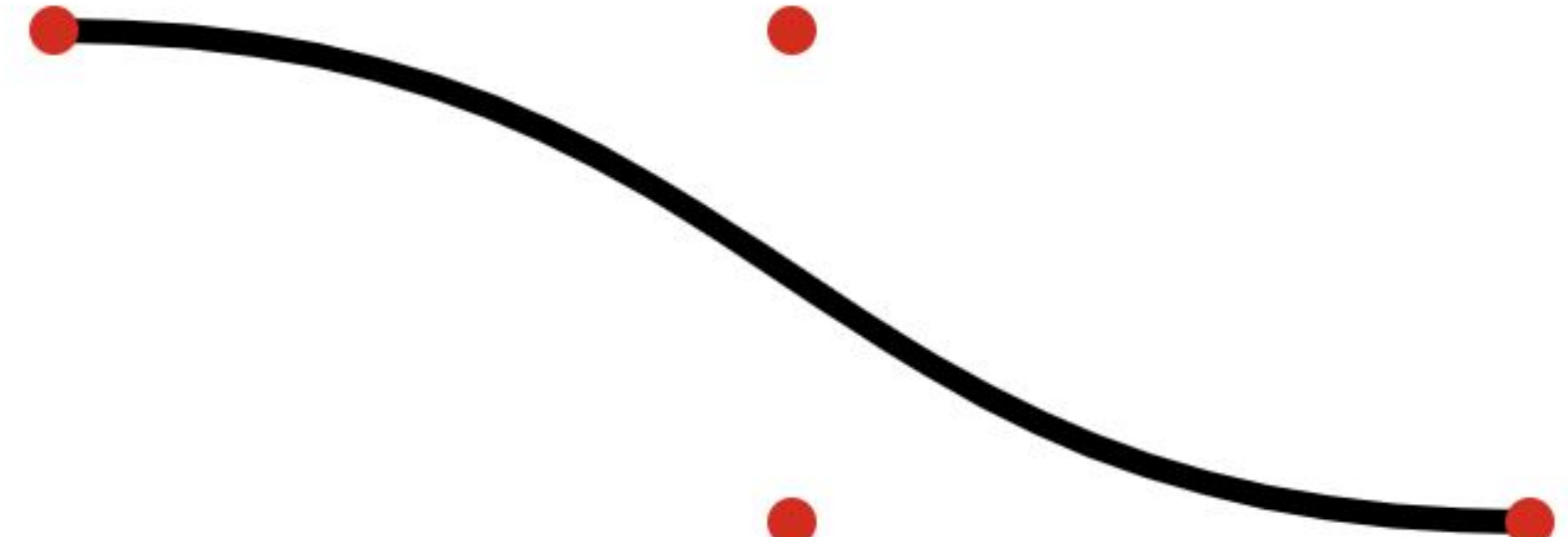
Как строятся стрелки с помощью кривых Безье в SVG



Как строятся стрелки с помощью кривых Безье в SVG



1



2

Как строятся стрелки с помощью кривых Безье в SVG

```
<path stroke="black" fill="none" d="M 10,10 C 40,10 40,30 70,30" />
```

Как строятся стрелки с помощью кривых Безье в SVG

```
<path d="M 0 0 L 10 5 L 0 10 z" />
```



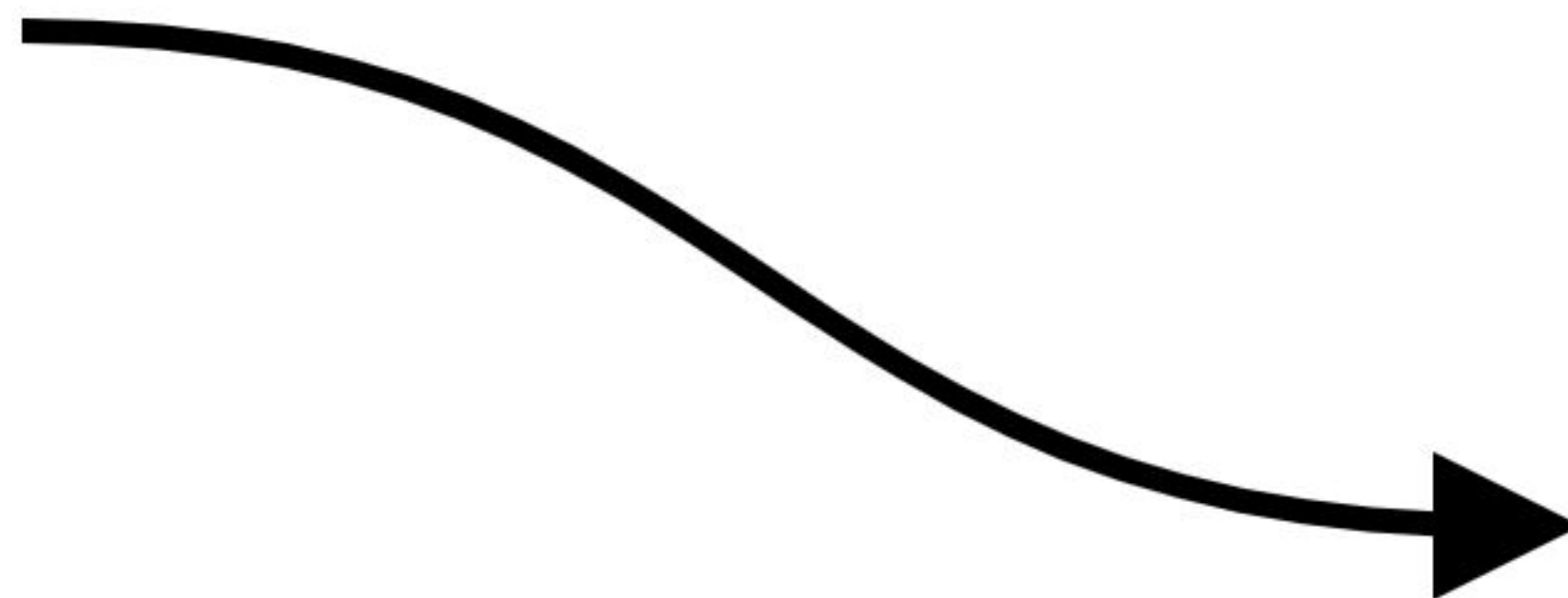
Как строятся стрелки с помощью кривых Безье в SVG

```
<marker
  id="arrow"
  viewBox="0 0 10 10"
  refX="5"
  refY="5"
  markerWidth="6"
  markerHeight="6"
  orient="auto"
>
  <path d="M 0 0 L 10 5 L 0 10 z" />
</marker>
```

Как строятся стрелки с помощью кривых Безье в SVG

```
<path stroke="black" fill="none" d="M 10,10 C 40,10 40,30 70,30"  
      marker-end="url(#arrow)" />
```

Как строятся стрелки с помощью кривых Безье в SVG

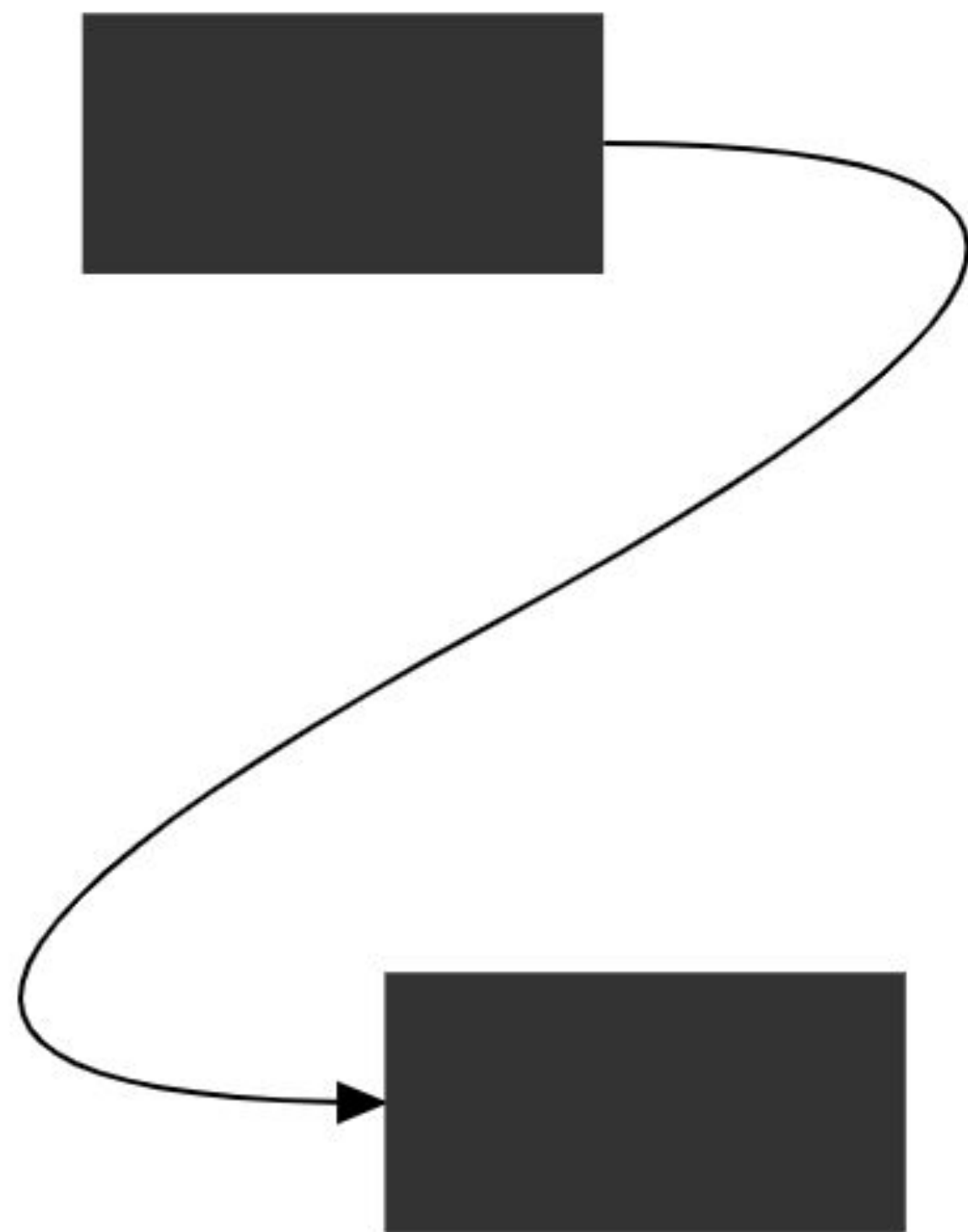


Стрелки почти готовы

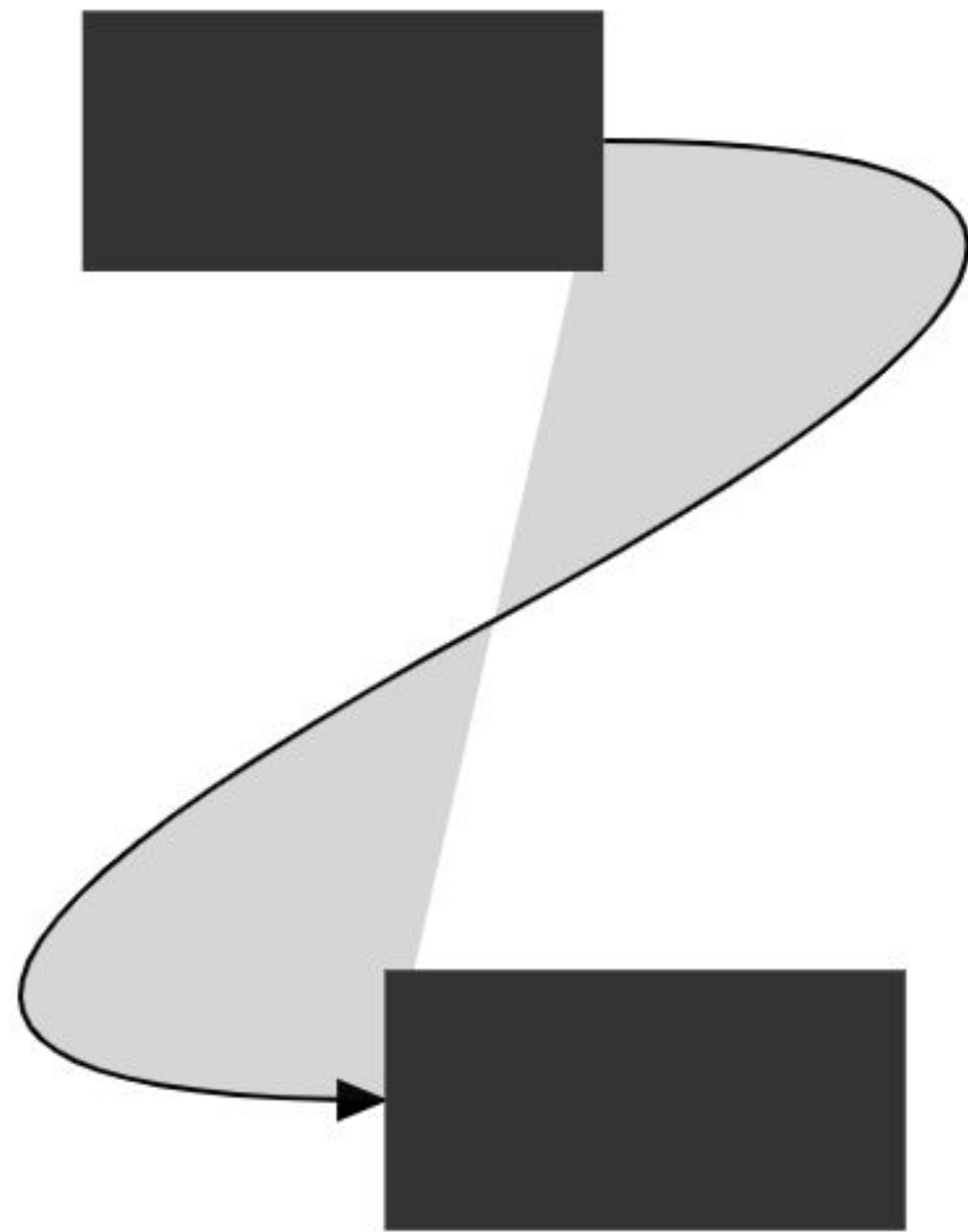
Улучшение удобства стрелок



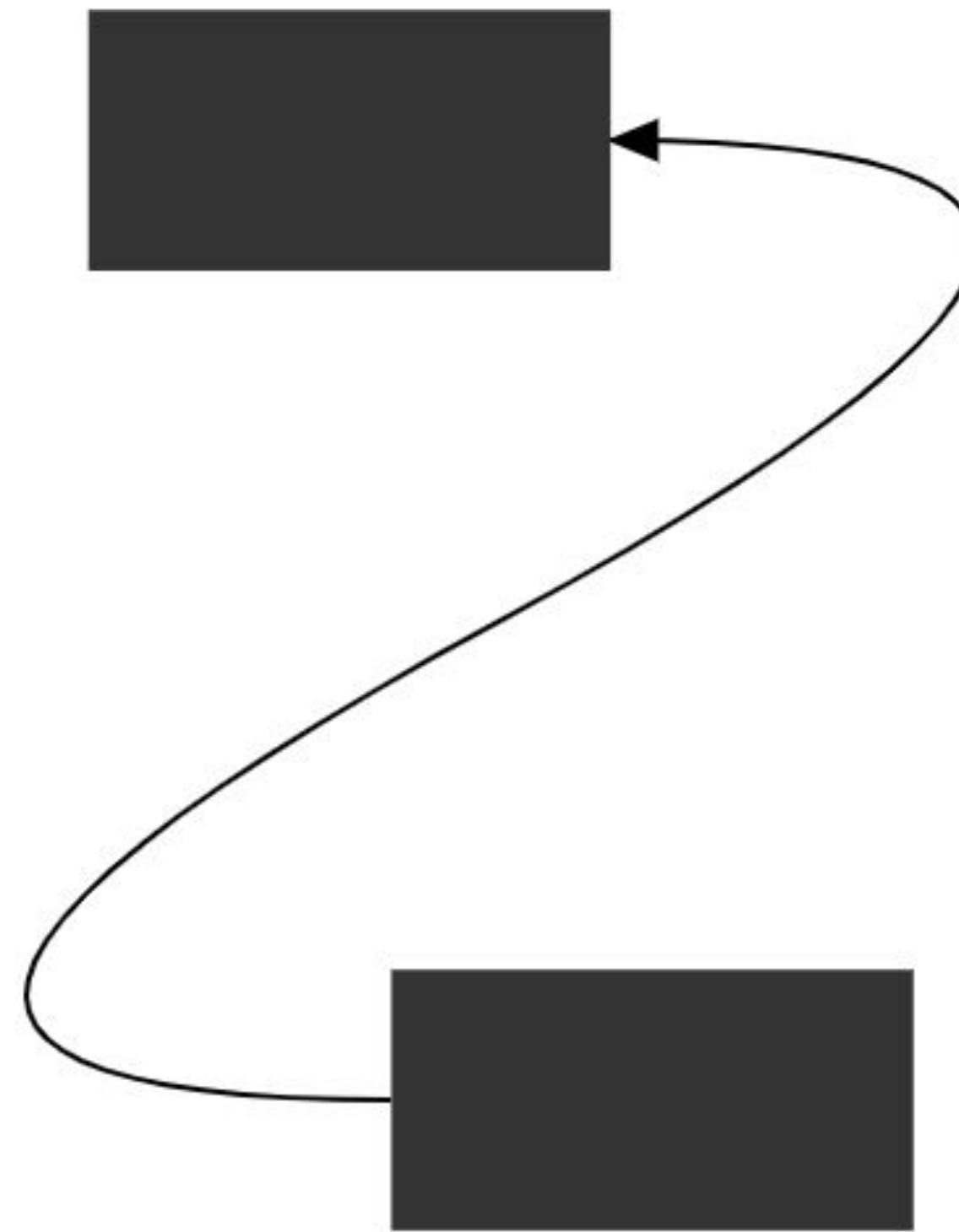
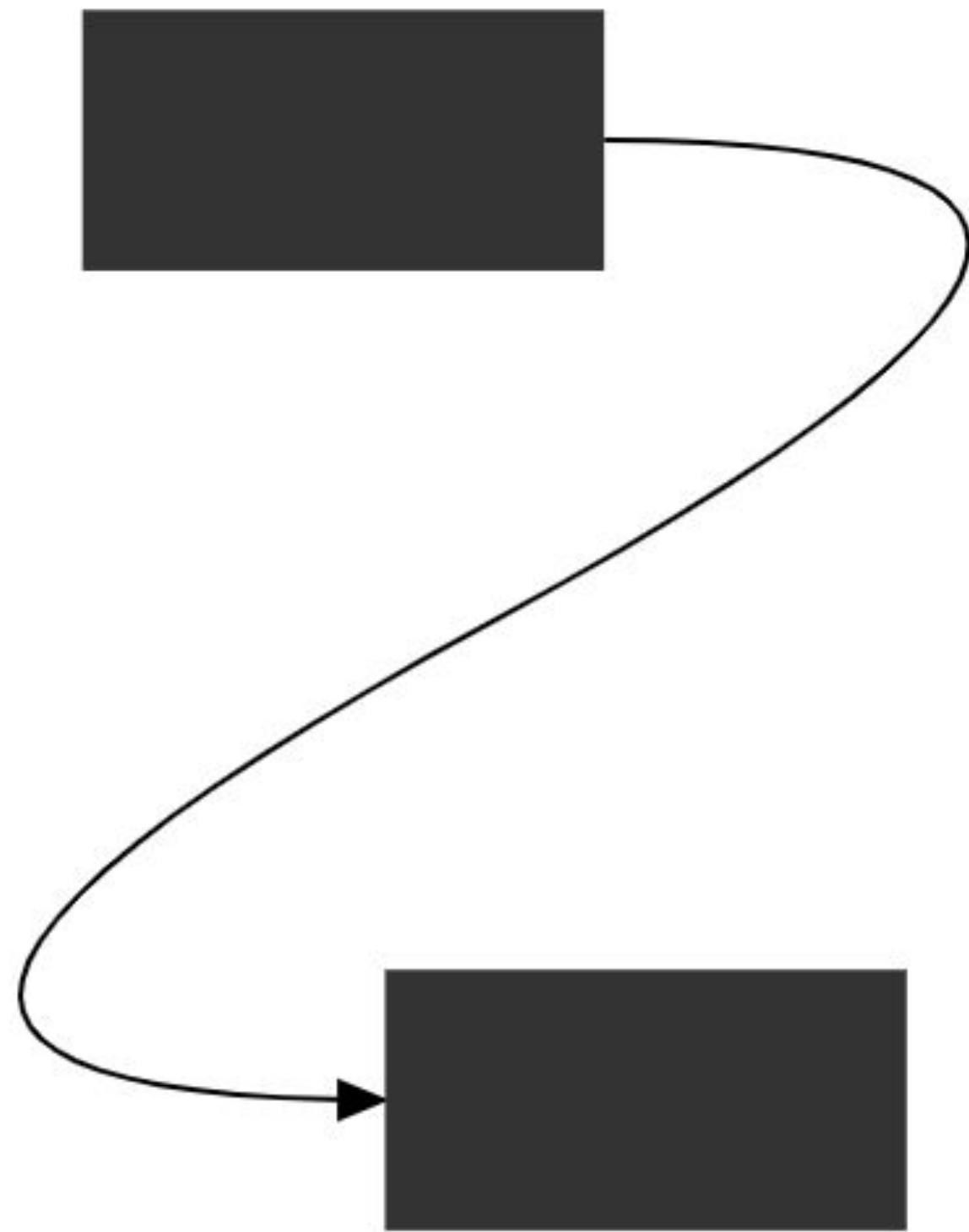
Проблема незакрытого пути стрелки и ее решение



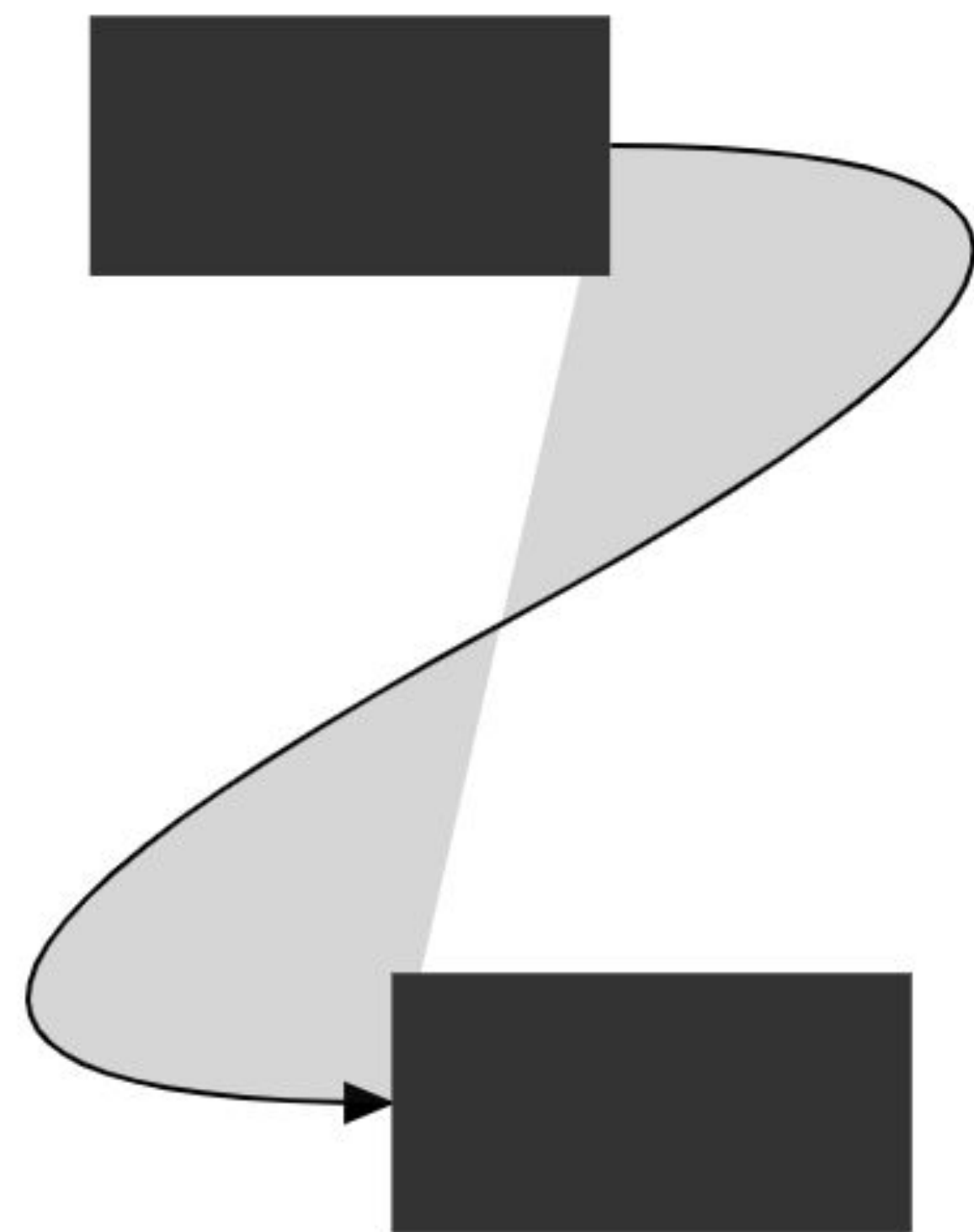
Проблема незакрытого пути стрелки и ее решение



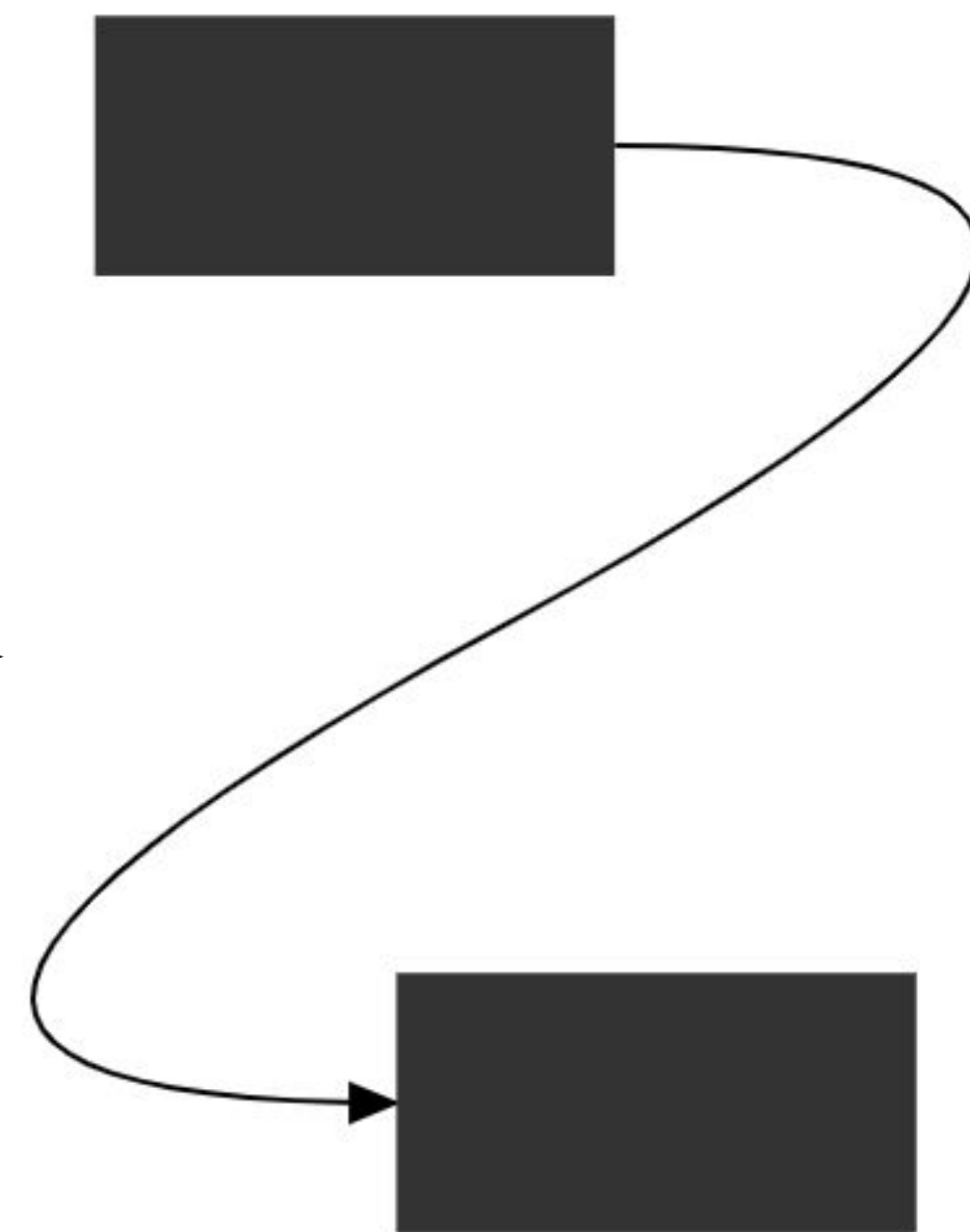
Проблема незакрытого пути стрелки и ее решение





Более качественное решение



`pointer-events: stroke;`



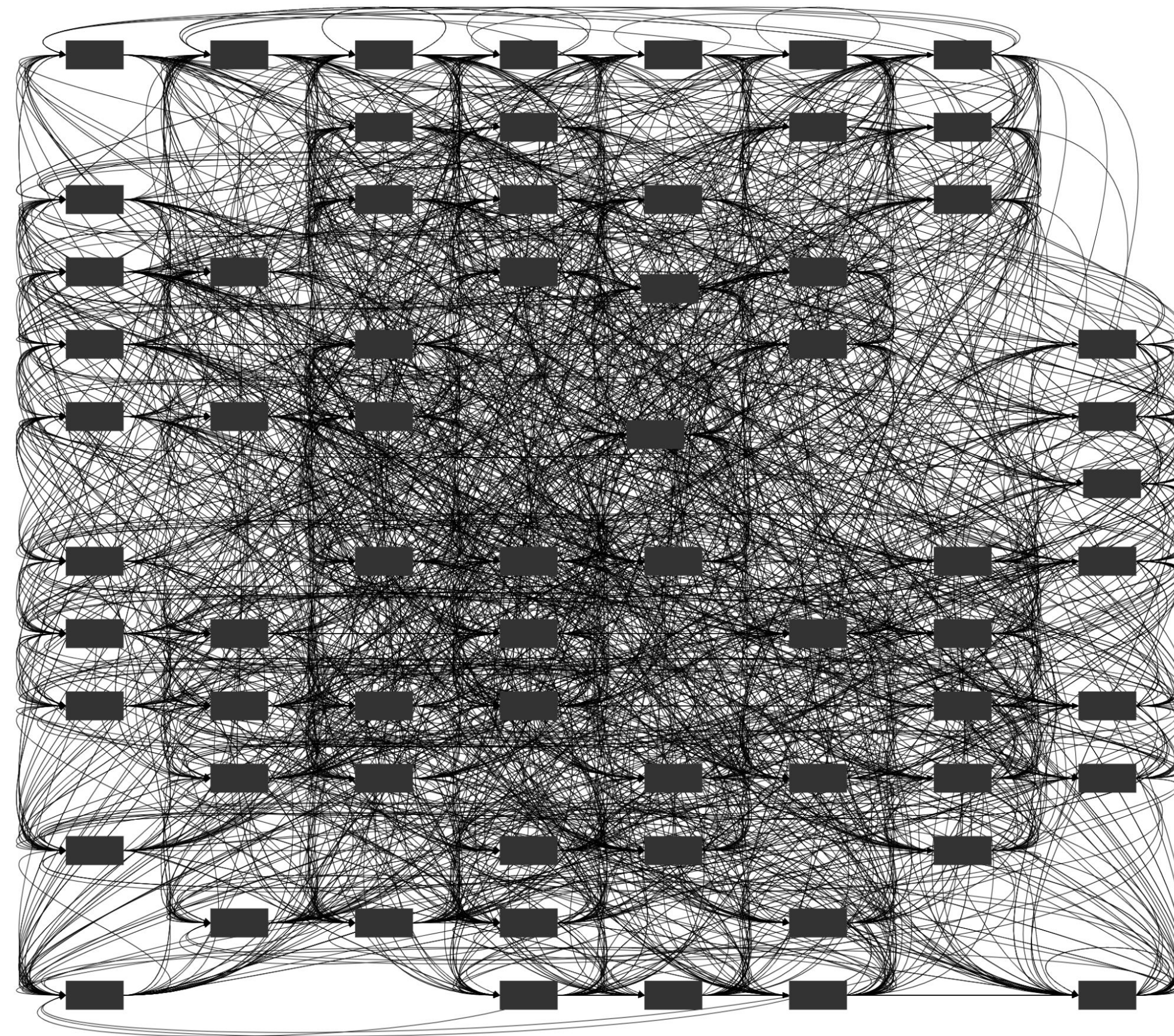
Поддержка в браузерах

												
	 Chrome	 Edge	 Firefox	 Opera	 Safari	 Chrome Android	 Firefox for Android	 Opera Android	 Safari on iOS	 Samsung Internet	 WebView Android	 WebView on iOS
<code>pointer-events</code>	✓ 1	✓ 12	✓ 1.5	✓ 9	✓ 4	✓ 18	✓ 4	✓ 14	✓ 3.2	✓ 1.0	✓ 2	✓ 3.2
Applies to HTML elements	✓ 2	✓ 12	✓ 3.6	✓ 15	✓ 4	✓ 18	✓ 4	✓ 14	✓ 3.2	✓ 1.0	✓ 37	✓ 3.2

Оптимизации

Оптимизация рендеринга стрелок

- ~70 блоков
- 2000 стрелок



Оптимизация рендеринга стрелок

5 FPS

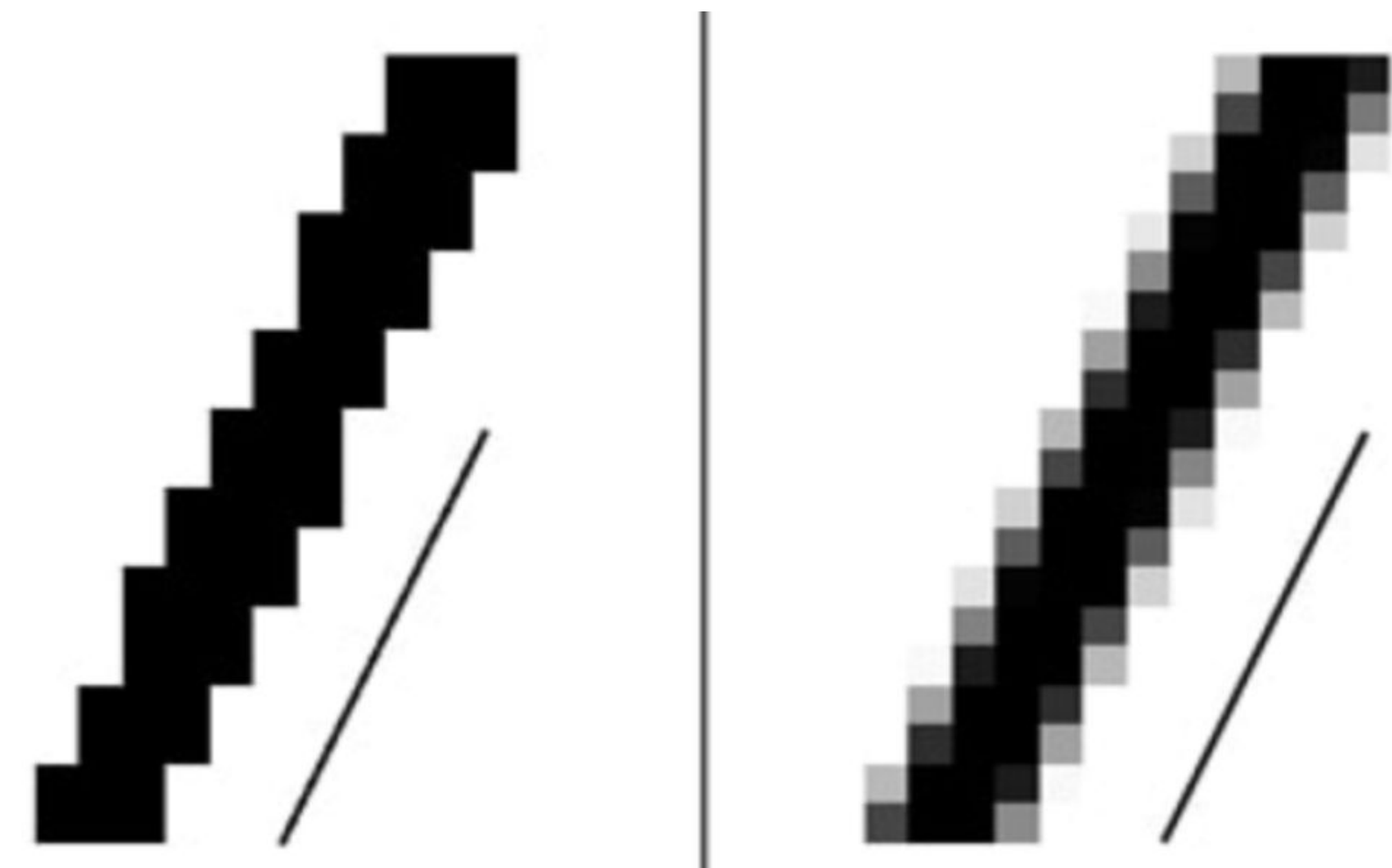
Оптимизация рендеринга стрелок

5_{FPS} shape-rendering: optimizeSpeed; → 113_{FPS}

Что делает optimizeSpeed?















Согласно спецификации SVGWG...

Indicates that the user agent shall emphasize rendering speed over geometric precision and crisp edges. This option will sometimes cause the user agent to turn off shape anti-aliasing.



Anti-aliasing

Поддержка в браузерах

												
	 Chrome	 Edge	 Firefox	 Opera	 Safari	 Chrome Android	 Firefox for Android	 Opera Android	 Safari on iOS	 Samsung Internet	 WebView Android	 WebView on iOS
shape-rendering	✓ 80	✓ 80	✓ 72	✓ 67	✓ 13.1	✓ 80	✓ 79	✓ 57	✓ 13.4	✓ 13.0	✓ 80	✓ 13.4

Возвращаем качество, когда ничего не происходит

```
const onSomeAction = () => {  
  activateOptimizeSpeed();  
  ...  
}
```

```
const onSomeActionEnd = () => {  
  deactivateOptimizeSpeed();  
  ...  
}
```

Возвращаем качество, когда ничего не происходит

```
const optimizeGraphics = () => {  
  activateOptimizeSpeed();  
  debounce(() => deactivateOptimizeSpeed(), 400);  
}
```

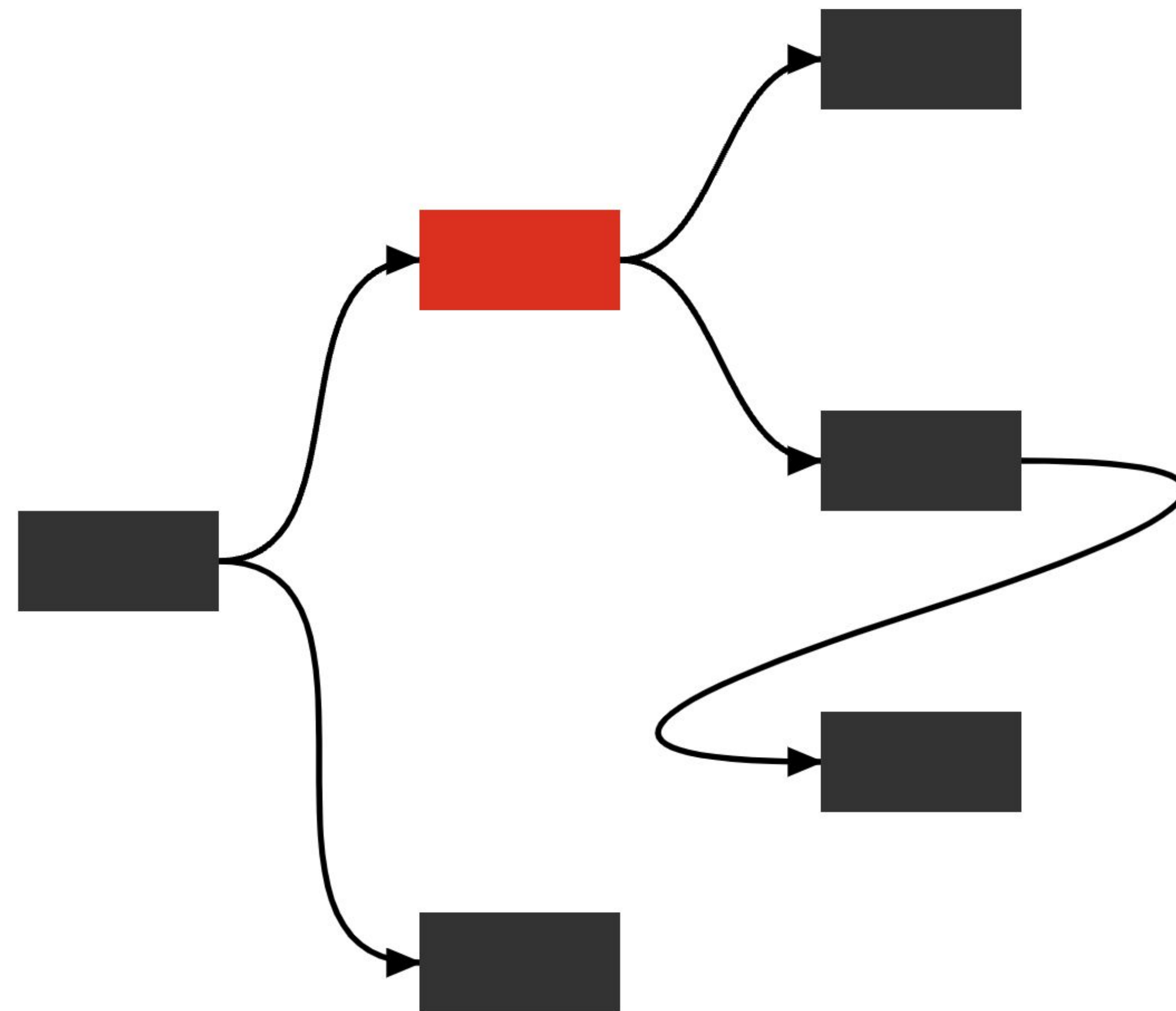
...

```
const someAction = () => {  
  optimizeGraphics();
```

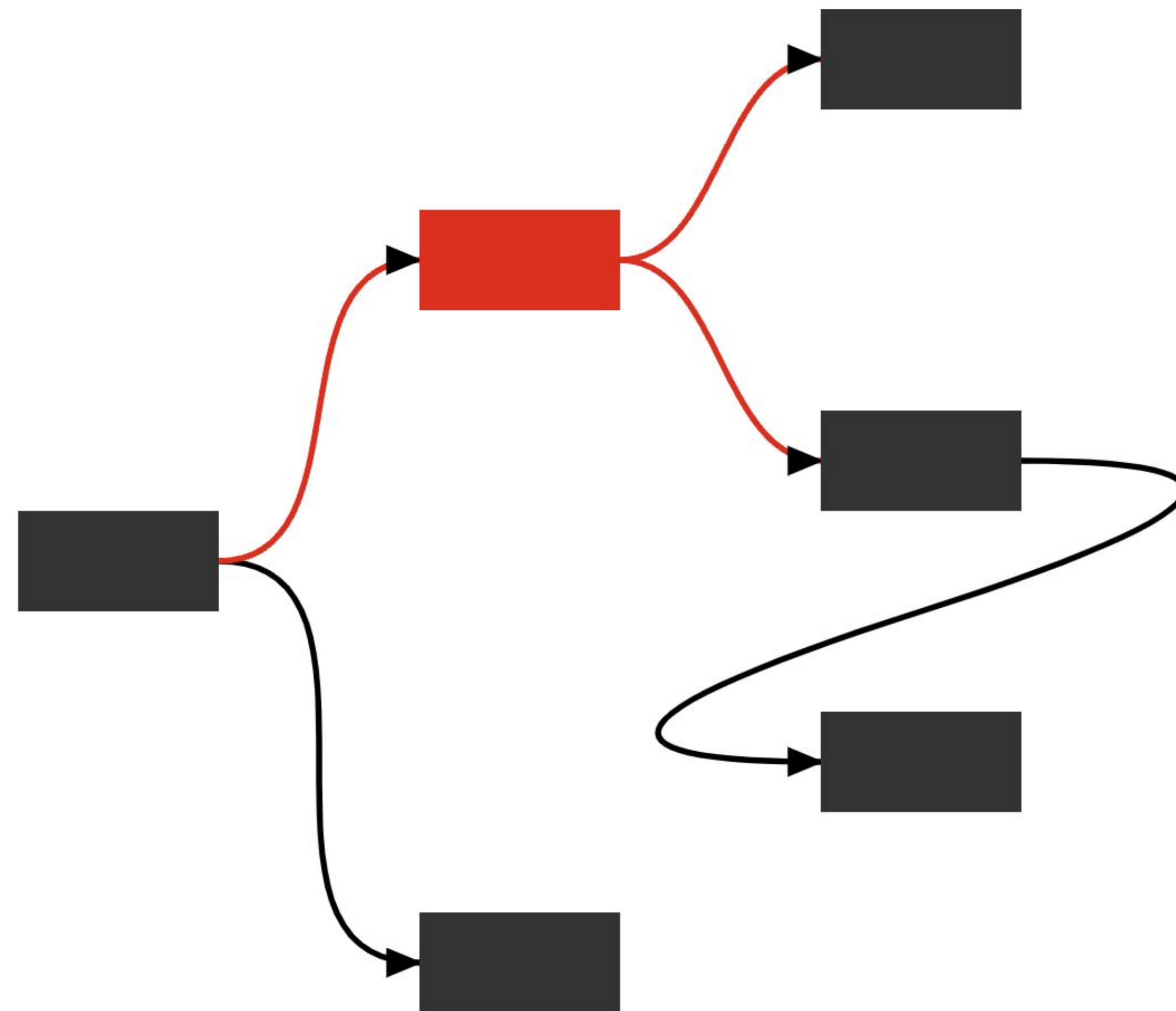
...

```
}
```

Оптимизация нагрузки на CPU



Оптимизация нагрузки на CPU



Оптимизация нагрузки на CPU

```
const targetsWeakMap = new WeakMap<HTMLElement, Set<() => void>>();
```

Оптимизация нагрузки на CPU

```
const targetsWeakMap = new WeakMap<HTMLElement, Set<() => void>>();
```

```
...
```

```
const update = (target) => {  
  if (target && targetsWeakMap.current.get(target)) {  
    targetsWeakMap.current.get(target)?.forEach((handler) => handler());  
  } else {  
    updateAll();  
  }  
}
```


Оптимизация нагрузки на CPU

```
const targetsWeakMap = new WeakMap<HTMLElement, Set<() => void>>();
```

```
...
```

```
const update = (target) => {  
  if (target && targetsWeakMap.current.get(target)) {  
    targetsWeakMap.current.get(target)?.forEach((handler) => handler());  
  } else {  
    updateAll();  
  }  
}
```

```
...
```

```
const handleUpdate: DraggableEventHandler = (mouseEvent, dragEvent) => {  
  update(dragEvent.node);  
};
```

ИТОГИ

Библиотека для рендеринга SVG-стрелок со встроенными оптимизациями и без зависимостей

ИТОГИ

Библиотека для рендеринга SVG-стрелок со встроенными оптимизациями и без зависимостей

+Bonus: встроенная возможность кастомизации стрелок

ИТОГИ

Библиотека для рендеринга SVG-стрелок со встроенными оптимизациями и без зависимостей

+Bonus: встроенная возможность кастомизации стрелок

Согласно bundlephobia.com

Minified: **322 B**

Minified + Gzipped: **207 B**



GitHub



@asmalcev_dev

baana-react