

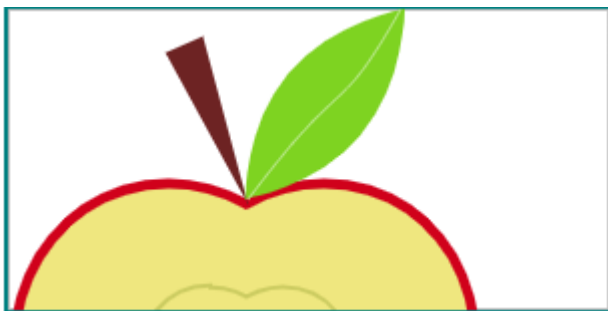
Ширина и высота SVG

SVG ведёт себя иначе, чем привычные HTML-элементы: его содержимое отрисовывается на бесконечном холсте, и его размеры не зависят от содержимого. Видимая часть холста соответствует размерам SVG-элемента, эта область отрисовки называется **вьюпорт**.

При этом можно управлять как размерами SVG-элемента, так и поведением его содержимого: оно может отображаться целиком, обрезаться или сжиматься не сохраняя пропорции.

Если SVG просто вставить на страницу не указывая размеры, он отобразится размером 300×150 пикселей:

```
<svg>
...
</svg>
```



Поменять ширину и высоту можно с помощью `width` и `height`:

```
<svg width="350" height="200">
...
</svg>
```



Задавать размеры можно как атрибутами, так и в CSS:

```
svg {  
  
  width: 350px;  
  
  height: 200px;  
  
}
```

Для размеров в CSS обязательно указывать единицы измерения. Для размеров в атрибутах, задаваемых в пикселях, единицы измерения не нужны.

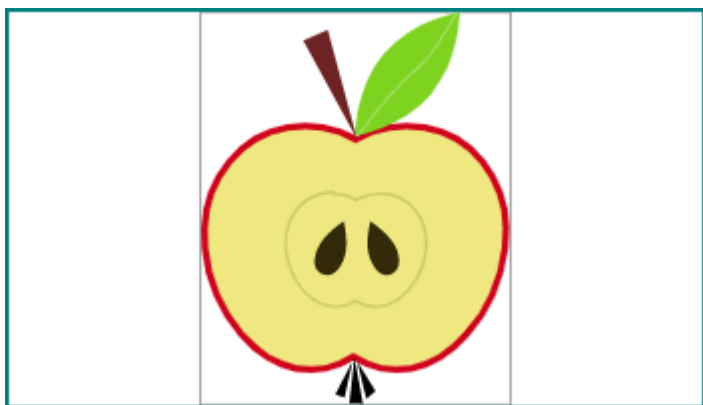
Атрибут viewBox

Вы наверняка заметили, что изменение размеров SVG-элемента не влияет на его содержимое — потому что содержимое отрисовывается на бесконечном холсте, и непонятно какого размера область нужно растягивать или сжимать.

Это поведение можно изменить, задав размер области, которая будет тянуться, с помощью свойства `viewBox` (его можно задать только атрибутом):

```
<svg viewBox="0 0 237 300" width="350" height="200">  
  
  ...  
  
</svg>
```

Первые два числа — координаты X и Y верхнего левого угла масштабируемой области, два других — её ширина и высота. Значения задаются в пикселях, единицы измерения указывать не нужно.



С `viewBox` содержимое масштабируется, чтобы поместиться целиком в контейнер, и выравнивается по центру.

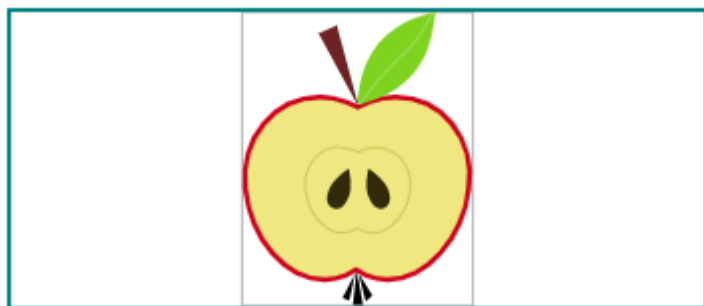
viewBox и размеры

Как мы увидели в прошлом задании, SVG без размеров, но с `viewBox`, пытается занять всё доступное пространство. Это означает, что если на странице есть инлайновые иконки, размеры которым задаются в CSS, без CSS могут растянуться на весь экран.

Чтобы этого избежать, достаточно всем инлайновым иконкам в атрибутах явно задавать размеры по умолчанию, они потом легко переопределяются в CSS.

Атрибут `preserveAspectRatio`

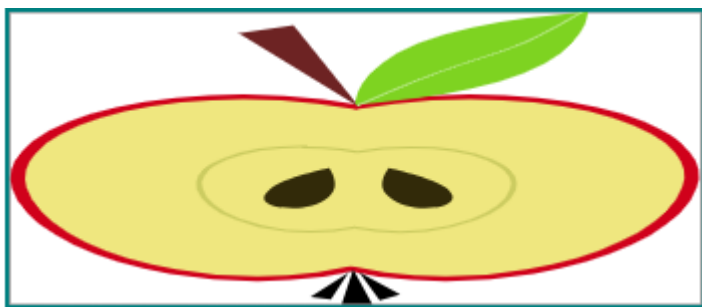
По умолчанию содержимое SVG с `viewBox` масштабируется сохраняя пропорции, и если соотношения сторон вьюпорта и вьюбокса не совпадают, вокруг содержимого появляются поля:



С помощью свойства `preserveAspectRatio` это поведение можно изменять: например, значение `none` указывает, что сохранять пропорции не нужно:

```
<svg viewBox="0 0 237 300" preserveAspectRatio="none">
...
</svg>
```

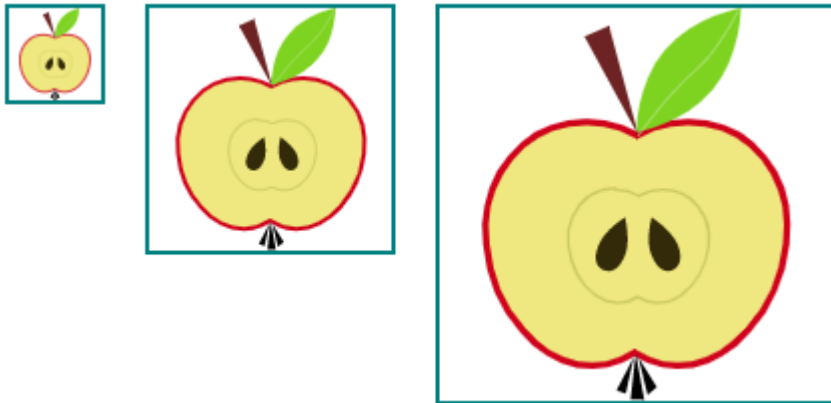
В этом случае область, размеры которой заданы вьюбоксом, растягивается на всё доступное пространство вьюпорт



`preserveAspectRatio` задаётся только атрибутом.

Резиновый фон с `preserveAspectRatio`

SVG, заданный в качестве фона, ведёт себя так же, как инлайновый SVG, поэтому, чтобы получить резиновый фон, используйте SVG с `viewBox`, но без размеров: в этом случае изображение подгонится под размер элемента, которому задан фон, и будет тянуться вместе с ним, сохраняя пропорции:



Это очень удобно для иконок: задайте размеры родительскому элементу, и иконка, заданная фоном, сама под него растянется.

Если же нужно, чтобы пропорции не сохранялись, добавьте `preserveAspectRatio="none"`. Это пригодится для резиновых фонов:



Выравнивание в `preserveAspectRatio`

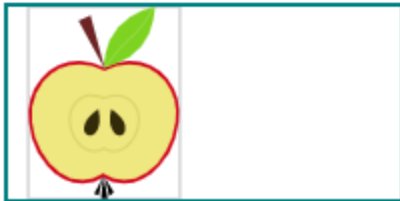
Содержимое SVG можно не только растягивать, но и сдвигать вправо-влево или вверх-вниз. Для этого нужно указать положение содержимого относительно осей X и Y, например `xMinYMid`:

```
<svg viewBox="0 0 237 300" preserveAspectRatio="xMinYMid">
  ...
</svg>
```

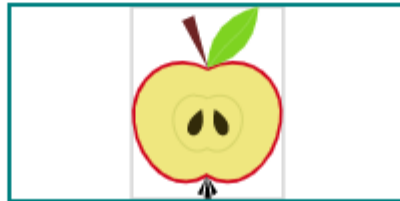


Возможные значения для каждой оси:

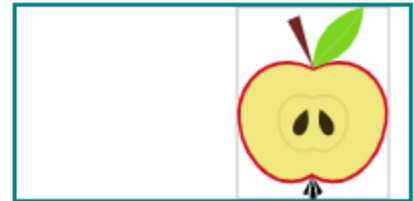
xMin



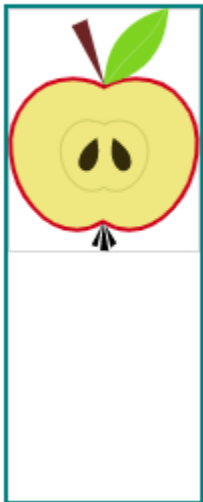
xMid



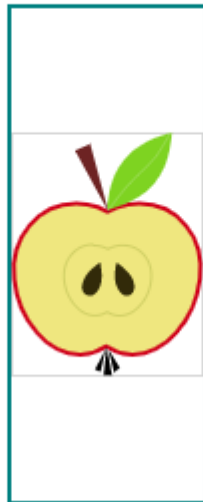
xMax



yMin



yMid



yMax



Положение задаётся двумя параметрами: первым всегда указывается положение по X, вторым по Y. Положение по оси Y всегда пишется с большой буквы. Оба параметра обязательны.

Значение по умолчанию — `xMidyMid` (содержимое выравнивается по середине большей стороны).

`preserveAspectRatio` и `viewBox`

Нужно помнить, что `preserveAspectRatio` не работает без `viewBox`. `viewBox` определяет масштабируемую область, `preserveAspectRatio` — как эта область выравнивается и как заполняет собой вьюпорт.

Также `preserveAspectRatio` не работает, если содержимое отрисовывается без полей (то есть соотношения сторон вьюпорта и вьюбокса совпадают), тогда в нём просто нет необходимости.

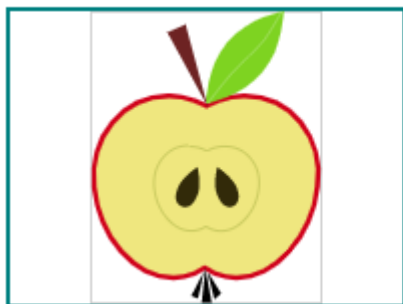
Заполнение пространства

Второй параметр в свойстве `preserveAspectRatio` задаёт поведение содержимого относительно вьюпорта, определяет как именно содержимое заполняет пространство:

```
<svg viewBox="0 0 237 300" preserveAspectRatio="xMinYMin meet">
  ...
</svg>
```

Возможные значения:

meet



slice



`meet` — содержимое уместается целиком, оставляя пустые поля (как при `background-size: contain`). Значение по умолчанию.

`slice` — содержимое заполняет собой всё пространство, при этом часть содержимого может быть обрезана (похоже на `background-size: cover`). Пропорции сохраняются в обоих случаях.

Заполнение — необязательный параметр, его можно не задавать.

Единицы измерения

Для базового использования SVG достаточно представлять как работают внешние размеры, но для создания более сложных конструкций нужно понимать как работают внутренние.

В SVG можно использовать разные единицы измерения, например: `px`, `em`, `ex`, `pt`, `pc`, `cm`, `mm`, `in` и проценты.

Также есть единицы системы координат — *user space units*, которые по умолчанию соответствуют пикселям, поэтому для размеров и координат в пикселях единицы измерения можно не указывать.

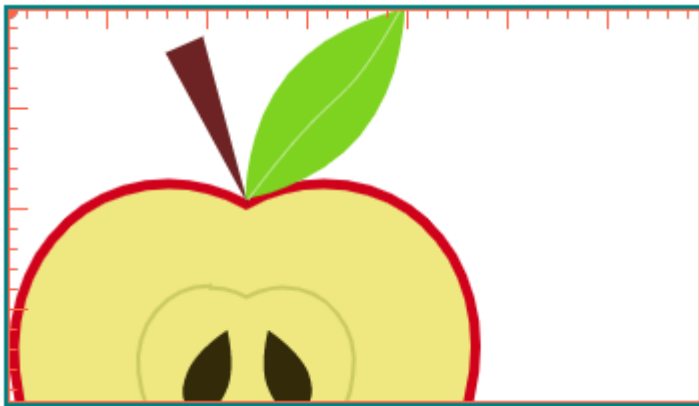
Системы координат

В SVG существует две системы координат:

1. Система координат вьюпорта — **viewport space**
2. Система координат содержимого — **user space**

Изначально системы и их единицы измерения соответствуют друг другу:

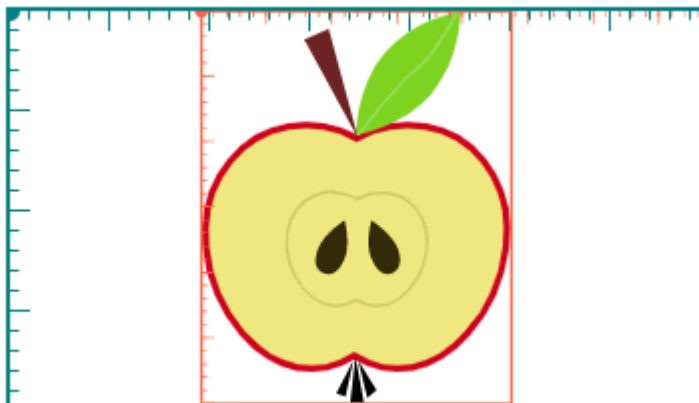
```
<svg width="350" height="200">  
...  
</svg>
```



Сейчас видно только систему координат содержимого (она показана красным), потому что системы совпадают и одна скрыта под другой.

Если добавить вьюбокс или трансформацию, содержимое и его система координат начинают смещаться и масштабироваться:

```
<svg width="350" height="200" viewBox="0 0 237 300">  
...  
</svg>
```



Отсчёт координат содержимого начинается из левого верхнего угла (в точке 0,0). Без вьюбокса это левый верхний угол вьюпорта (бирюзовая точка), с вьюбоксом — левый верхний край вьюбокса (красная точка).

То есть теперь расположение содержимого будет отсчитываться относительно новой системы координат, а не от вьюпорта, из-за чего фигура оказывается не слева, а ближе к центру, а системы координат больше не совпадают.

Системы координат и трансформации

Трансформации тоже создают свою систему координат. Чтобы применить трансформацию ко всему содержимому, обернём его в группу (элемент `<g>`):

```
<svg width="350" height="200">

  <g>

    ...

  </g>

</svg>
```

И добавим трансформацию:

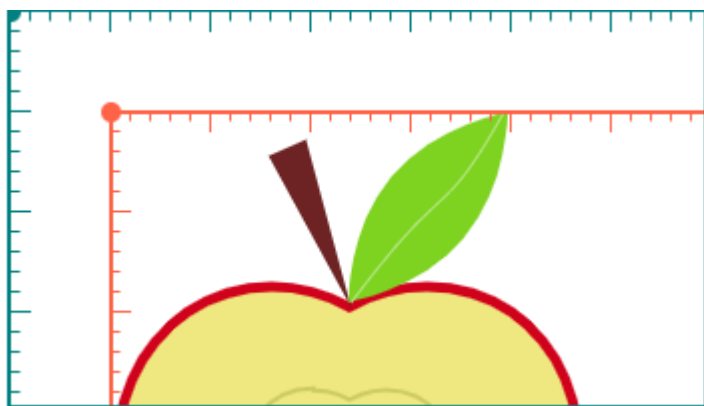
```
<svg width="350" height="200">

  <g transform="translate(50, 50)">

    ...

  </g>

</svg>
```



Всё содержимое сместилось на 50 пикселей по вертикали и по горизонтали вместе с системой координат, и если теперь добавить ещё одну трансформацию, она уже рассчитывается от новой системы координат:

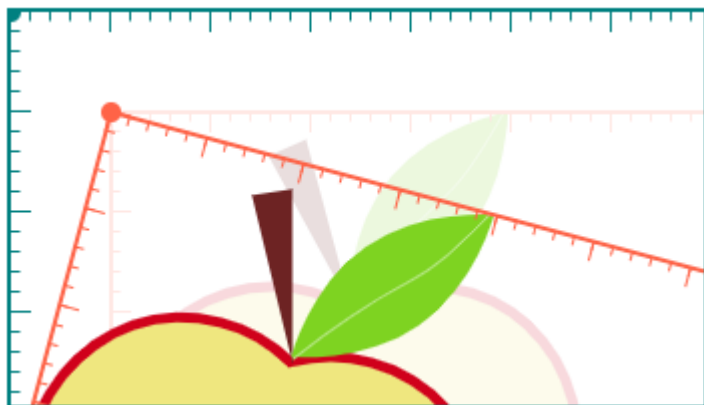
```
<svg width="350" height="200">

  <g transform="translate(50, 50) rotate(15)">

    ...

  </g>
```


</svg>



В SVG центр вращения по умолчанию находится в точке 0,0. До первой трансформации это был левый верхний угол вьюпорта, после трансформации — левый верхний угол трансформируемого содержимого. Вторая трансформация снова изменит систему координат группы.