**Лабораторная работа 6.** **Размеры в 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.

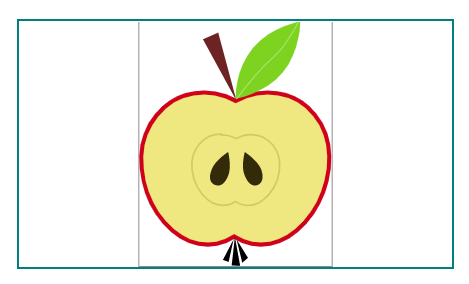
Это поведение можно изменить, задав размер области, которая будет тянуться, с помощью свойства viewBox (его можно задать только атрибутом):

<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.



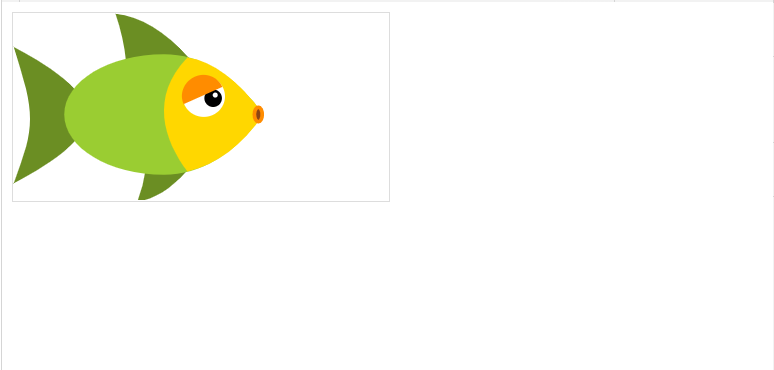
С вьюбоксом содержимое масштабируется, чтобы поместиться целиком в контейнер, и выравнивается по центру.

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.

**Задание 1. Рыбка**

Исходные размеры изображения — 200 на 150 пикселей.



<svg>

<g id="fish">

<path fill="olivedrab" d="M81.592.37c19.9 2.102 39.47 13.838 58.707 35.207L90.11 38.462C88.937 25.884 86.097 13.186 81.592.37zM99.354 150c12.867-1.36 25.852-9.268 38.954-23.727l-32.45-1.865c-1.423 8.774-3.59 17.305-6.504 25.592zM59.08 81.25c0 27.163-58.707 53.685-58.707 54.327C17.8 89.394 17.765 81.25.373 26.923c0 .952 58.707 27.164 58.707 54.327z"/>

<path fill="yellowgreen" d="M120.398 128.846c29.31 0 55.843-16.025 79.602-48.077-23.76-32.052-50.293-48.078-79.602-48.078-43.963 0-79.602 21.525-79.602 48.077 0 26.55 35.64 48.076 79.602 48.076z"/>

<path fill="gold" d="M139.13 126.32c23.436-5.678 43.726-20.86 60.87-45.55-20.02-27.17-40.293-42.348-60.818-45.537-30.57 31.603-18.83 65.88-.052 91.086z"/>

<g transform="translate(129.353 43.27)">

<ellipse cx="22.886" cy="23.077" fill="#FFF" rx="16.915" ry="16.346"/>

<circle cx="30.348" cy="24.519" r="7"/>

<circle r="2" cx="32" cy="22" fill="white"/>

<path fill="darkorange" d="M37.683 15.49C34.01 7.243 24.116 3.637 15.58 7.437c-8.533 3.8-12.475 13.566-8.803 21.813l30.906-13.76z"/>

</g>

<path fill="saddlebrown" stroke="darkorange" stroke-width="3" d="M195.788 86.538c1.728 0 3.13-2.583 3.13-5.77 0-3.185-1.402-5.768-3.13-5.768-1.73 0-3.13 2.583-3.13 5.77 0 3.185 1.4 5.768 3.13 5.768z"/>

</g>

</svg>

**style.css**

svg {

background-color: #ffffff;

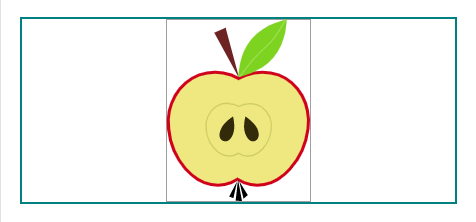
border: 1px solid #dddddd;

}

Увеличьте рыбку до 400 на 300.

# **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:

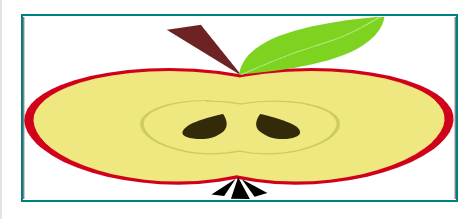
<svg viewBox="0 0 237 300" preserveAspectRatio="none">

…

</svg>

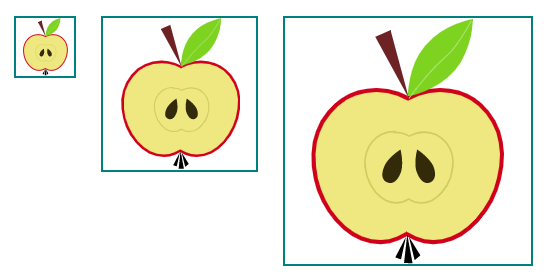
В этом случае область, размеры которой заданы вьюбоксом, растягивается на всё доступное пространство вьюпорта:

preserveAspectRatio задаётся только атрибутом.



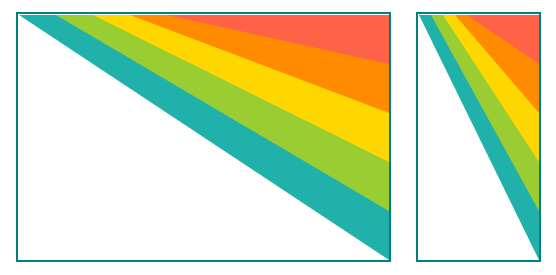
# **Резиновый фон с preserveAspectRatio**

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:



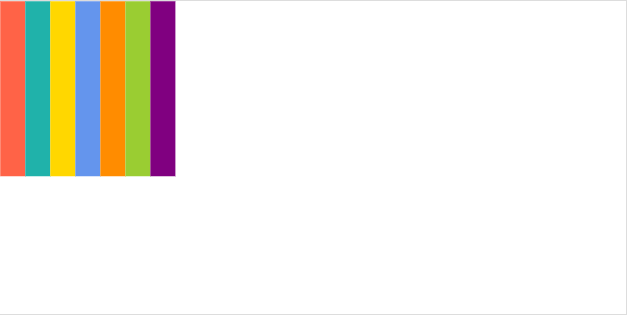
Это очень удобно для иконок: задайте размеры родительскому элементу, и иконка, заданная фоном, сама под него растянется.

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



**Задание 2. резиновые полосы**

Исходные размеры изображения — 140 на 140. Добавьте <svg> нужные атрибуты, чтобы при текущих размерах SVG-элемента полосы заполняли всё доступное пространство.



<svg width="500" height="250">

<g id="colored-stripes">

<path fill="tomato" d="M0 0h20v140H0z"/>

<path fill="lightseagreen" d="M20 0h20v140H20z"/>

<path fill="gold" d="M40 0h20v140H40z"/>

<path fill="cornflowerblue" d="M60 0h20v140H60z"/>

<path fill="darkorange" d="M80 0h20v140H80z"/>

<path fill="yellowgreen" d="M100 0h20v140h-20z"/>

<path fill="purple" d="M120 0h20v140h-20z"/>

</g>

</svg>

**style.css**

svg {

background-color: #ffffff;

border: 1px solid #dddddd;

}

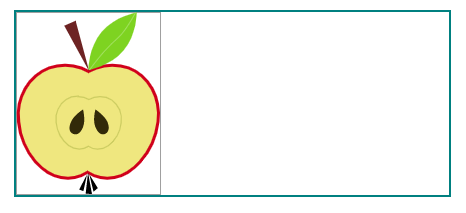
# **Выравнивание в preserveAspectRatio**

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:

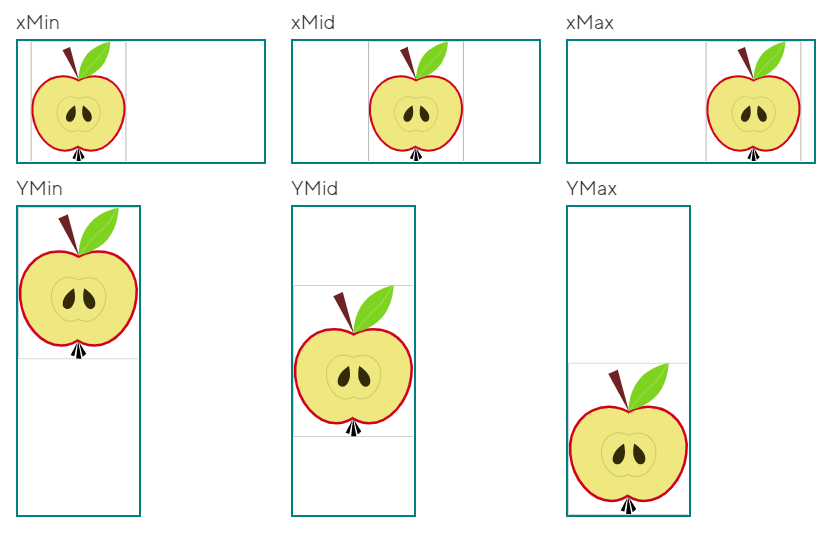
<svg viewBox="0 0 237 300" preserveAspectRatio="xMinYMid">

…

</svg>



Возможные значения для каждой оси:



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**

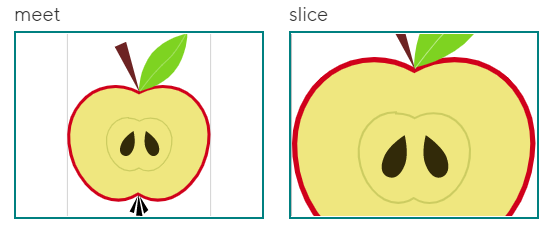
Второй параметр в свойстве preserveAspectRatio задаёт поведение содержимого относительно вьюпорта, определяет как именно содержимое заполняет пространство:

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

…

</svg>

Возможные значения:



**meet** — содержимое умещается 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:

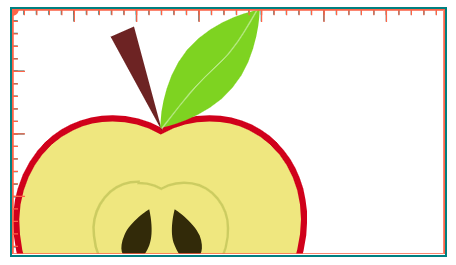
1. система координат вьюпорта — viewport space;
2. система координат содержимого — user space

Изначально системы и их единицы измерения соответствуют друг другу:

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

…

</svg>



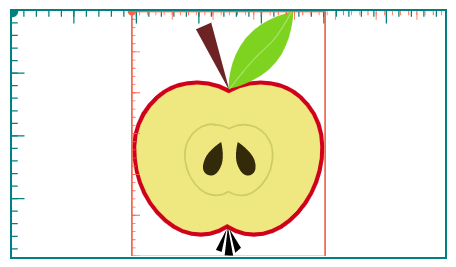
Сейчас видно только систему координат содержимого (она показана красным), потому что системы совпадают и одна скрыта под другой.

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.

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:

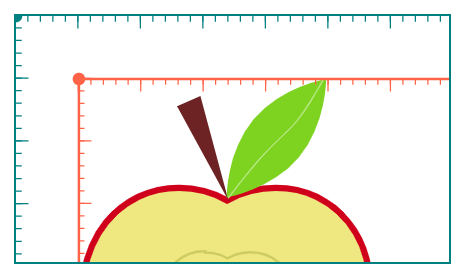
<svg width="350" height="200">

<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:

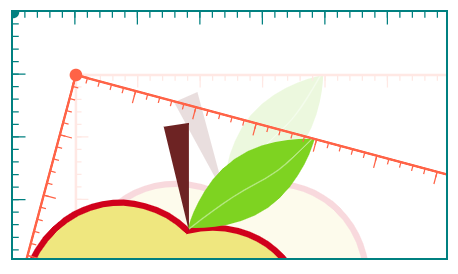
<svg width="350" height="200">

<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.