

# **Автоматическая анимация, анимированные СИМВОЛЫ**

**Урок 5**

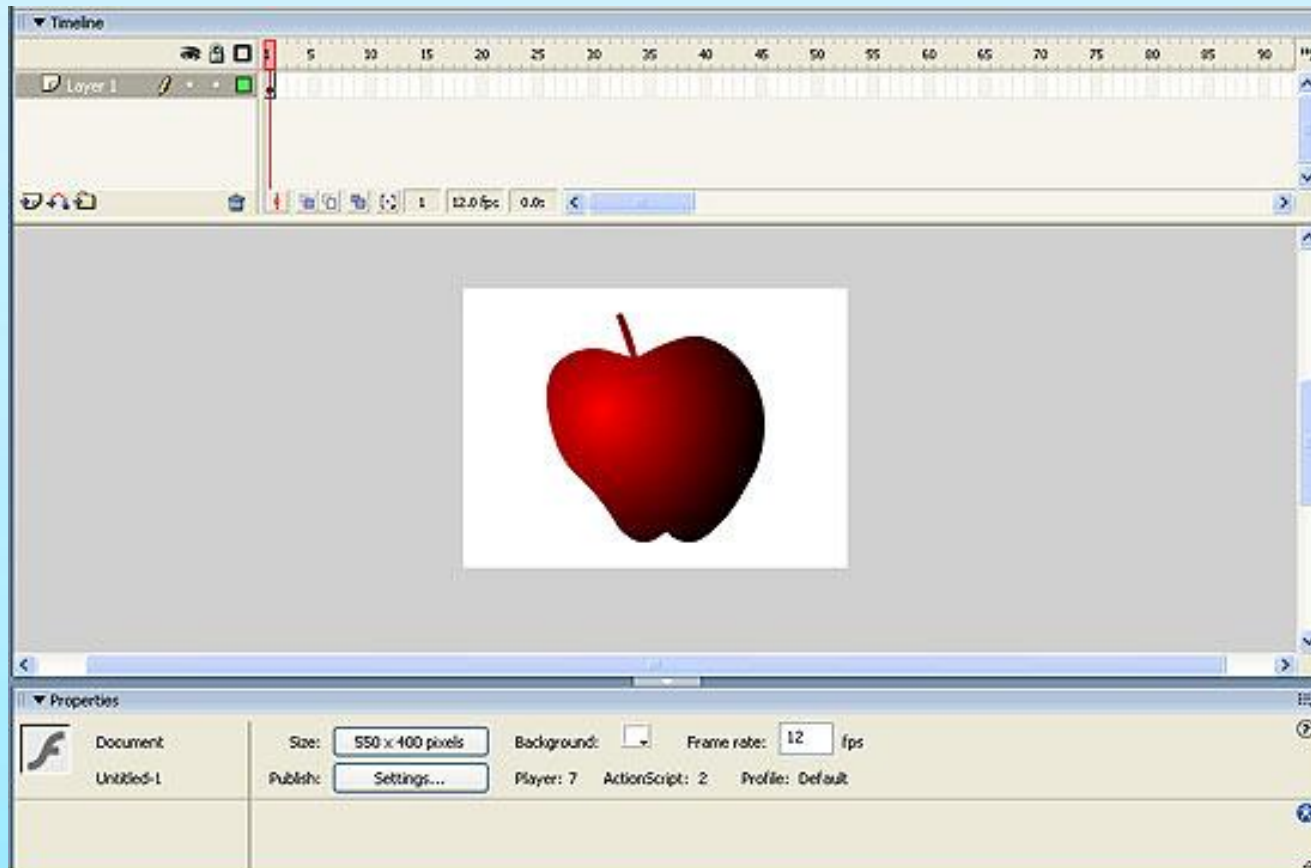
# Содержание:

- Автоматическая анимация формоизменения
- Использование хинтов в анимации формы
- Использование символов в сложной анимации
- Создание анимированного графического символа:
- Создание мувиклипов
- Преобразование анимации на сцене в отдельный мувиклип
- Дублирование символов

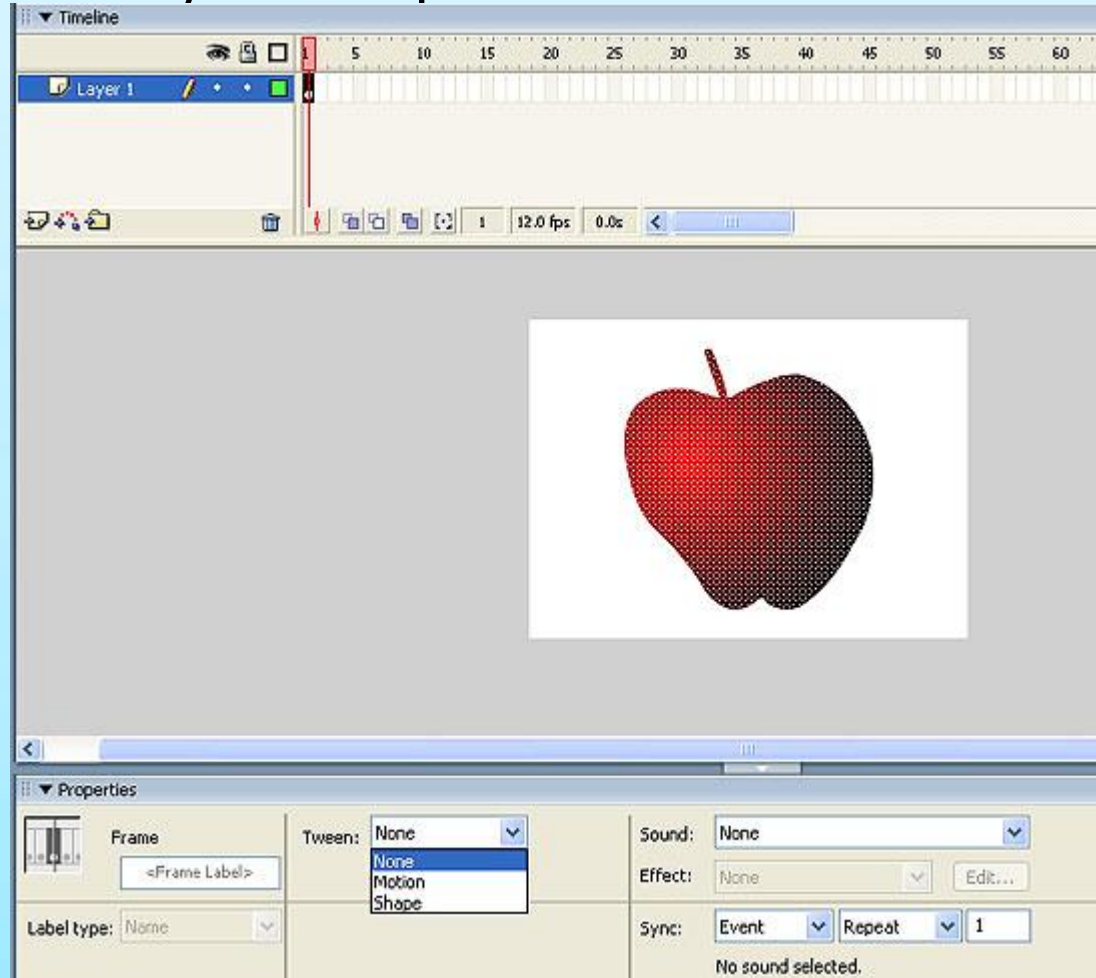
# АВТОМАТИЧЕСКАЯ АНИМАЦИЯ ФОРМОИЗМЕНЕНИЯ

- Сегодня рассмотрим автоматическую анимацию формоизменения (Shape Tween). Имеется в виду, что данный вид анимации строится на базе начального и конечного кадра, а всё, что находится между этими кадрами, интерполируется по тем или иным алгоритмам. Анимация формоизменения позволяет создать эффект перетекания одной фигуры в другую с одновременным изменением расположения, размера и цвета фигур.
- Нужно отметить, что анимацию формоизменения можно применить только к фигурам, но нельзя применять ни к группам, ни к экземплярам, ни к растровым изображениям. Для того чтобы применить к этим объектам анимацию формоизменения, их необходимо вначале разбить (break apart) на составляющие.

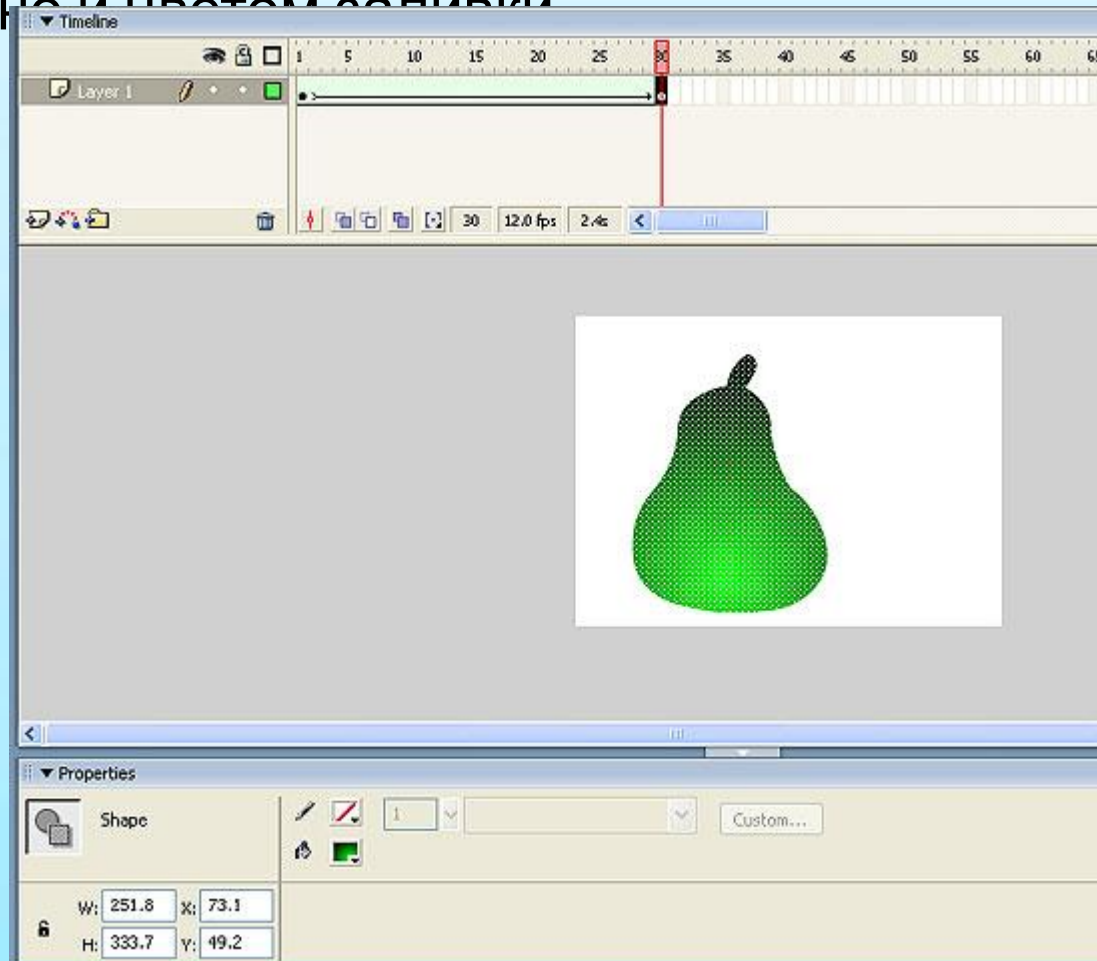
- Рассмотрим простой пример, в котором яблоко будет превращаться в грушу. Нарисуйте замкнутый контур с заливкой в форме яблока.



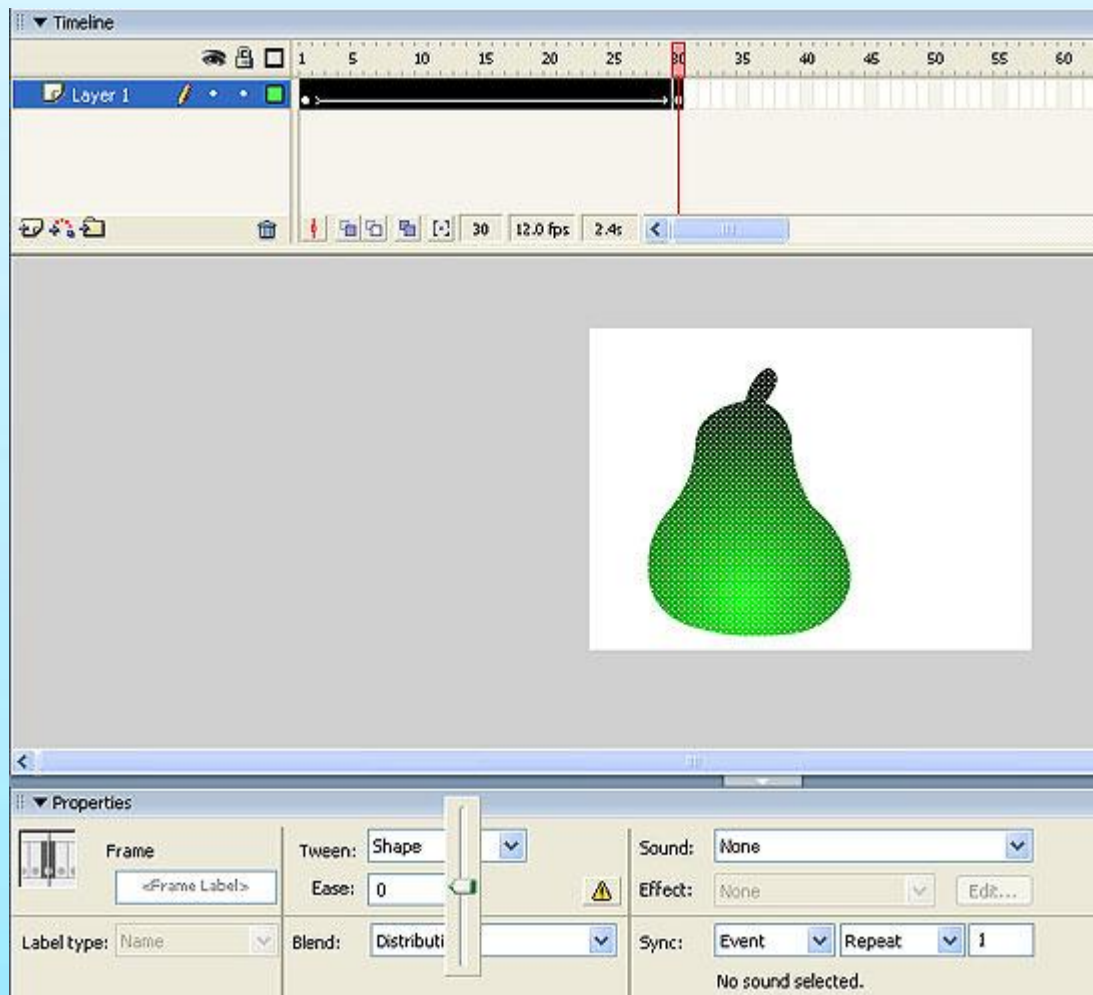
- Затем щелкните на имени слоя нарисованного объекта (в данном случае имя слоя по умолчанию принято Layer 1) — слой станет выделенным, а внизу в панели Properties появится окошко Tween, в котором нужно выбрать пункт Shape



- Теперь добавим конечный кадр — пусть формоизменение будет происходить за 30 кадров. Вставьте ключевой фрейм в 30-й кадр и поместите там замкнутый контур в форме груши. Обратите внимание, что в нашем примере начальный и последний кадр отличаются не только формой контура, но и цветом заливки.

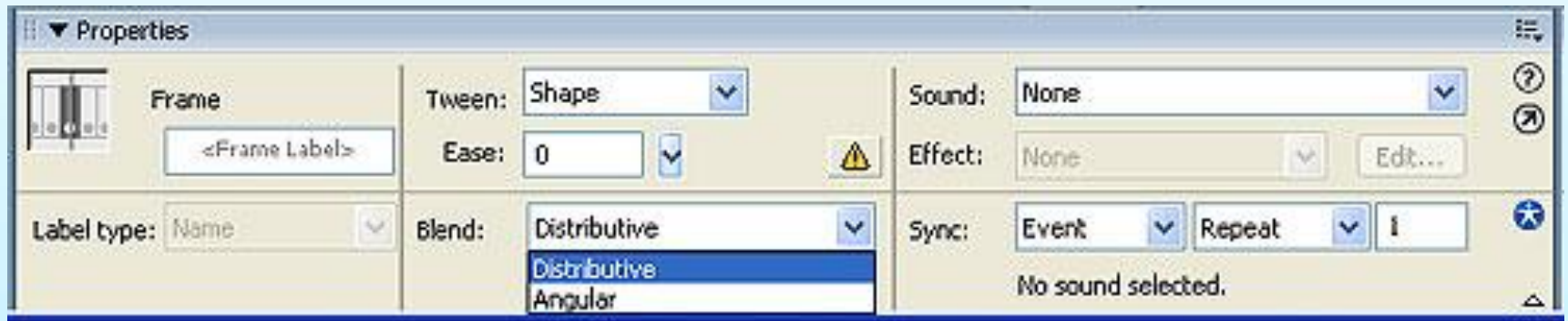


- После того как мы определили конечный кадр трансформации, на панели Timeline промежуток от первого до 30-го кадра окрасился в светло-зеленый цвет, а внутри этого промежутка протянулась стрелка, что свидетельствует о создании анимации формоизменения, то есть о создании фильма.



- Аналогично тому, как это реализовано в автоматической анимации движения, в автоматической анимации формоизменения возможно задание анимации с замедлением или ускорением. Продемонстрируем эту возможность на нашем примере. Выделите слой и нажмите на треугольник рядом с окошком Ease, в результате чего появится ползунок, который позволяет установить режим ускорения/замедления. Чтобы движение ускорились, необходимо ввести отрицательное число от  $-1$  до  $-100$ , а для замедления процесса превращения введите положительное число от  $1$  до  $100$ . Зададим параметр Easing, например равный  $+100$ .



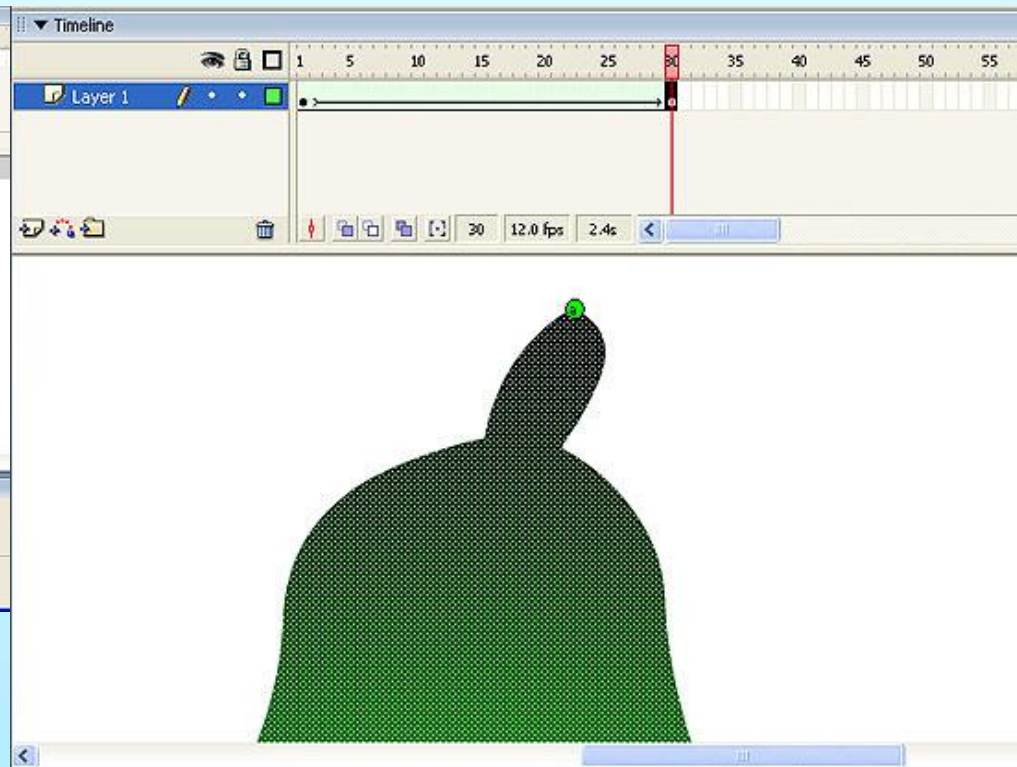
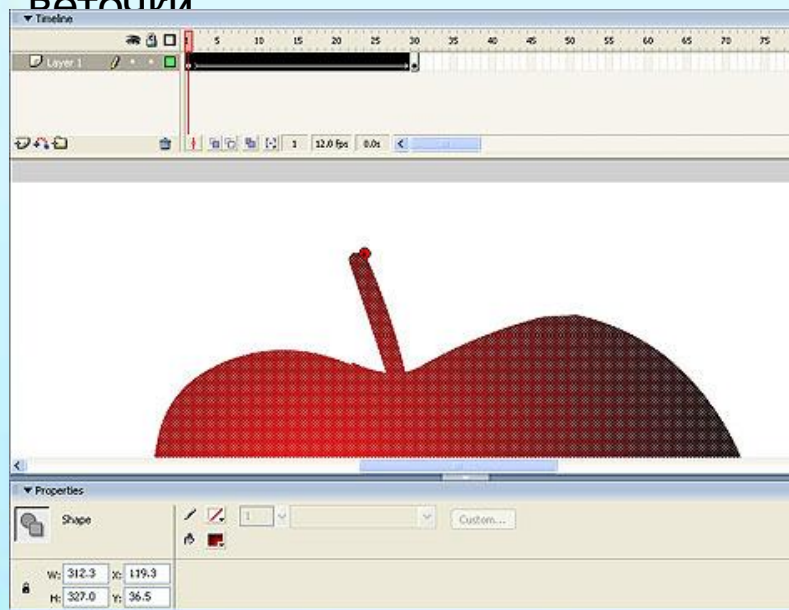


- Обратите внимание на возможность выбора характера анимации, которое задается в выпадающем меню Blend. В данном меню можно выбрать два варианта — Distributive, при котором промежуточные фигуры будут более гладкими, и Angular, при котором фигуры будут угловатыми.

# ИСПОЛЬЗОВАНИЕ ХИНТОВ В АНИМАЦИИ ФОРМЫ

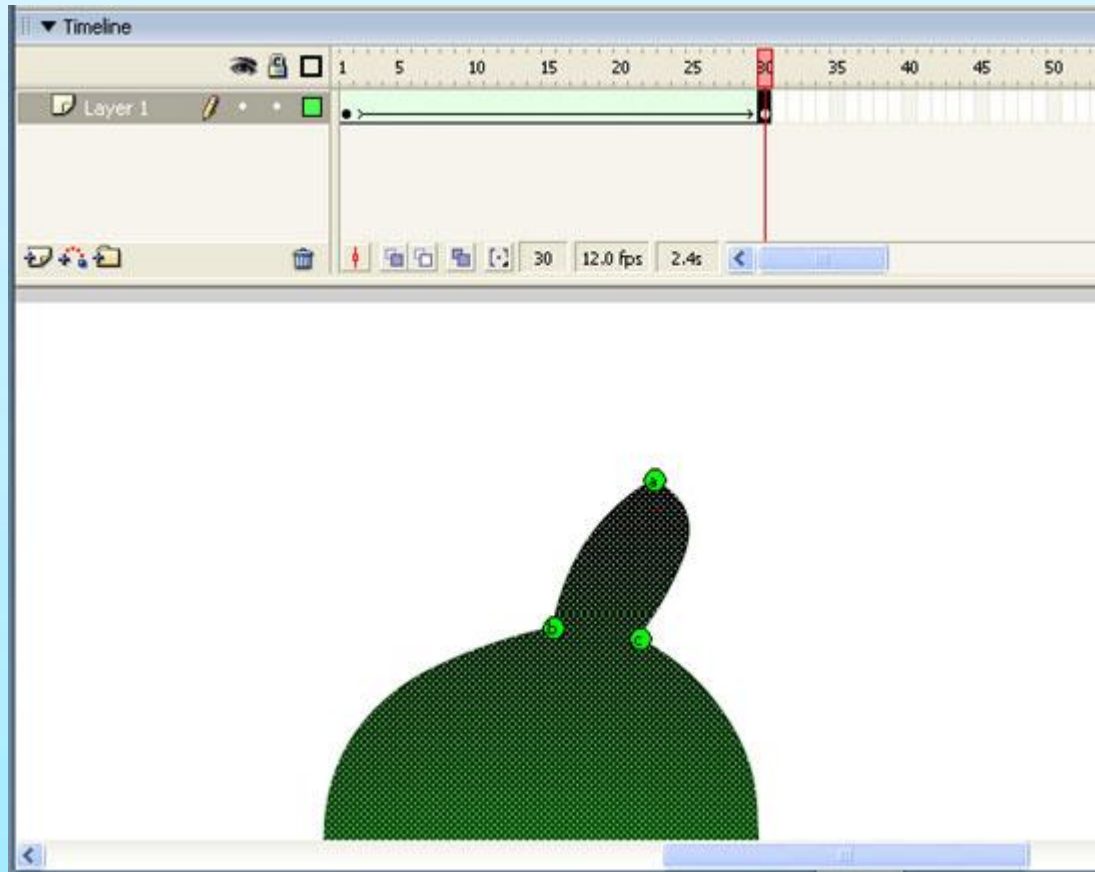
Ранее мы изучали процесс анимации формоизменения, при котором яблоко превращается в грушу. Логичным был бы процесс морфинга, при котором веточка переходит в веточку, а плод в плод, программа же генерирует промежуточные кадры преобразования, «ничего не зная» о строении яблок и груш. Поэтому, если мы хотим сохранить некоторую логику морфинга, необходимо указать программе, какие области должны наследоваться. С этой целью в программе Flash используются так называемые **ХИНТЫ** (shape hints) — контрольные маркеры, с помощью которых можно управлять процессом формоизменения. Хинтами отмечаются точки, которые должны переходить сами в себя в начальной и конечной фигурах. Хинты помечаются буквами (от «a» до «z»).

Выделим первый ключевой кадр и выполним команду Modify => Shape => Add Shape Hint, в результате выполнения которой на фигуре появится начальный хинт в виде красного кружка с буквой «а». Передвинем хинт в верхнюю точку сетки

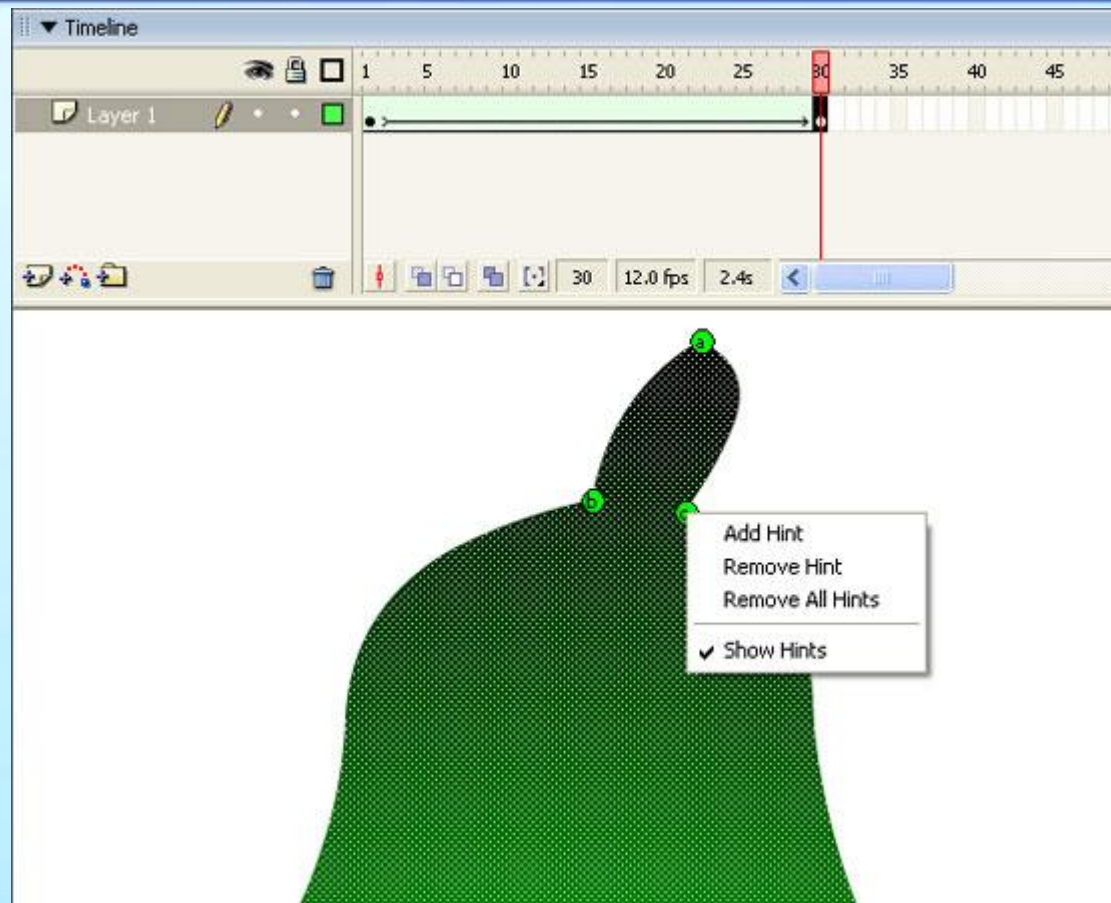


Перейдите на последний ключевой кадр, и вы обнаружите, что внутри контура результирующей фигуры появится конечный хинт в виде зеленого кружка с буквой «а». Передвиньте хинт в точку, которая должна соответствовать первой отмеченной точке.

- После того как вы выставите на кривую хинт в последнем кадре, в первом кадре хинт поменяет свой цвет (с красного на желтый) — это означает, что установлено соответствие между хинтами. Повторите этот процесс, добавляя новые хинты. Для получения наилучших результатов следует размещать хинты против часовой стрелки, начиная с левого верхнего угла фигуры.



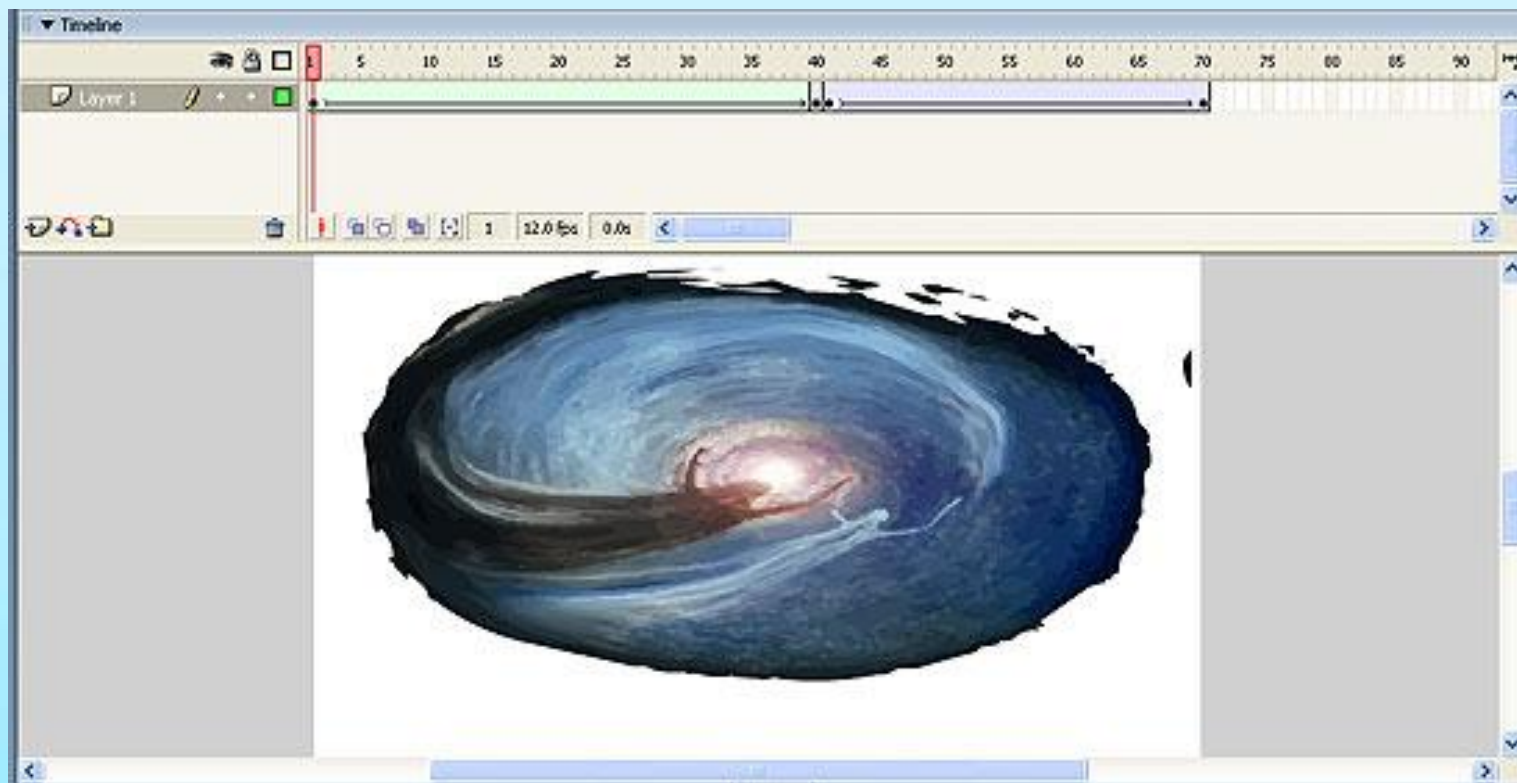
В результате мы получим фильм с более естественной трансформацией. В нашем примере нам хватило всего трех хинтов, однако программа позволяет одновременно использовать до 26 хинтов. При анимации фигур сложной формы можно разбить процесс на несколько этапов и на каждом использовать по 26 хинтов.



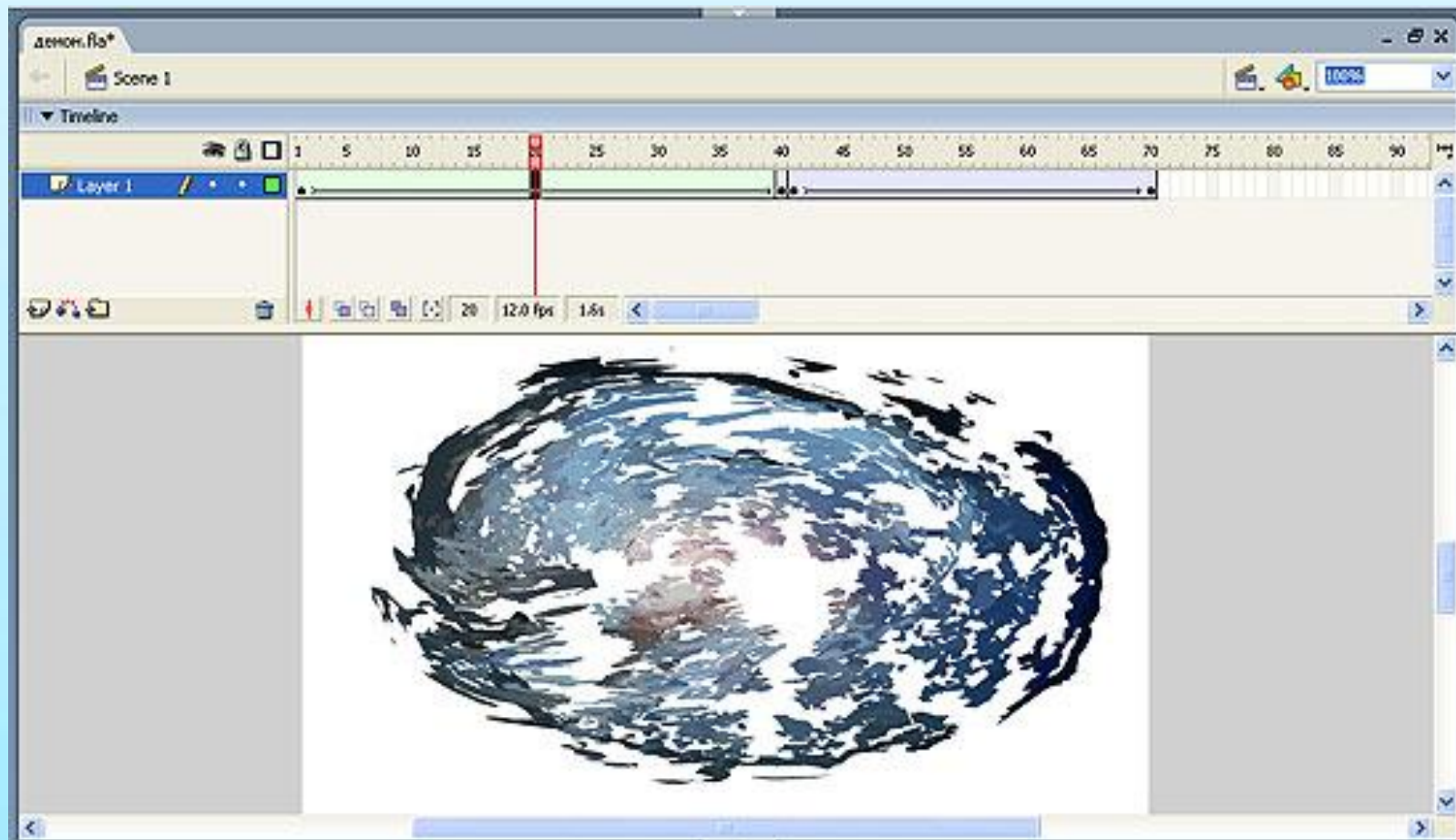
- Сделать хинты видимыми или невидимыми можно по команде View => Show Shape Hints. Чтобы удалить хинт, перетащите его за сцену, а чтобы удалить все хинты, выберите команду Modify => Transform => Remove All Hints. Данные команды удобнее выполнить правым щелчком мыши.

- До сих пор мы говорили о том, как упорядочить процесс морфинга, однако следует отметить, что с помощью функции Shape Tween можно добиться очень интересных эффектов неупорядоченной трансформации, которые трудно получить другими способами. В частности, стандартный эффект «взрыв», который предлагает программа Flash MX, смотрится довольно примитивно по сравнению с тем, что можно сделать посредством функции Shape Tween.

- Приведем пример, где в первом кадре была помещена растровая картина, которая затем была переведена в векторный вид командой Trace Bitmap.



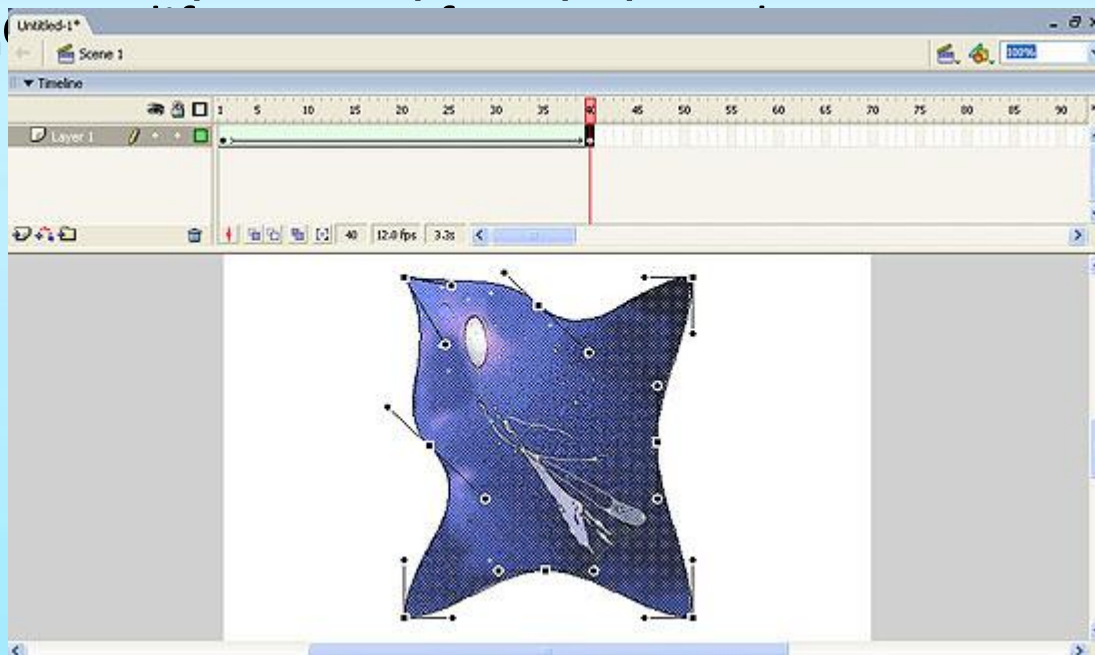
- Затем ко всей совокупности контуров с заливками была применена автоматическая анимация формоизменения. В 40-м кадре была помещена та же картина, но в несколько большем масштабе.





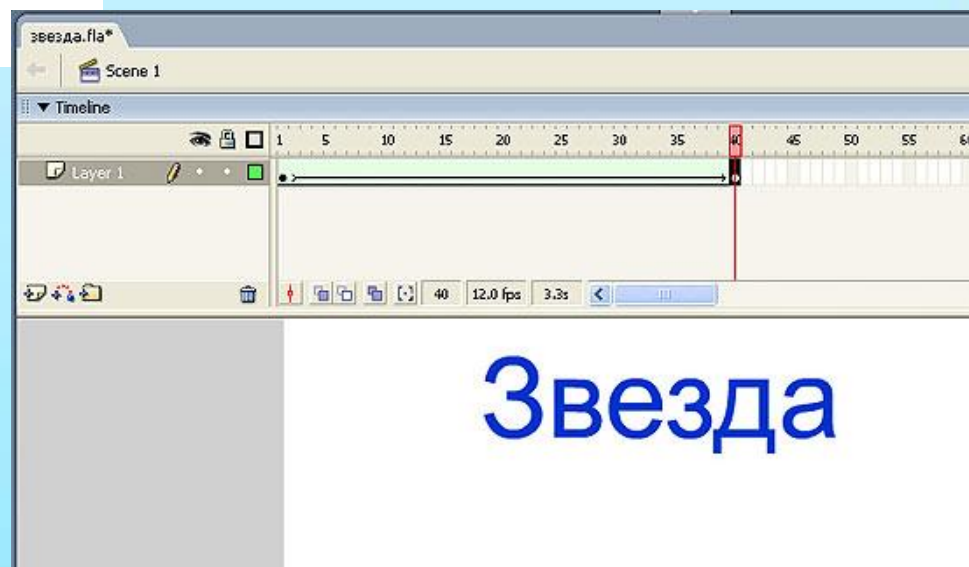
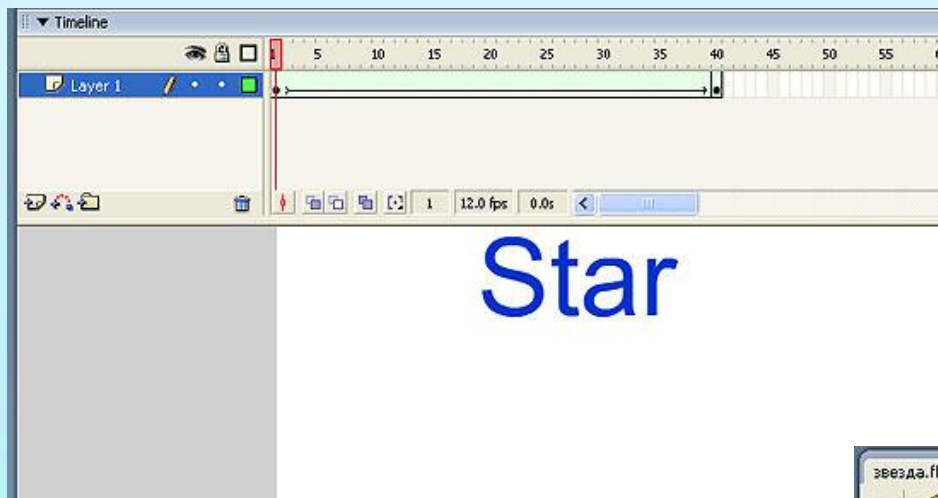
- Как видно, в процессе морфинга каждый контур с заливкой определенного цвета перемещается по своей траектории. В результате исходная картинка как бы распадается на отдельные «мазки», которые впоследствии вновь объединяются в картинку большего размера. Далее с 41-го по 70-й кадр мы применили анимацию движения Motion Tween (70-й кадр отличается от 41-го увеличением масштаба). Этот пример наглядно показывает, что функции Shape Tween и Motion Tween (которые приводят к одинаковым результатам масштабирования при переходе из первого кадра в последний) создают совершенно разные эффекты перехода: в первом случае каждый контур заливки перемещается по своей траектории, а во втором картинка остается целостной и лишь подвергается непрерывному масштабированию.

- В начале урока мы говорили о том, что автоматическую анимацию нельзя применить непосредственно к растровым изображениям. Однако если перевести растровое изображение в контур с растровой заливкой, то становится возможной анимация формоизменения контура. Для примера импортируем на сцену растровый рисунок, применим команду Modify => Break Apart, после чего установим в поле Tween тип анимации Shape и вставим в конечном кадре анимации ключевой кадр. В последнем кадре применим режим трансформации Envelope, доступный по команд



- В результате получим фильм в котором, растровое изображение не претерпевает никаких изменений, а происходит формоизменение только векторного контура, благодаря чему зрителю видна та или иная часть растровой заливки, ограниченная изменяющимся контуром. Поэтому некорректно говорить о том, что анимация формоизменения применена к растровому изображению, как и о том, что возможна анимация формоизменения применительно к тексту. Анимацию формоизменения можно применить к тексту только после того, как мы переведем текст в контуры с заливками.

В первом кадре напечатано слово Star, после чего дважды применена функция Break apart, что привело к переводу текста в контуры с заливками, далее применена анимация формоизменения, а в конечном кадре аналогичная процедура перевода текста в графику проведена со словом «Звезда».



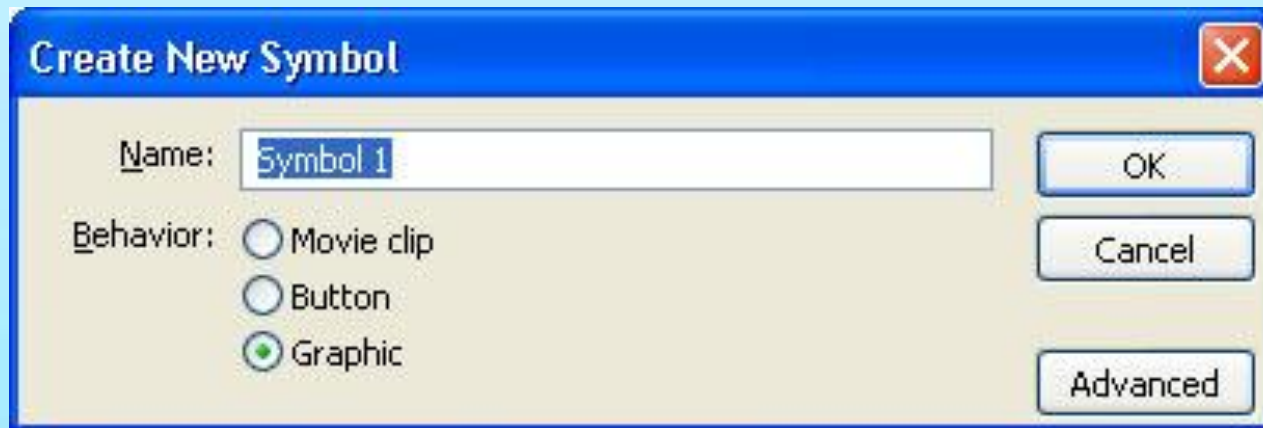
- В итоге получается фильм, в котором слово «Star» трансформируется в слово «Звезда». В принципе, подобного рода трансформации (когда количество символов в начальном и конечном кадре разное) выглядят не очень эффектно. Если же в начальном и конечном кадре одинаковое количество символов и идет переход из символа в символ, а при трансформации символа в символ используются хинты, можно добиться более выразительных эффектов.

# ИСПОЛЬЗОВАНИЕ СИМВОЛОВ В СЛОЖНОЙ АНИМАЦИИ

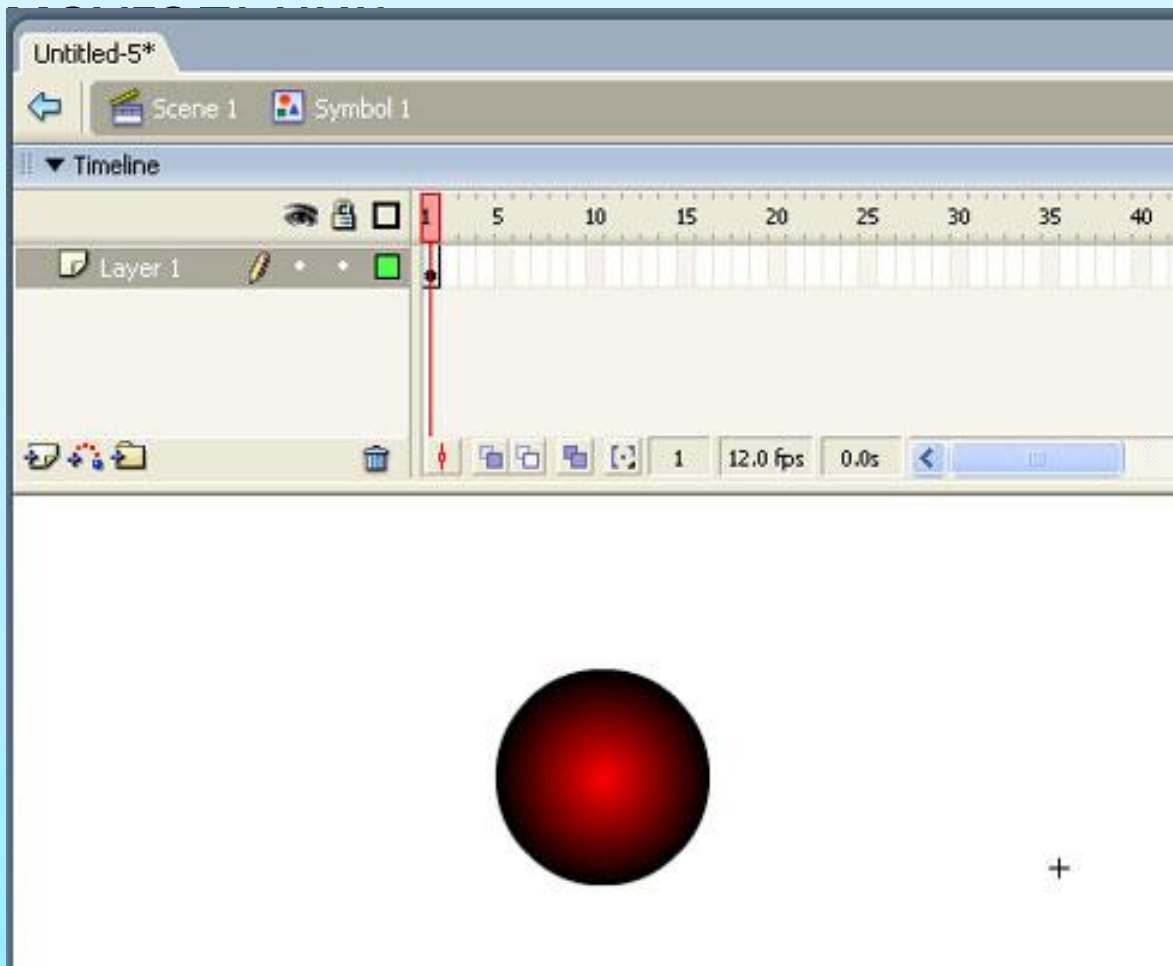
Возможность автоматической прорисовки промежуточных кадров позволяет резко сократить объем работы для большого класса задач, но этим возможности Flash по автоматизации разработки анимации не ограничиваются. Существует целый ряд задач, при решении которых Flash позволяет значительно ускорить процесс создания анимации — речь идет об использовании анимированных символов. На практике часто бывает нужно создать анимацию, в которой используется несколько элементов, обладающих однотипным движением (например, растущие листья, вращающиеся колеса и т.п.). И здесь нецелесообразно будет многократно анимировать каждый элемент — логичнее записать движение в некотором анимированном символе, а затем применить копии этого символа там, где это необходимо. Для реализации указанной схемы могут быть использованы анимированные графические символы и так называемые мувиклипы. Рассмотрим создание каждого из этих символов.

# СОЗДАНИЕ АНИМИРОВАННОГО ГРАФИЧЕСКОГО СИМВОЛА:

Прежде всего выполним команду Insert => New Symbol, в результате чего появится панель. Выберем тип символа Graphic — по умолчанию его имя будет Symbol 1. Если вас это не устраивает, можно присвоить символу другое имя.

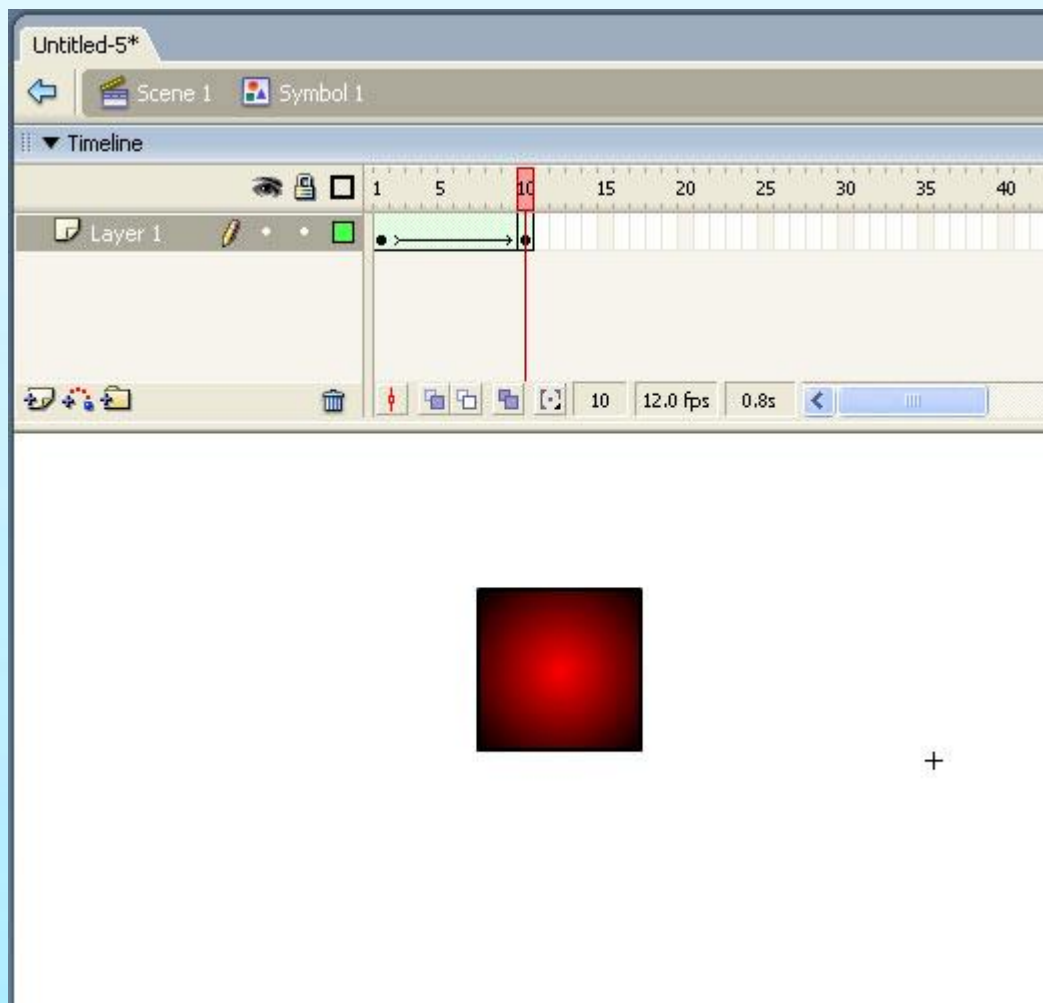


- Нажав кнопку ОК, перейдем в режим редактирования символа. Создадим символ с анимацией формоизменения — пусть это будет процесс превращения окружности в пря





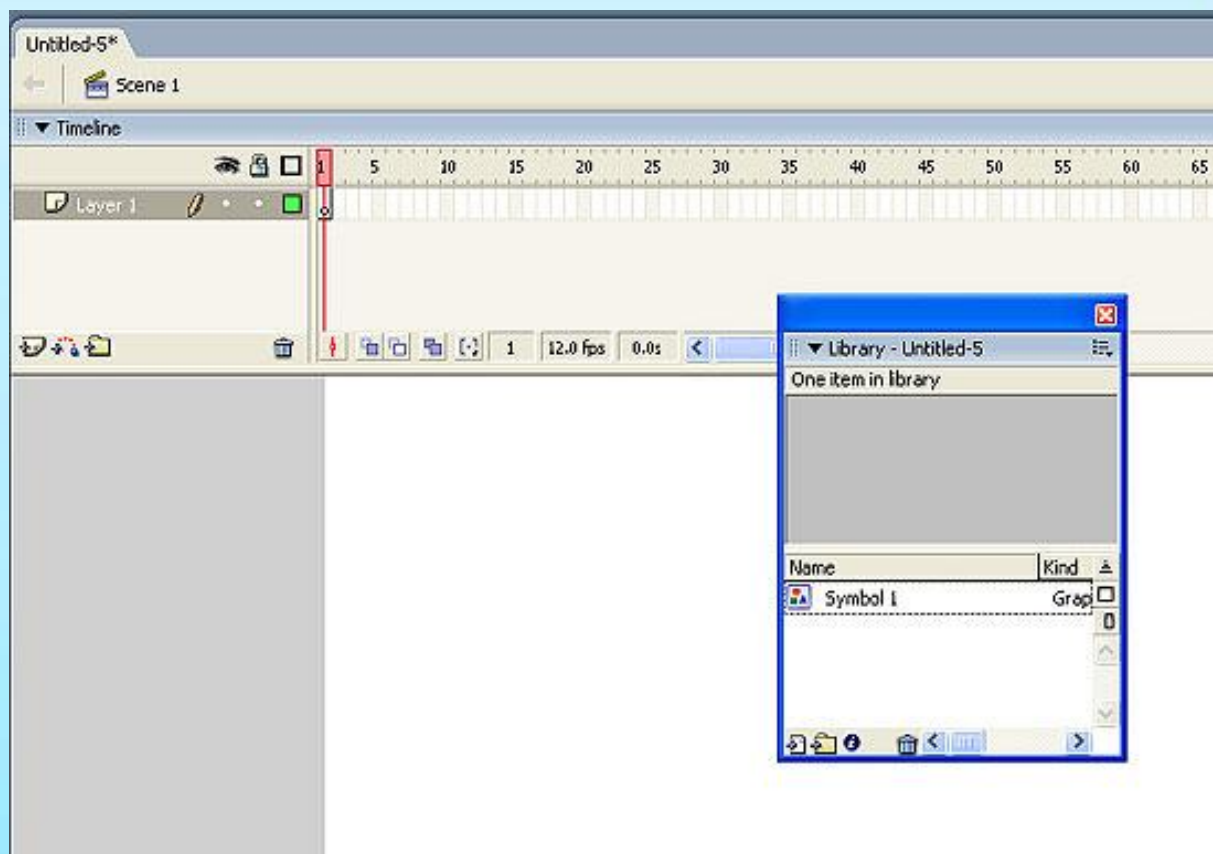
В первом кадре поместим окружность, а в последнем — прямоугольник .



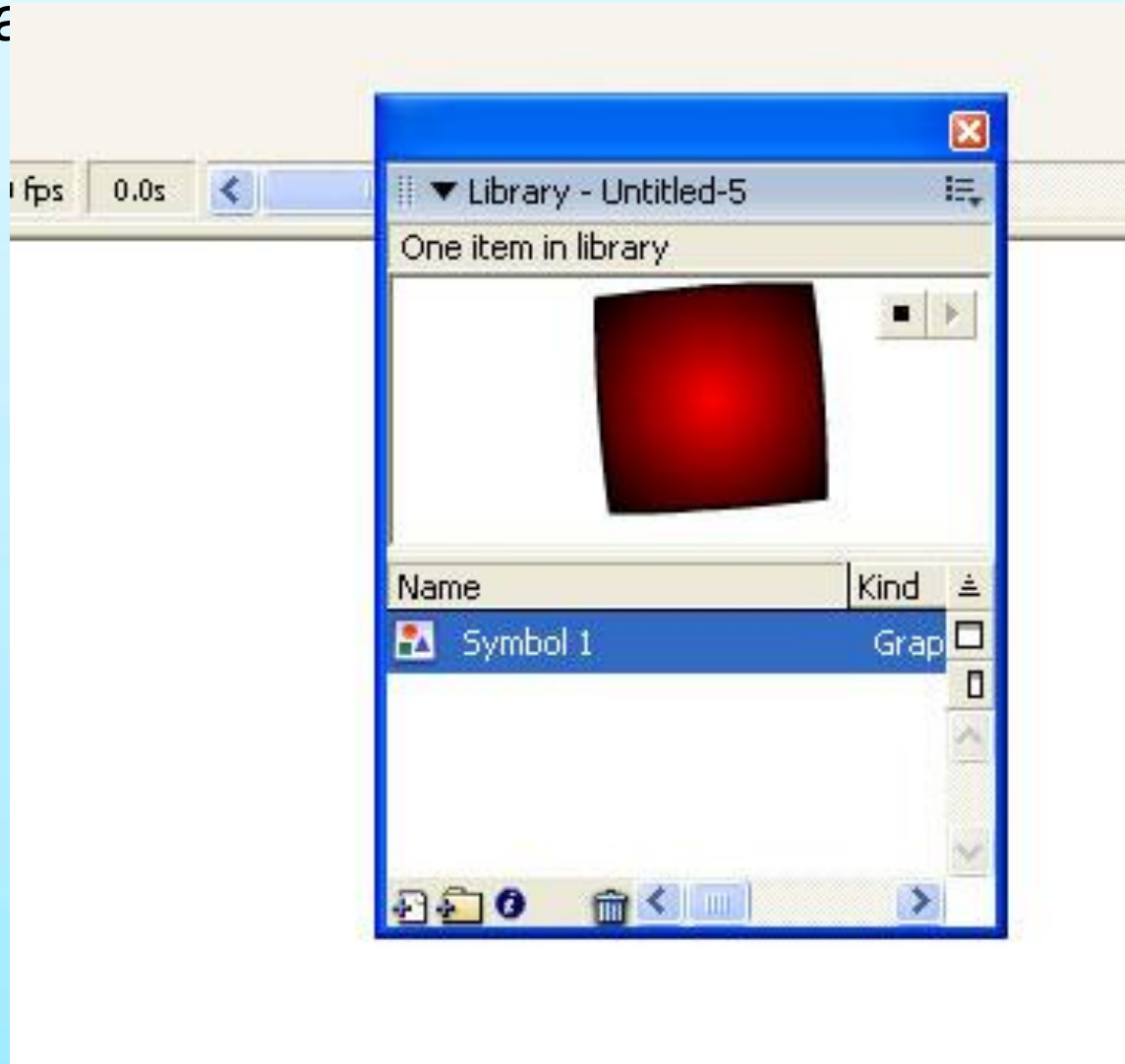
- В режиме редактирования символа его имя отображается наверху, а крестик в центре отмечает центральную точку символа.

- Таким образом, мы создали графический анимированный символ, содержащий анимацию формоизменения, которая реализуется на протяжении 10 кадров.
- После завершения редактирования символа вернемся в режим редактирования основного фильма. Это можно сделать, нажав на кнопку с именем сцены (в нашем случае Scene 1), либо используя кнопку Back, расположенную рядом с именем сцены, либо выполнив команду Edit => Edit Document.

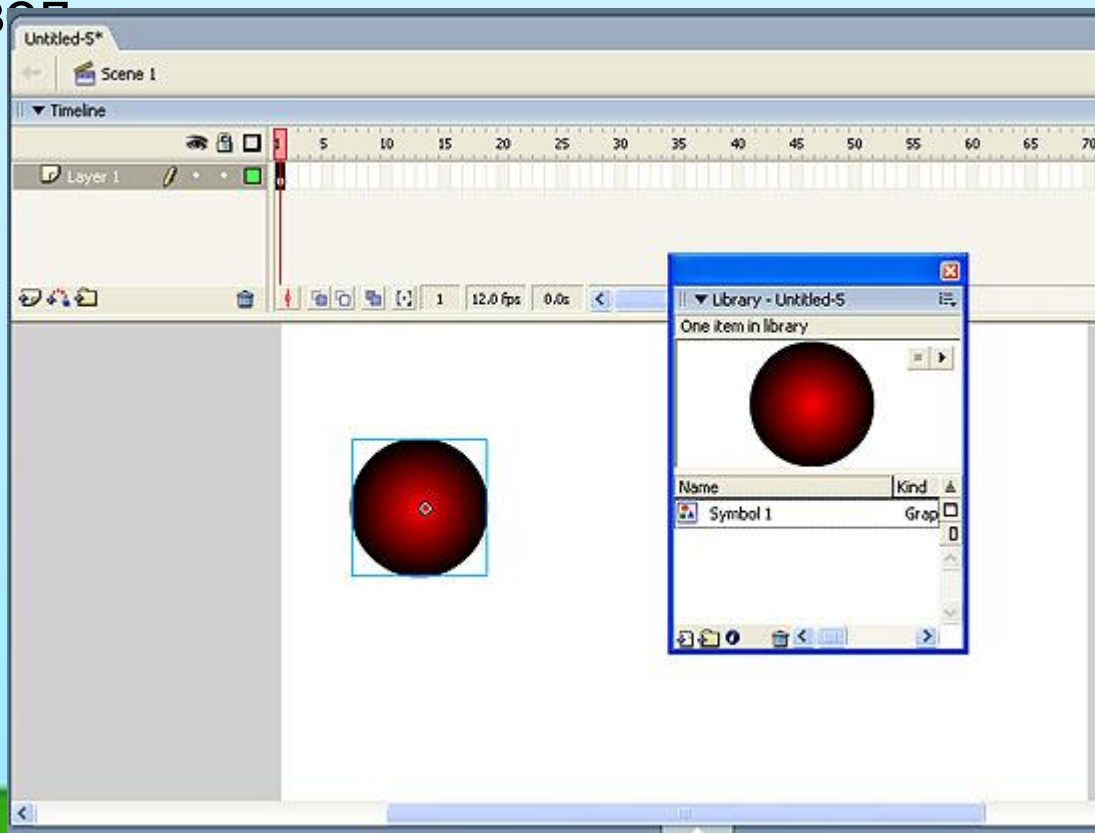
Перейдя на сцену Scene 1, выполним команду Window => Library — на экране появится панель библиотеки



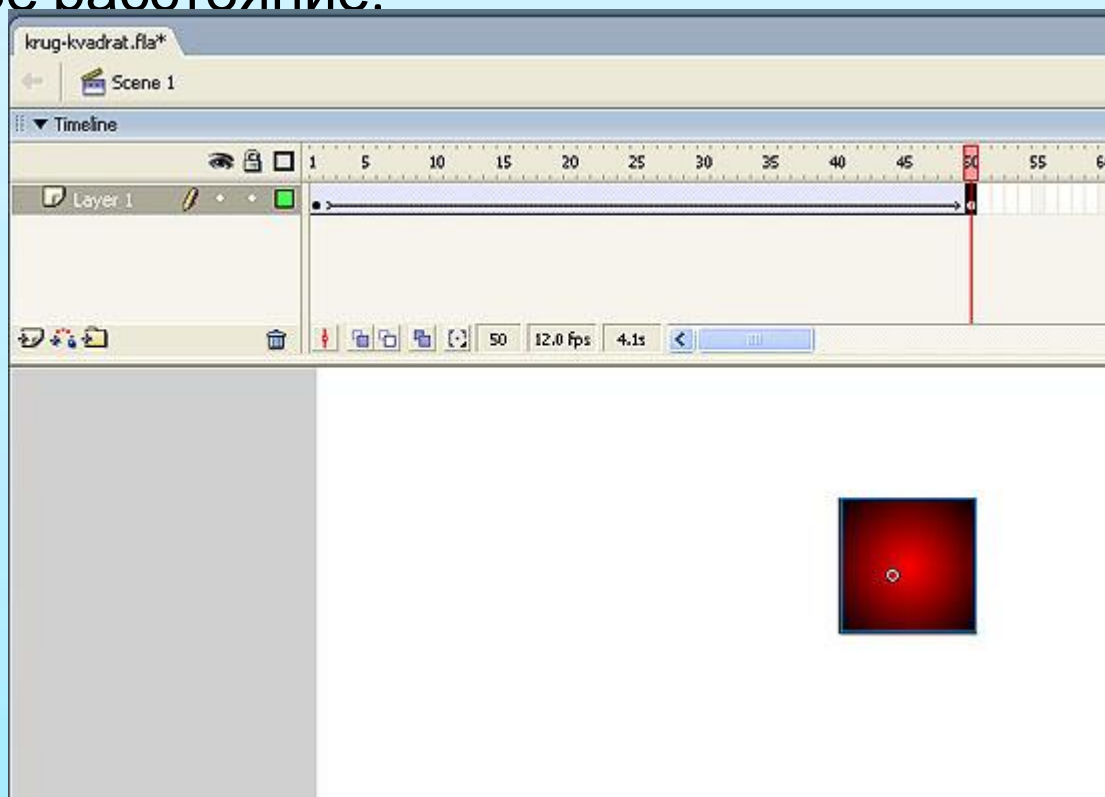
- Щелкнув по имени символа, получим его изображение в окне библиотеки. С помощью клавиши «Проигрывание» можно просмотреть анимацию символа



Перетащим символ из окна библиотеки на рабочее поле сцены. При переносе символа из библиотеки на сцену мы создаем экземпляр символа. **Экземпляр** — это копия символа, расположенная на сцене или внедренная в другой символ. Над экземпляром могут производиться различные преобразования, так что он может сильно отличаться от изначального символа цветом, размером и функцией. Редактирование символа приводит к автоматическому обновлению всех экземпляров, но применение различных эффектов к экземплярам никак не влияет на символ.

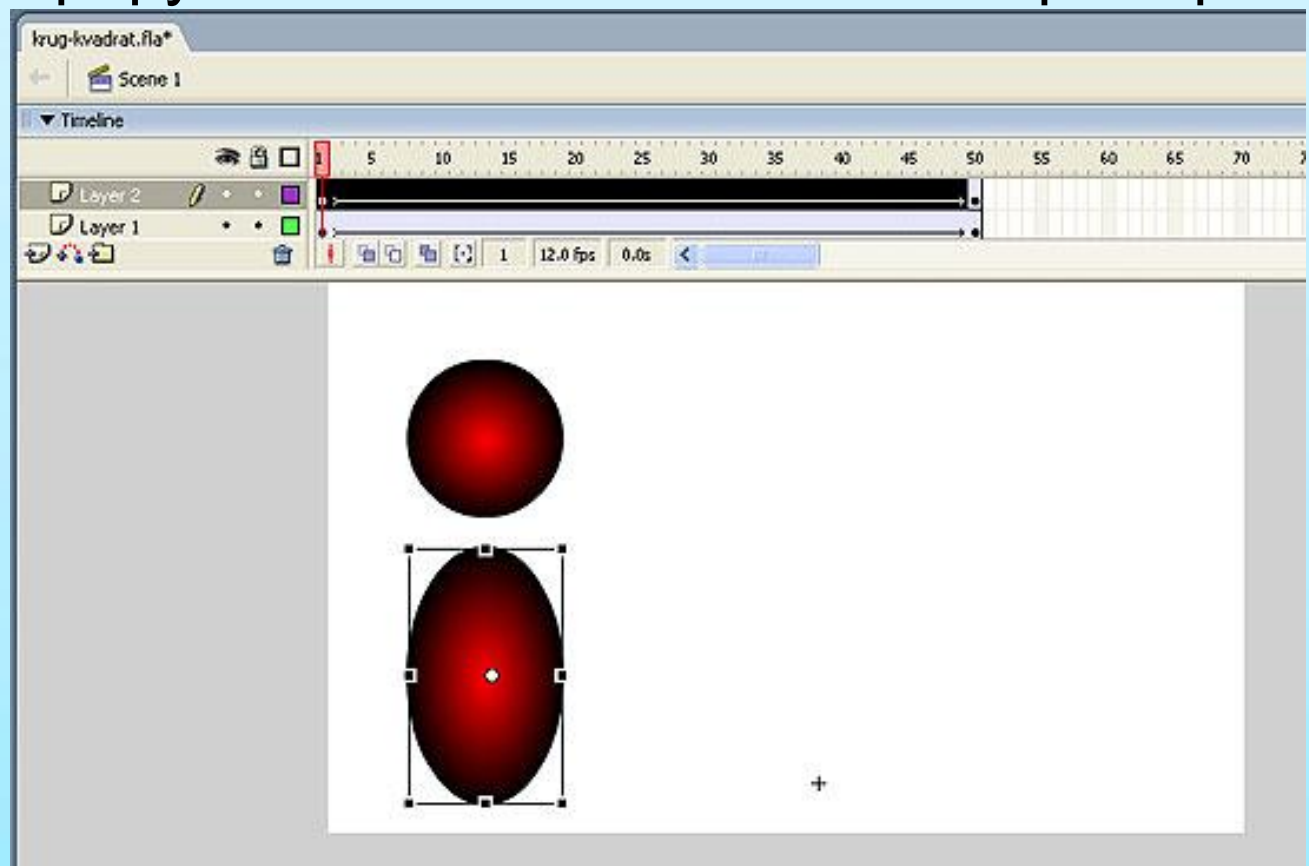


- Мы поместили на сцене анимированный графический символ, но до тех пор, пока у нас имеется только один кадр в основной сцене, мы не можем просмотреть анимацию, заложенную в данном символе.
- Создадим анимацию движения из 50 кадров, в течение которых наш символ должен переместиться вправо по сцене на некоторое расстояние.

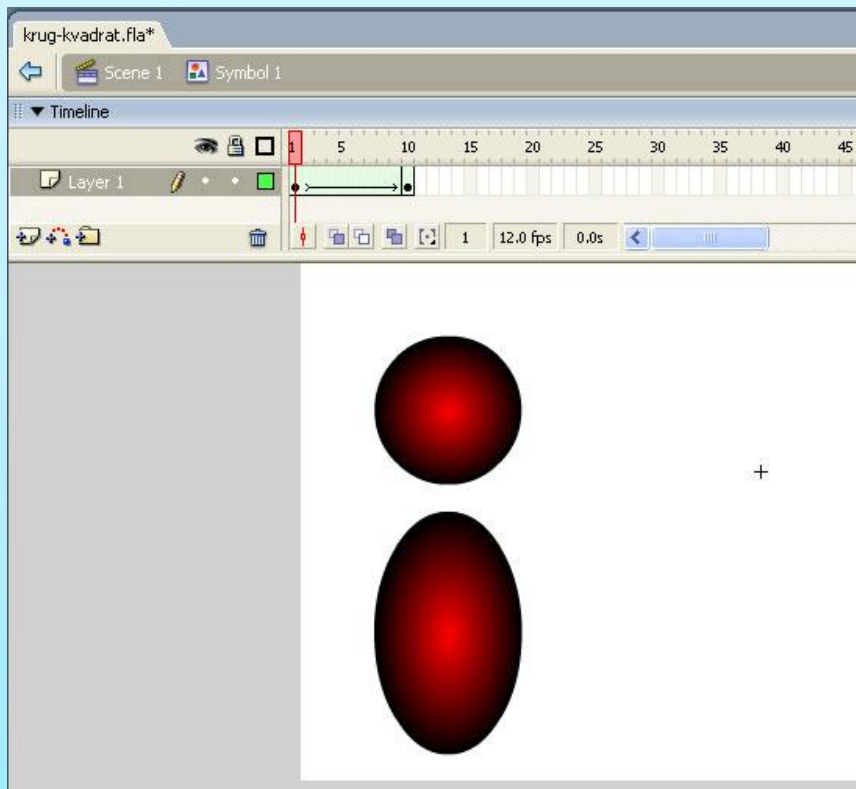


В результате мы получим фильм, в котором анимация формоизменения наложена на анимацию движения. Если взяться мышью за указатель кадра и протащить по монтажной линейке, то мы увидим, что за каждые 10 кадров отрабатывается анимация, заложенная в анимированный графический символ, а в 50-м кадре круг полностью превратился в квадрат. Если бы мы взяли в основной сцене не 50, а 55 кадров (число, не кратное количеству кадров в графическом символе), то в 55-м кадре мы бы получили другую фигуру — круг, на 50% трансформированный в квадрат. Таким образом, графические символы однозначно привязаны к монтажной линейке основного фильма.

- Если мы хотим добавить новый экземпляр символа и применить к нему ту же анимацию движения, нам придется создать для нового экземпляра символа новый слой (это тоже отличает графический анимированный символ от мувиклипа).  
Продemonстрируем все вышесказанное на примере.

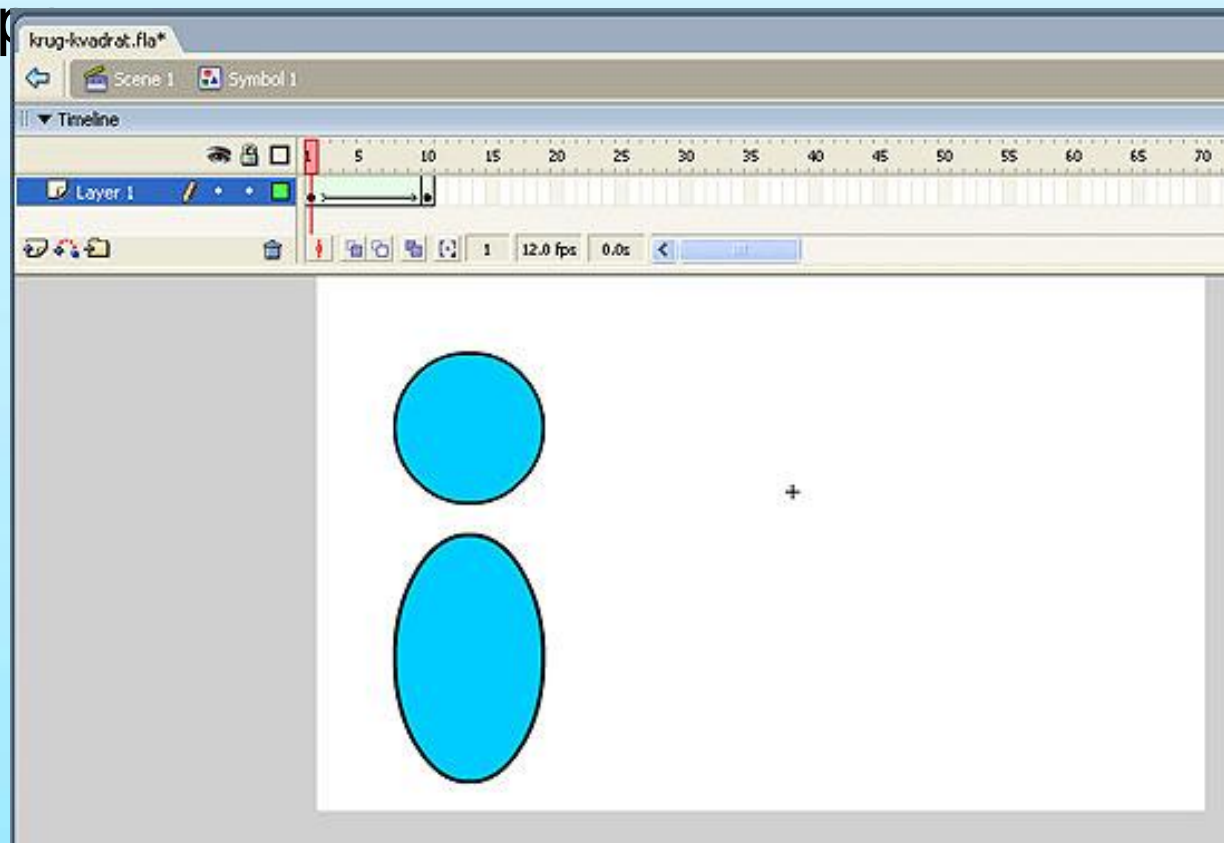


- Добавим еще один слой, скопируем в первый кадр нового слоя созданный нами символ, трансформируем его (растянем по вертикали), используя команду Modify => Transform => Free transform, и создадим анимацию движения, аналогичную той, что мы создали в первом слое. Теперь рассмотрим, как повлияет на наш фильм редактирование самого символа. Для этого дважды щелкнем по экземпляру символа в виде окружности в первом кадре, в результате чего появится сцена с 10 кадрами, соответствующими анимации нашего графического анимированного символа





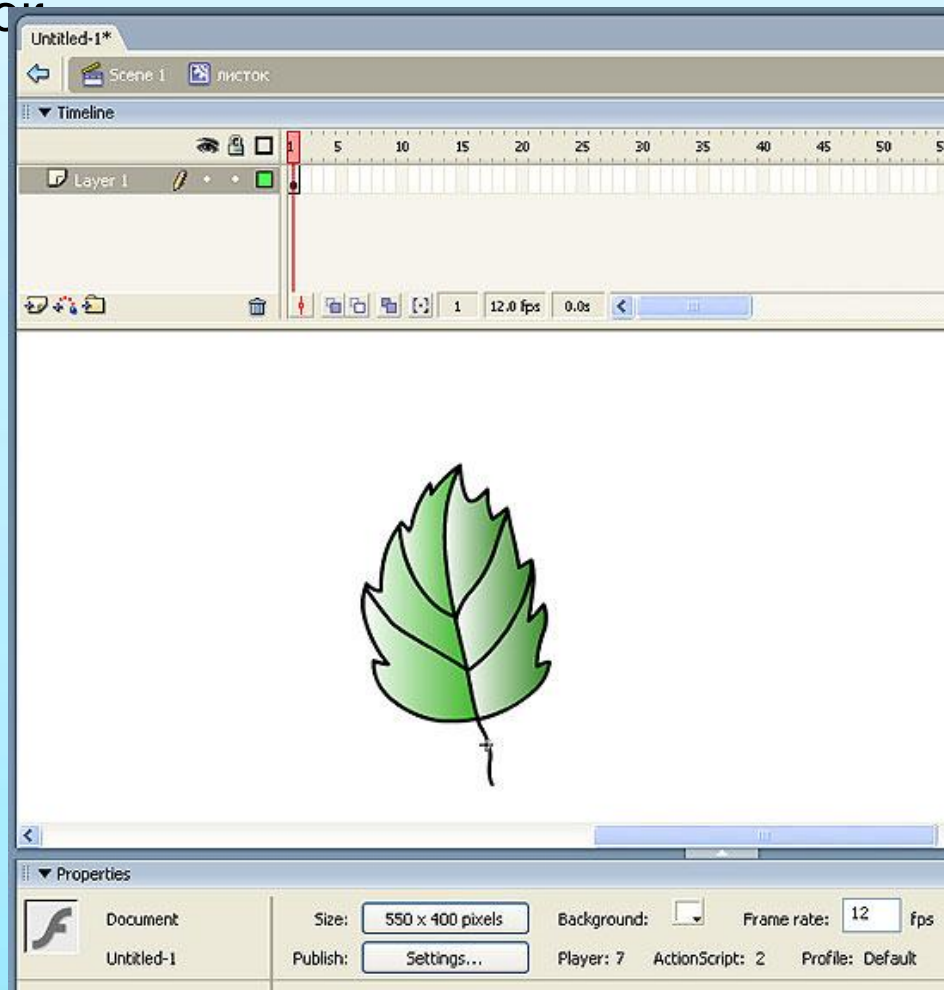
Зальем контур окружности в первом кадре заливкой другого цвета, но не градиентной, а одноцветной. В данном случае редактирование будет относиться уже не к экземпляру, а к самому символу. Как только вы зальете окружность другим цветом, эллипс автоматически зальется соответствующим цветом, то есть редактирование символа приводит к изменениям во всех экземплярах.



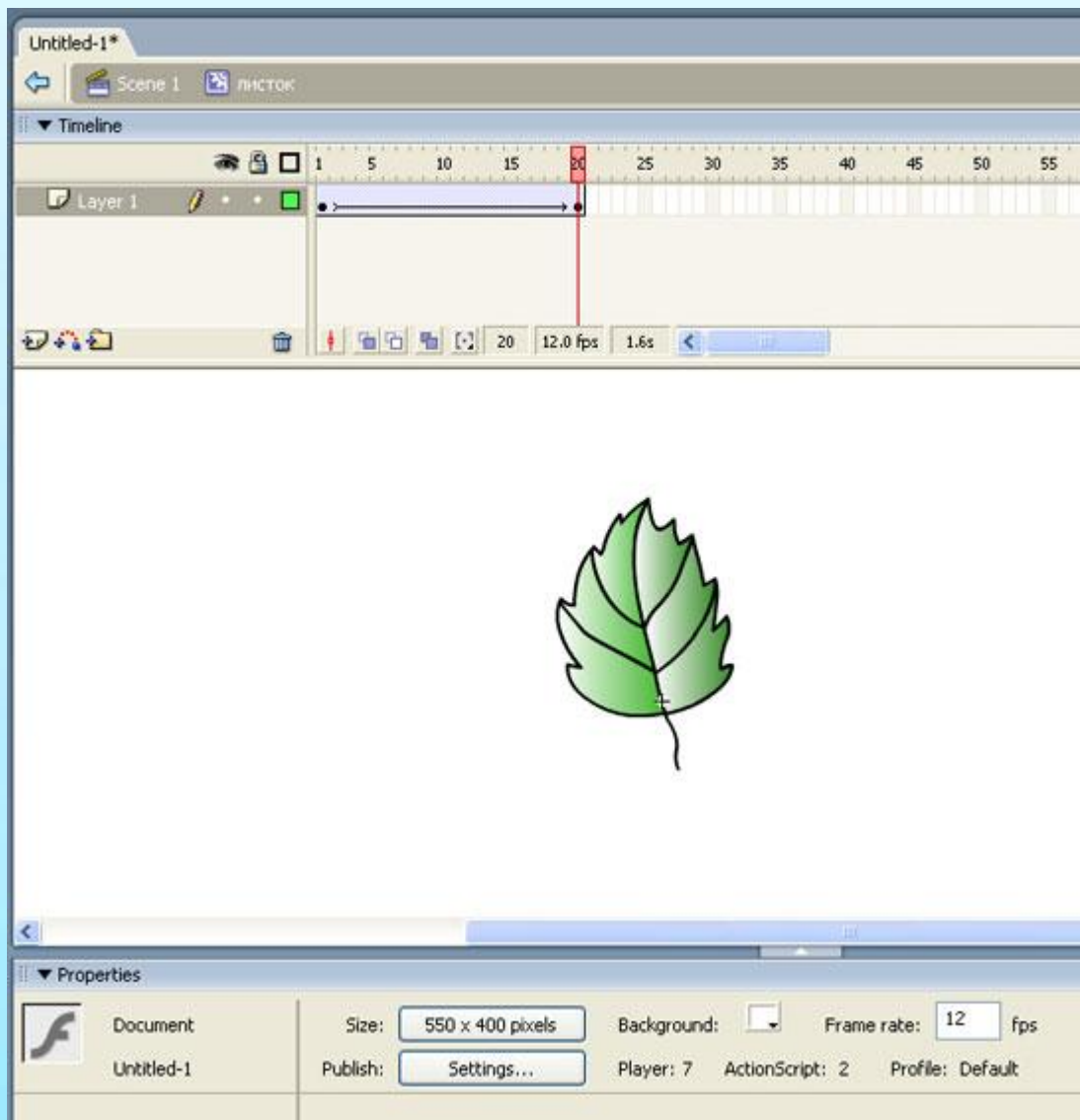
# СОЗДАНИЕ МУВИКЛИПОВ

- **Символы-мувиклипы** — это анимированные символы, имеющие монтажную линейку, которая (в отличие от графических анимированных символов) существует независимо от линейки основного фильма. Поясним это отличие: если для проигрывания в основном документе графического анимированного символа, состоящего из 10 кадров, нужно выделить ровно столько же кадров на монтажной линейке основного документа, то для мувиклипа в 10 кадров достаточно одного кадра на монтажной линейке основного документа. Кроме того, мувиклип может содержать собственные интерактивные элементы, звуки и экземпляры других мувиклипов.

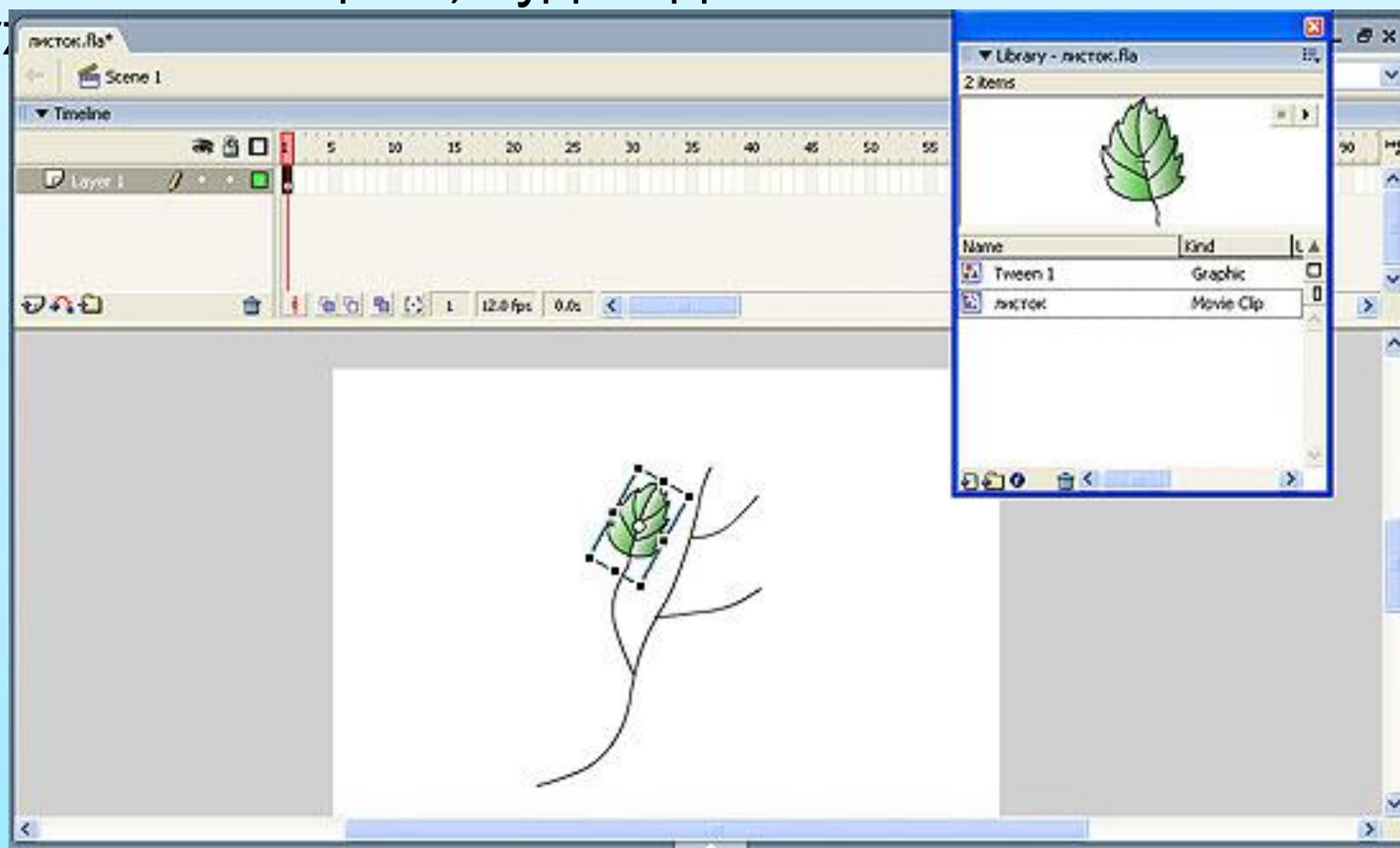
- Рассмотрим принципы работы с мувиклипами на примере создания фильма, в котором фигурирует дерево с распускающимися листочками. Создадим мувиклип «листок» по аналогии с созданием графического символа, но при этом в панели вместо пункта Graphic выберем пункт Movie Clip и дадим этому мувиклипу название «листок». В появившемся окне нарисуем листок



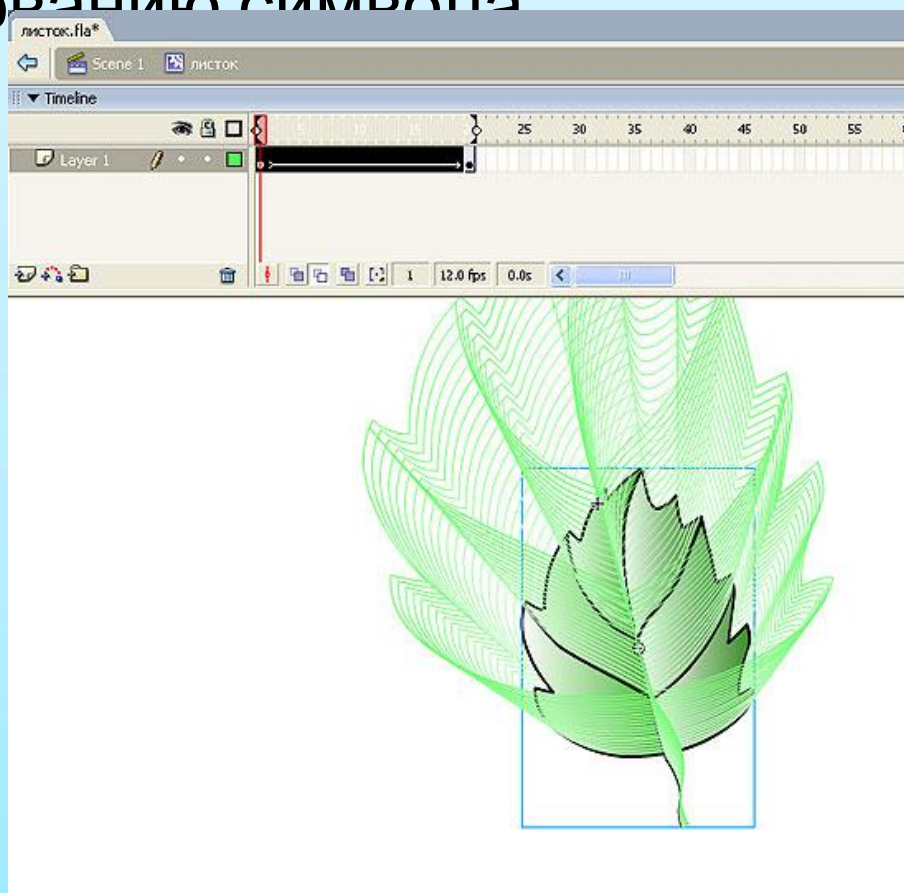
- Чтобы смоделировать рост этого листка, создадим в мувиклипе «листок» анимацию Motion Tween на 20 кадров с увеличением масштаба.



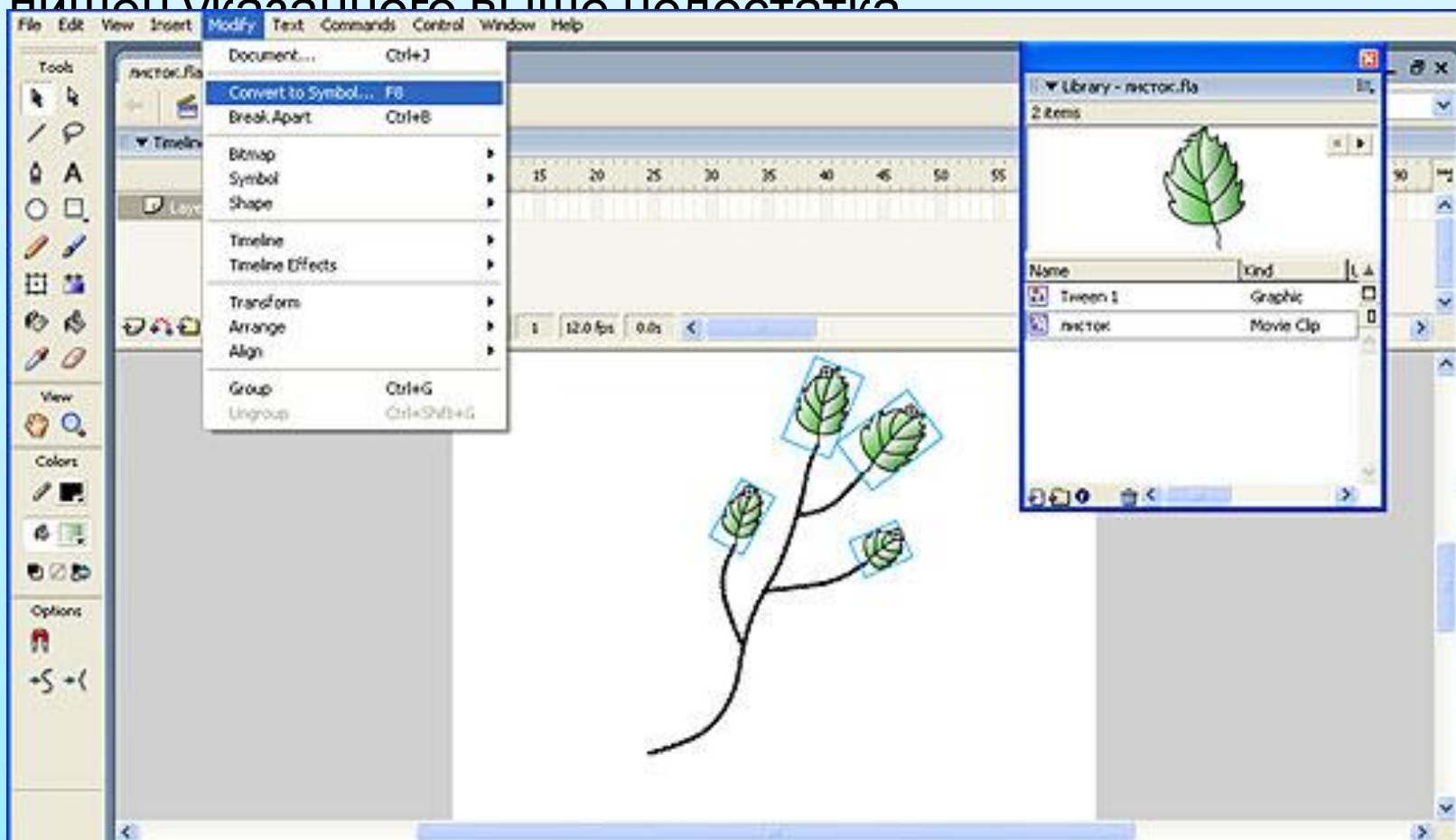
- Затем перейдем к основному документу Scene 1, нарисуем ветку, вызовем библиотеку и будем добавлять экземпляры путем перетаскивания символа на сцену. Используя функцию свободного трансформирования получаемых экземпляров символа на сцене, будем добавлять листья на ветку ну



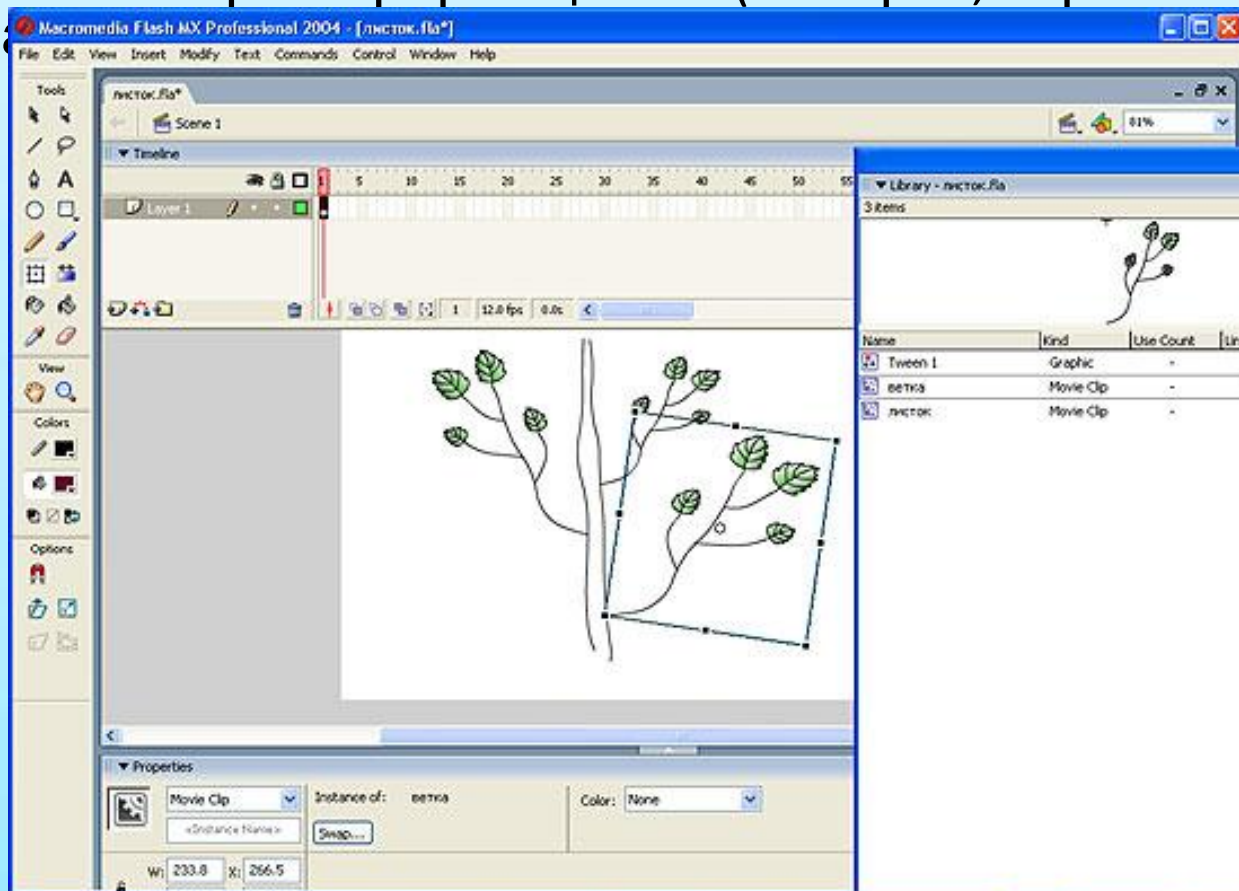
- После того как мы разместим все листья, получится фильм. В этом фильме черенки листочков в процессе роста листьев как бы отъезжают от мест крепления черенка. Для того чтобы избежать этого эффекта, вернемся к редактированию символа



- Нам необходимо, чтобы в первом и последнем кадре анимации точка прикрепления черенка листа к ветке оставалась на месте. С этой целью включим режим контурного калькирования и совместим на кальке первый и последний кадр таким образом, чтобы точка крепления листка оставалась на месте. После этого фильм будет

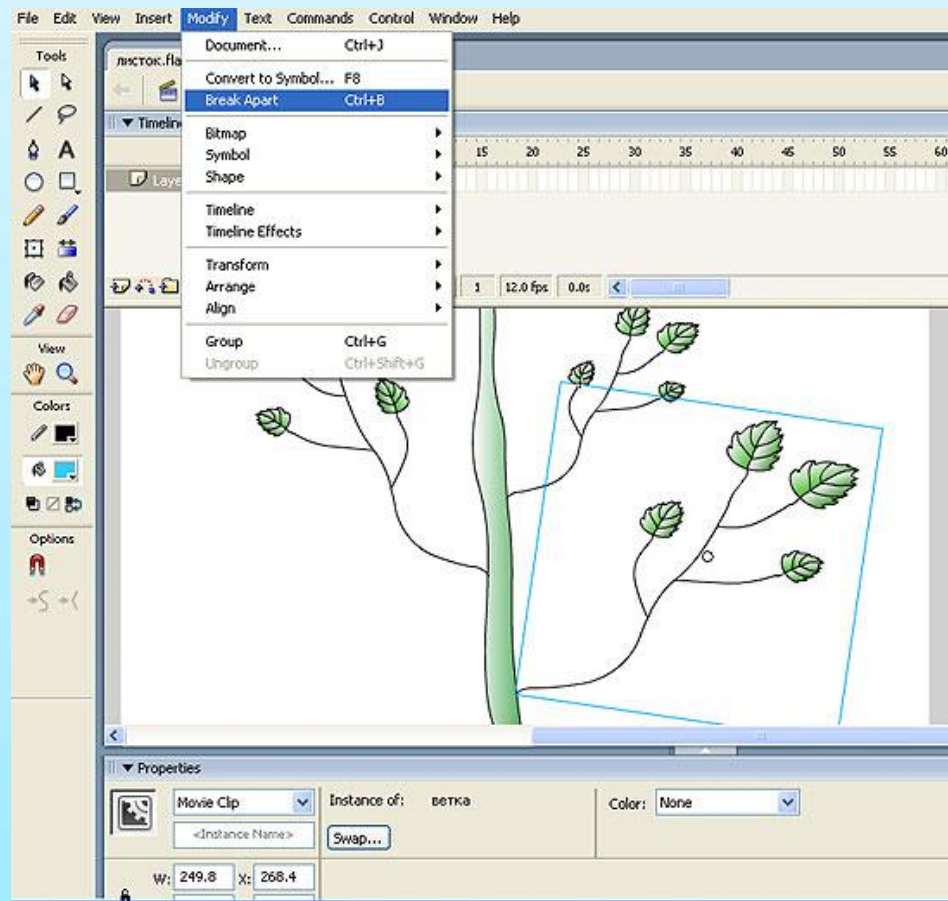


- Поскольку нашей задачей является построение анимированного дерева, которое, в свою очередь, состоит из веток, превратим созданную нами ветку в мувиклип, выполнив команду Modify => Convert to symbol. Теперь можно из совокупности клипов «ветка» построить дерево. Как видно, клип «ветка» многократно применяется с различными трансформациями (поворот, зеркальное отображение).

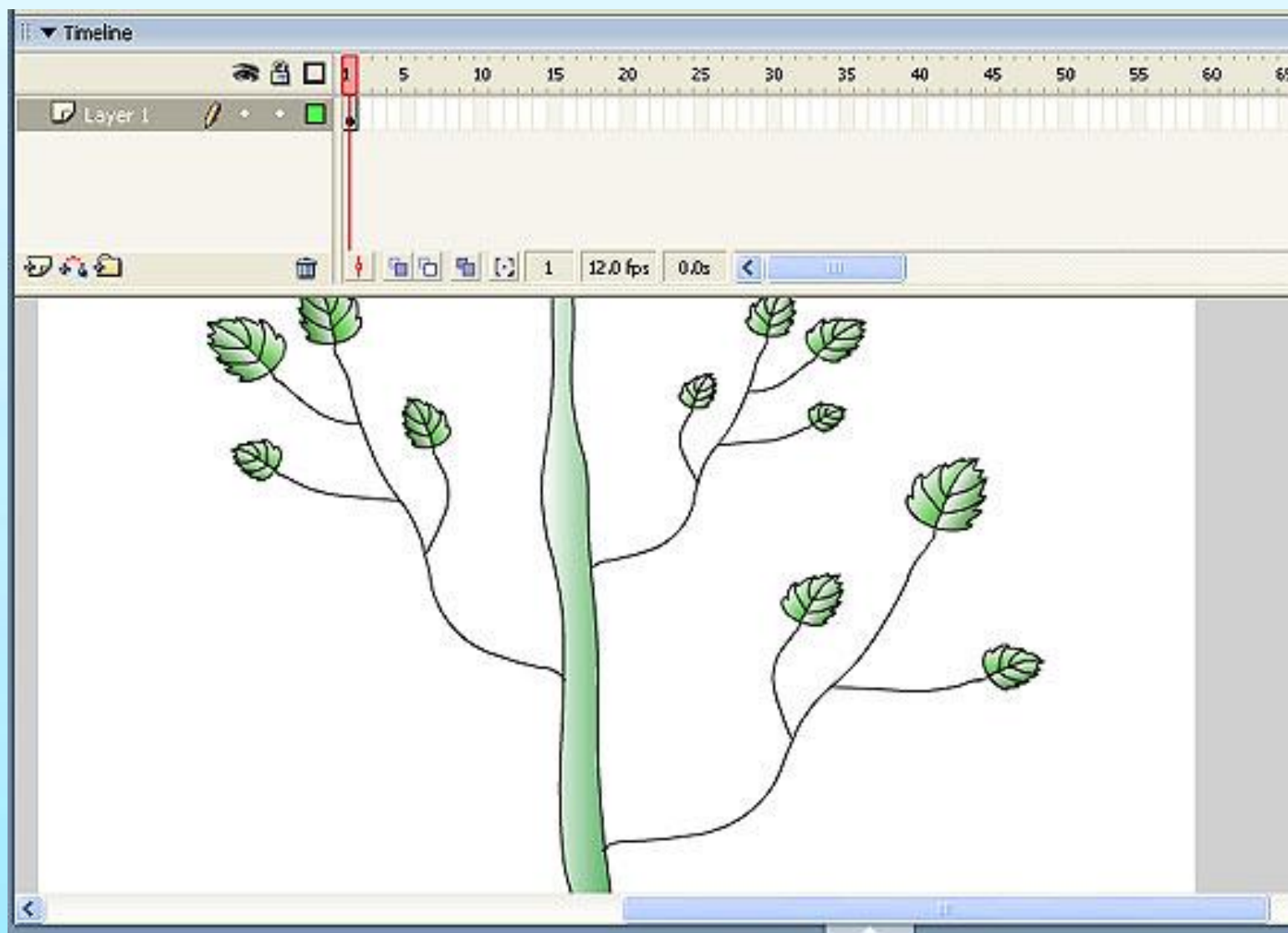




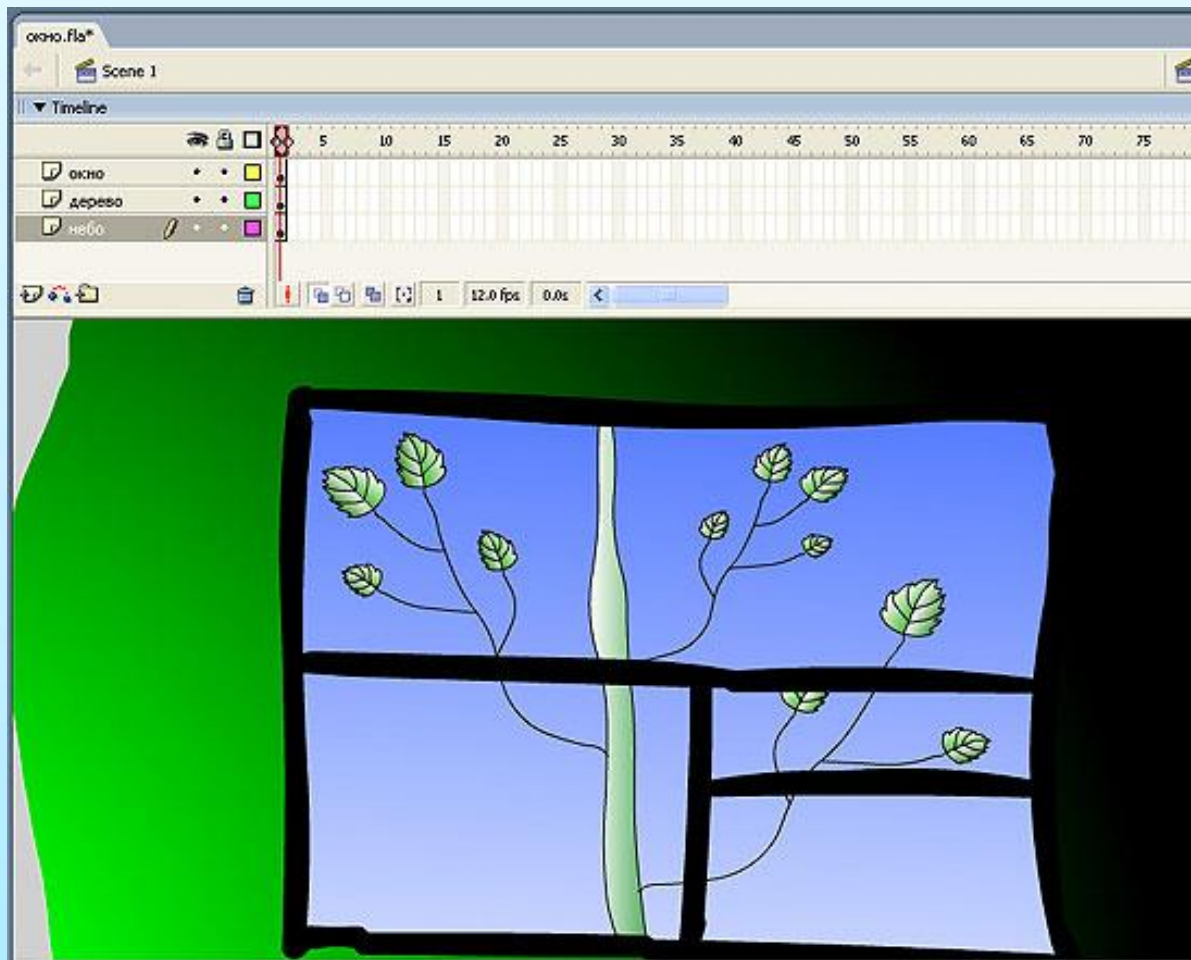
- В результате мы получим фильм, в котором изображено дерево с растущими листочками, все ветки которого отличаются друг от друга только стандартными изменениями, полученными на базе функции Free transform. Если же мы захотим поменять в какой-то ветке количество листьев, то нам необходимо будет перевести ее из экземпляра символа в редактируемый объект. Для этого выделим нужный экземпляр символа «ветка» и выполним команду Modify => Break Apart



- Теперь можно убрать один из листочков



- Чтобы придать нашему фильму некоторую законченность, добавим два новых слоя — задний план «небо» и передний план «вид из окна»

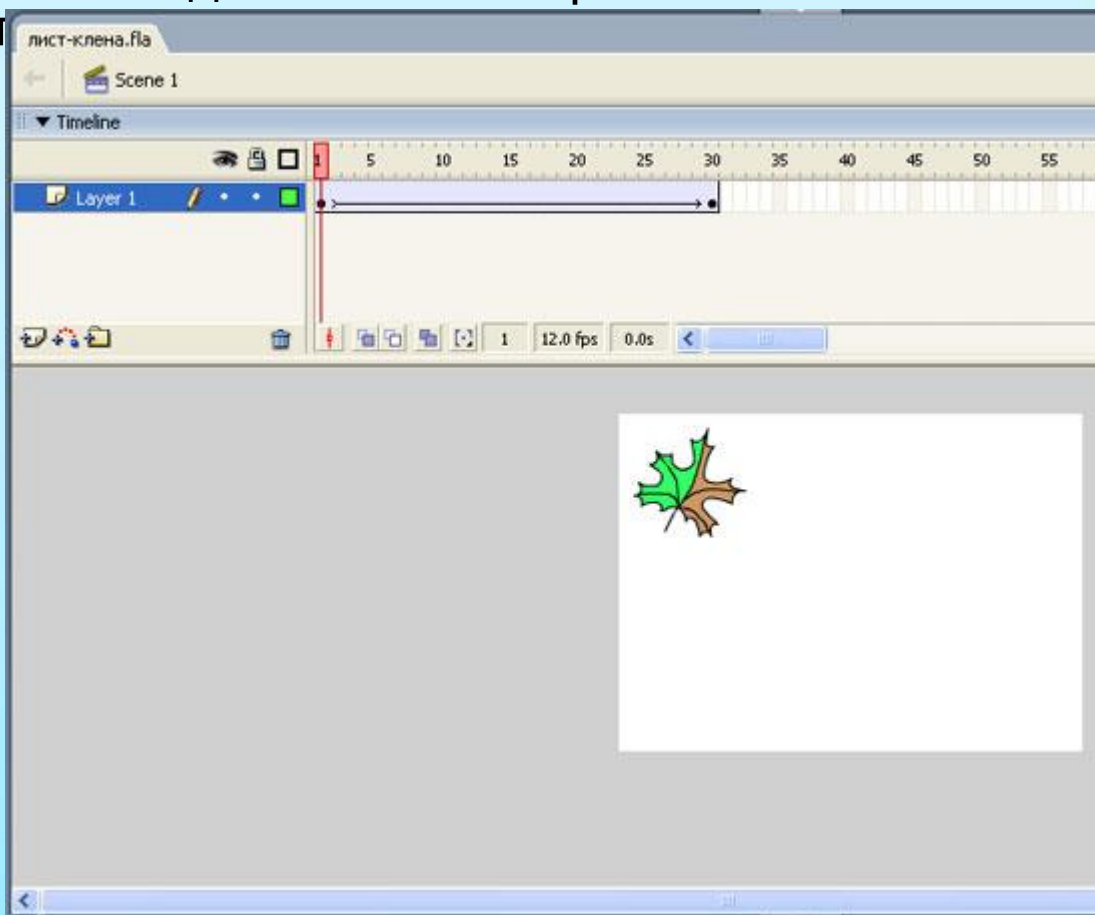


Обратите внимание, что весь фильм состоит только из одного кадра основной сцены. Вся анимация заложена в анимированных символах-мувиклипах.

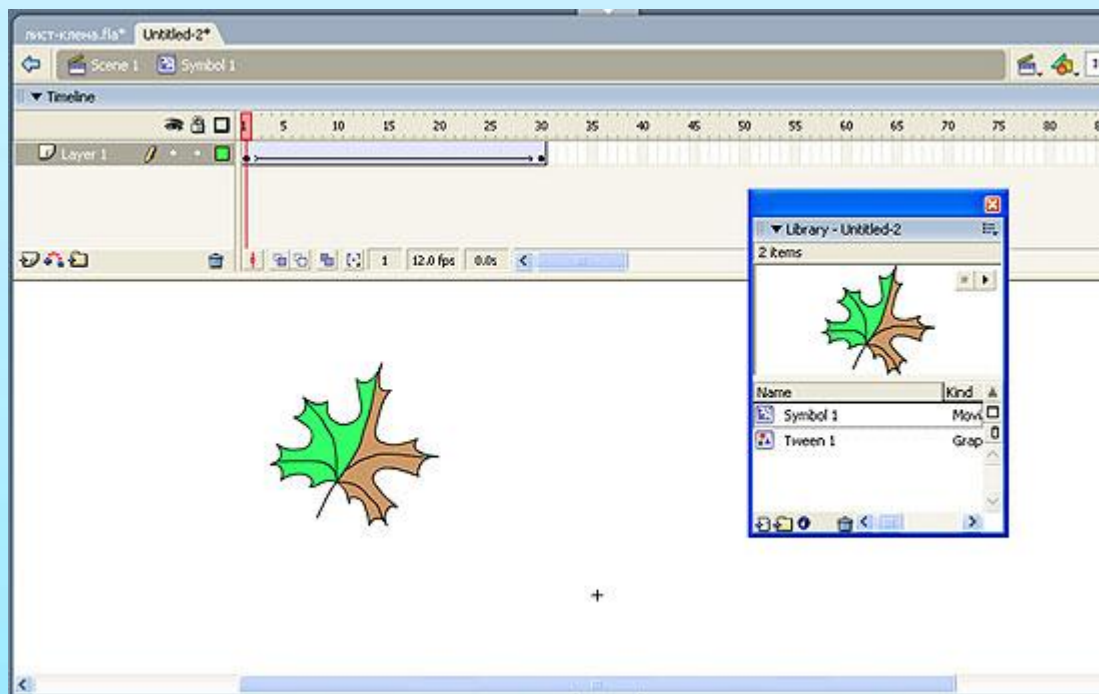
# ПРЕОБРАЗОВАНИЕ АНИМАЦИИ НА СЦЕНЕ В ОТДЕЛЬНЫЙ МУВИКЛИП

Если на сцене создана анимационная последовательность, которую необходимо повторно использовать в клипе или манипулировать ею как отдельным экземпляром, можно выделить ее и сохранить как отдельный символ-мувиклип.

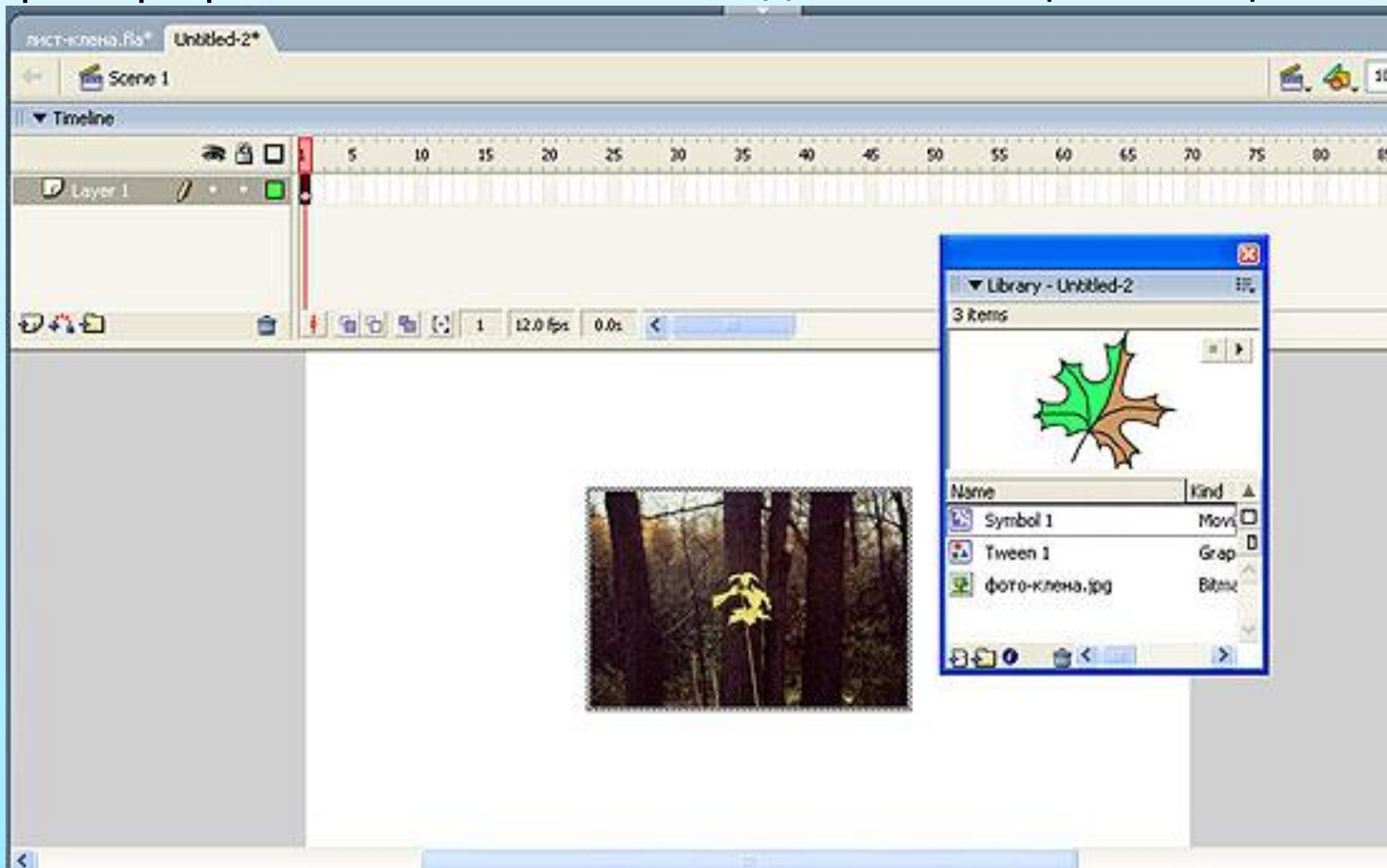
Создадим новый Flash-фильм, в котором создадим эффект падающих листьев на фоне фотографии осеннего леса. Воспользуемся фильмом, созданным на прошлом уроке



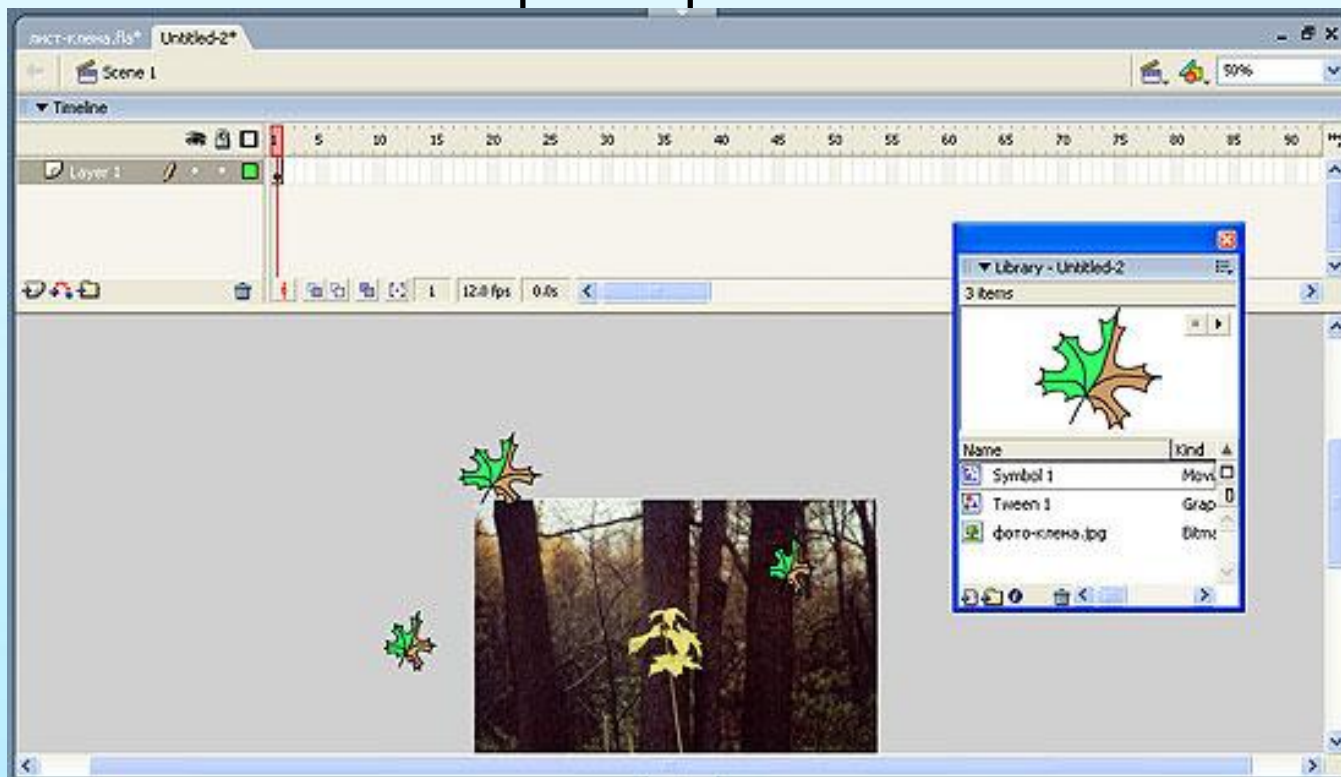
- Выделим все кадры этого фильма, щелкнув правой кнопкой мыши по любой ячейке кадра на монтажной линейке, и в выпадающем меню выберем пункт Select all frames, в результате чего все ячейки окрасятся в черный цвет. После этого правым щелчком мыши вновь вызовем меню и выберем пункт Copy Frames. Далее откроем новый документ и вставим в нем новый символ-мувиклип посредством команды Insert => New Symbol (в пункте Behavior следует выбрать Movie clip). На монтажной линейке мувиклипа щелкнем на кадре 1 и выполним команду Edit => Paste Frames.



- В библиотеке нового документа (в нашем примере его имя Untitled 2) появится мувиклип по имени symbol 1. Вернемся к сцене основного документа Scene 1 и импортируем на сцену фотографию осеннего леса командой File => Import => Import to



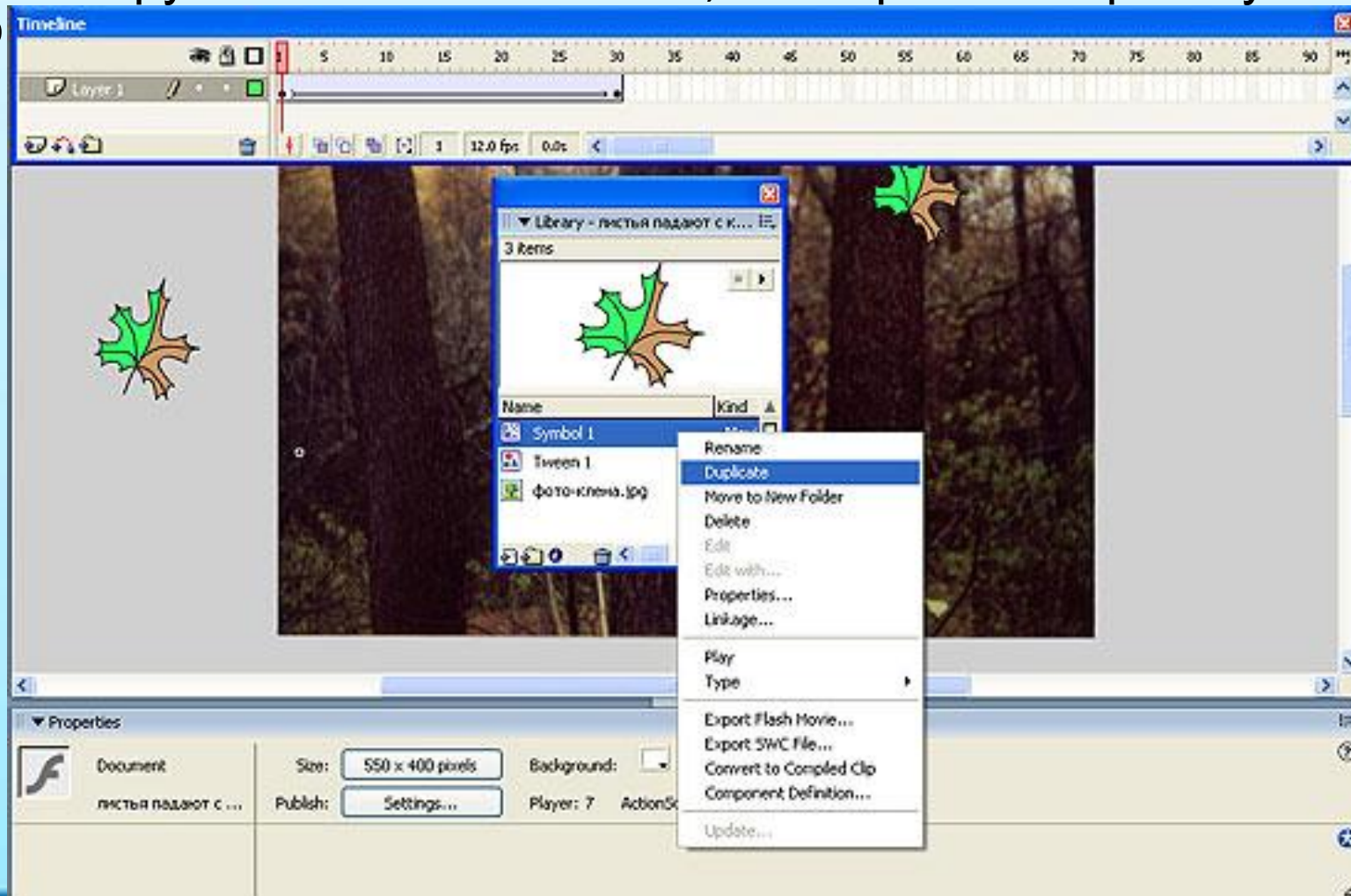
- Создадим несколько экземпляров листьев и в режиме Free transform изменим их размер



В итоге получим фильм. Поскольку мы использовали экземпляры только одного мувиклипа (хотя и модифицировали каждый из них), полученный фильм выглядит неестественно, так как все листья движутся синхронно. Поэтому желательно создать несколько символов-мувиклипов, а чтобы не создавать их с нуля, удобно воспользоваться функцией дублирования символов.

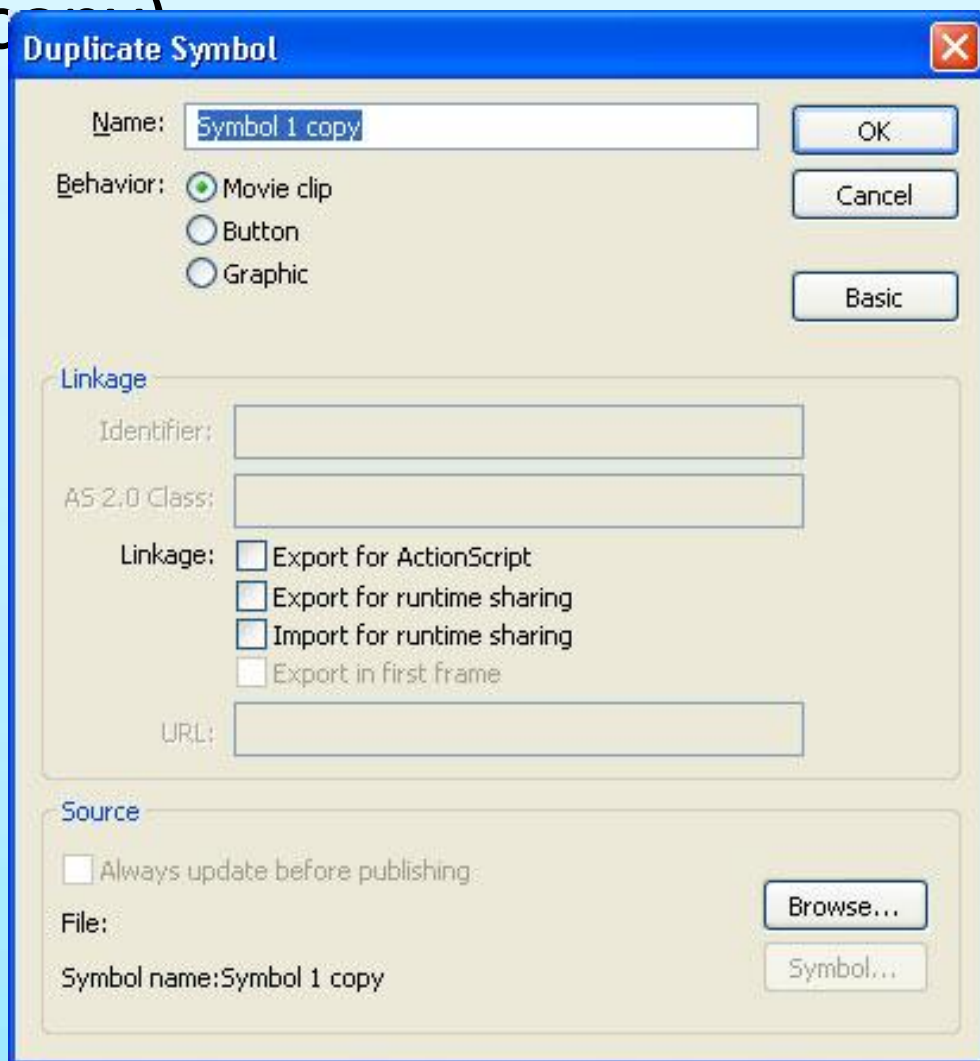
# ДУБЛИРОВАНИЕ СИМВОЛОВ

Чтобы дублировать символ, вызовем панель библиотеки по команде Window => Library и правым щелчком мыши активизируем контекстное меню, в котором выберем пункт Dup

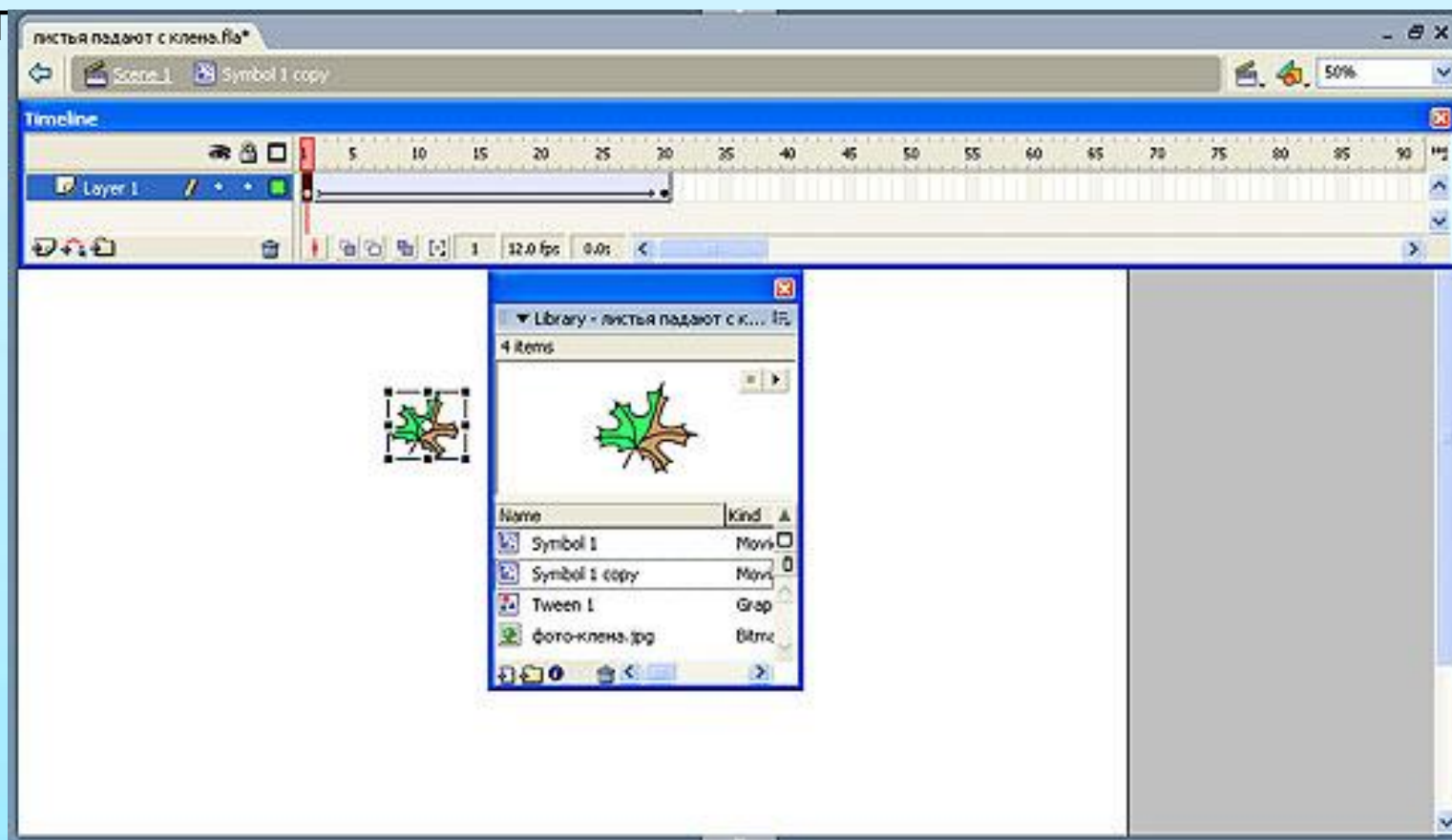




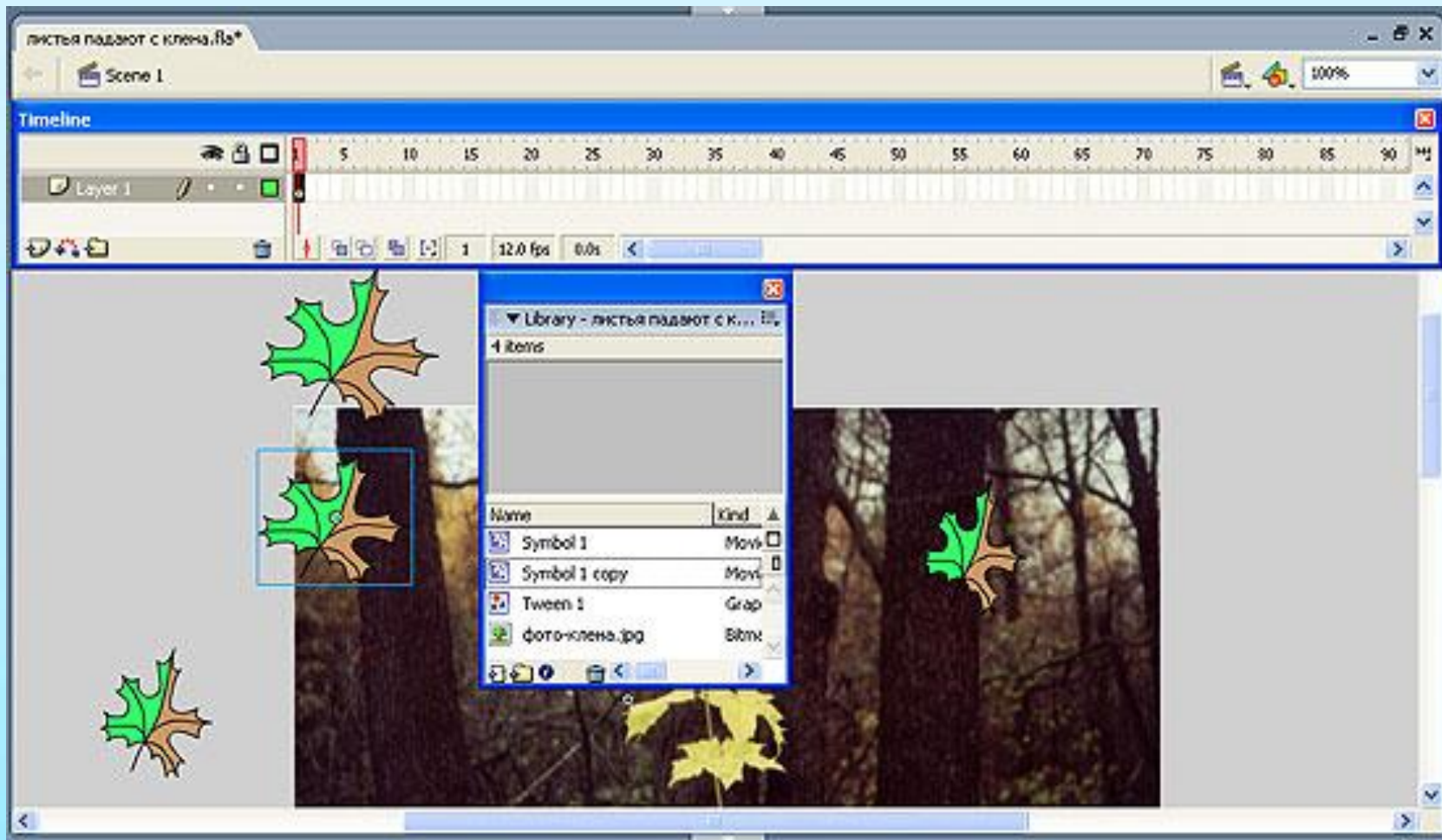
- В появившейся панели введем имя нового символа (по умолчанию это будет Symbol 1 copy)



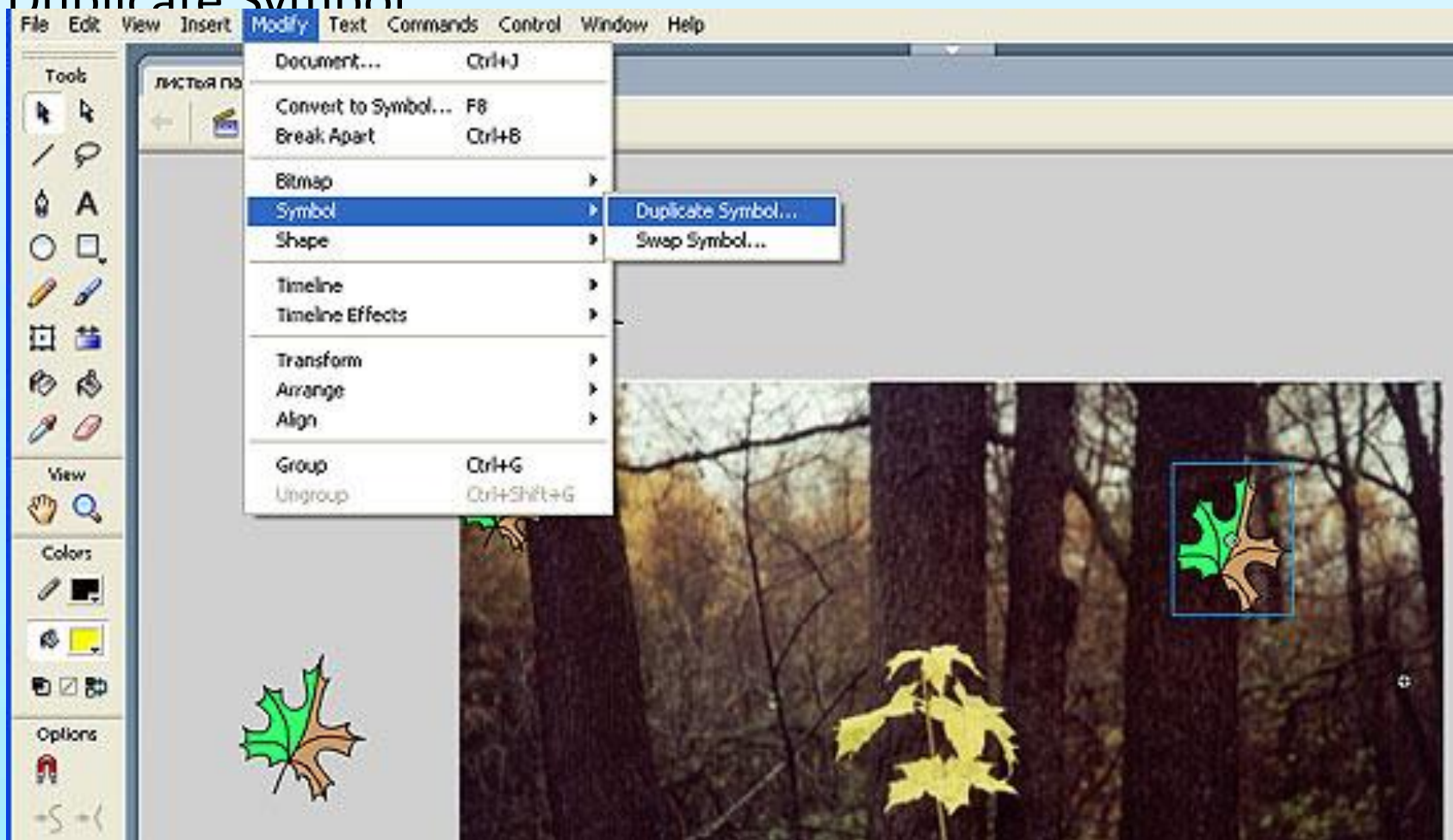
- Итак, у нас появилась копия имеющегося символа, которую мы будем использовать как заготовку для создания нового символа. Изменим в этой заготовке некоторые параметры, например замедлим скорость падения листа, для чего достаточно перетащить последний кадр вправо на монтажной линейке. А потом применим зеркальное отображение листа в последнем кадре — новый символ с именем Symbol 1 copy ГОТ



- А теперь из библиотеки добавляем на сцену экземпляры нового символа Symbol 1 сору. После того как мы добавили на сцену экземпляры второго символа, поведение листьев уже не кажется столь однообразным.



- Дублировать символ можно и с помощью использования его экземпляра. Продемонстрируем это на том же примере. Выделим экземпляр символа Symbol 1 на сцене и выполним команду Modify => Symbol => Duplicate Symbol



- После этого появится еще один символ — Symbol1 copy 2. Дважды щелкнем по имени нового символа в окне библиотеки, что вызовет окно редактирования нового символа. Поменяем положение листа в последнем кадре, перейдем к основной сцене и добавим несколько экземпляров нового символа. В результате мы получим оконч

