

## 5 jQuery эффекты и анимация

# Плавно и изящно

Взгляните, как я умею двигаться; я так грациозна. Спорим, вы так не сможете!



**Реализация всяких интересных возможностей — дело замечательное**, но если ваша страница не будет хорошо смотреться, люди не станут приходить на сайт. И здесь на первый план выходят визуальные эффекты и анимация jQuery. Вы научитесь организовывать переходы, скрывать и отображать нужные части элементов, изменять размеры элементов на странице — и все это на глазах у ваших посетителей! Вы научитесь планировать выполнение анимаций, чтобы они происходили с различными интервалами, отчего ваша страница будет выглядеть исключительно динамично.

## Новый заказ

Фирма DoodleStuff поставляет детям принадлежности для рисования. Несколько лет назад фирма открыла популярный веб-сайт с интерактивными приложениями для детей. Популярность фирмы стала расти настолько быстро, что она не успевает справляться с пожеланиями своих клиентов.

Ориентируясь на новую, более широкую аудиторию, руководитель веб-проектов хочет создать приложение, которое не потребует установки Flash или других дополнительных модулей для браузеров.

# DOODLESTUFF



Детские проекты должны быть веселыми и простыми. Сможете создать приложение для детей в возрасте от 6 до 10 лет? Не забудьте про визуальные эффекты и взаимодействие с пользователем. Только, пожалуйста, без Flash!

## Проект «Собери монстра»

Перед вами схема нового проекта, которую вам дал руководитель веб-проектов, а также переданные веб-дизайнером графические файлы.

### Проект «Собери монстра»

Развлекательное приложение «Собери монстра» адресовано детям конкретной возрастной группы. Оно позволяет «собрать» изображение монстра из 10 разных вариантов головы, глаз, носа и рта. Выбор частей монстра должен сопровождаться анимацией.

#### Пользовательский интерфейс

**Контейнер**

**Рамка**

Область головы  
*Щелкните, чтобы изменить голову монстра.*

Область глаз  
*Щелкните, чтобы изменить глаза монстра.*

Область носа  
*Щелкните, чтобы изменить нос монстра*

Область рта  
*Щелкните, чтобы изменить рот монстра.*

img - lightning1  
img - lightning2  
img - lightning3

#### Анимация

Модель изменения лица монстра



Модель анимации с молниями



*После девяти щелчков каждая полоса должна «перематываться» к началу.*

---

#### Графические файлы

*frame.png*  
ширина: 545 пикселей  
высота: 629 пикселей



*headsstrip.png* ширина: 3670 пикселей, высота: 172 пикселей



*eyessstrip.png* ширина: 3670 пикселей, высота: 79 пикселей



*nosessstrip.png* ширина: 3670 пикселей, высота: 86 пикселей



*mouthsstrip.png* ширина: 3670 пикселей, высота: 117 пикселей



*lightning\_01.jpg*



*lightning\_02.jpg*



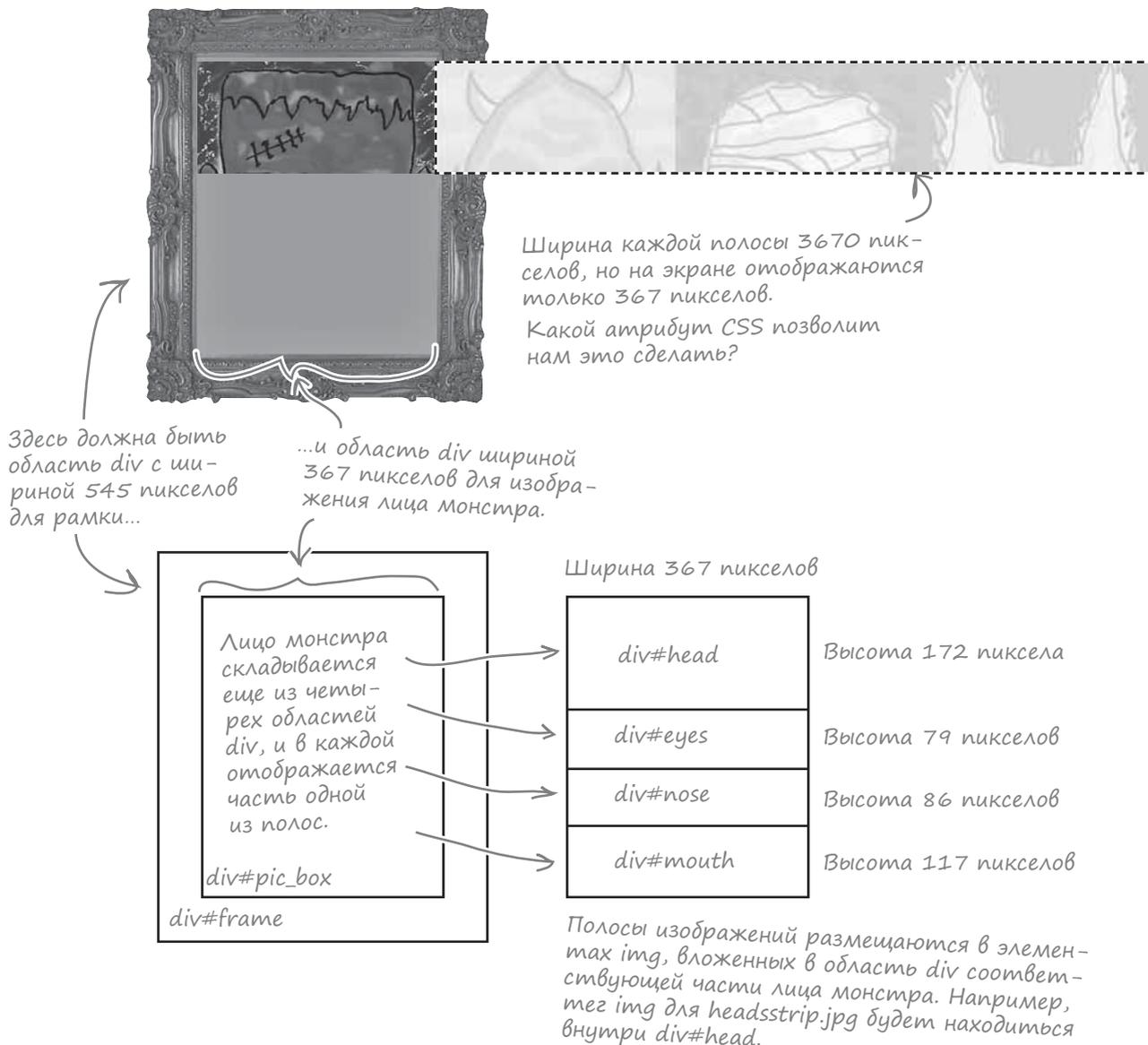
*lightning\_03.jpg*



У вас имеется подробное описание требований и необходимые графические файлы, но нет разметки HTML и кода CSS — с этого и следует начинать. Что для этого необходимо?

## Макет и позиционирование

Мы уже неоднократно говорили о том, как важно тщательно продумать структуру и стиль приложения до перехода к программированию. В данном случае это еще важнее — если не позаботиться о правильном расположении всех визуальных элементов, с эффектами и анимацией *очень скоро* начнутся проблемы. Нет ничего противнее, чем глазеть в свой код jQuery и ломать голову над тем, почему же он не работает так, как нужно вам. Лучше заранее построить эскиз приложения и поразмыслить над тем, что будет происходить на экране.





## Упражнение

Запишите в пустых местах файлов HTML и CSS идентификатор CSS, свойство или значение, обеспечивающее нужное размещение элементов приложения «Собери монстра». Если сомневаетесь, перечитайте две предыдущие страницы. Мы заполнили несколько пропусков за вас.

```
body>
<header id="top">
<p>Make your own monster face by clicking on the picture.</p></header>

<div id="frame">
  <div id="pic_box">
    <div id="....."class="face"></div>
    <div id="....."class="face"></div>
    <div id="....."class="face"></div>
    <div id="....."class="face"></div>
  </div>
</div>
<script type="text/javascript" src="scripts/jquery-1.6.2.min.js"></script>
<script type="text/javascript" src="scripts/my_scripts.js"></script>
</body>
```

index.html

```
#frame {
  position:.....
  left:100px;
  top:100px;
  width:545px;
  height:629px;
  background-image:url (images/frame.png);
  z-index: 2;
  overflow: .....
}

#pic_box{
  position: relative;
  left:91px;
  top:84px;
  height:460px;
  z-index: 1;
  overflow:.....
}

.face{
  position:.....
  left:0px;
  top:0px;
  z-index: 0;
}

#head{
  height:172px;
}

#eyes{
  .....
}

#nose{
  .....
}

#mouth{
  .....
}
```

my\_style.css



Запишите в пустых местах файлов HTML и CSS идентификатор CSS, свойство или значение, обеспечивающее нужное размещение элементов приложения Monster Mashup. Если засомневаетесь, просмотрите еще раз две предыдущие страницы. Мы заполнили несколько пропусков за вас.

```

body>
<header id="top">
<p>Make your own monster face by clicking on the picture.</p></header>

<div id="frame">
  <div id="pic_box">
    <div id="head"....class="face"></div>
    <div id="eyes".....class="face"></div>
    <div id="nose".....class="face"></div>
    <div id="mouth"!.class="face"></div>
  </div>
</div>
<script type="text/javascript" src="scripts/jquery-1.6.2.min.js"></script>
<script type="text/javascript" src="scripts/my_scripts.js"></script>
</body>

```



index.html

```

#frame {
  position: absolute;
  left:100px;
  top:100px;
  width:545px;
  height:629px;
  background-image:url (images/frame.png) ;
  z-index: 2;
  overflow: hidden;
}

#pic_box{
  position: relative;
  left:91px;
  top:84px;
  width:367px;
  height:460px;
  z-index: 1;
  overflow: hidden;
}

#head{
  height:172px;
}

#eyes{
  height:79px;
}

#nose{
  height:86px;
}

#mouth{
  height:117px;
}

.face{
  position: relative;
  left:0px;
  top:0px;
  z-index: 0;
}

```

При анимации позиции элемента используется абсолютное или относительное позиционирование.

Присваивание overflow значения hidden позволяет скрыть часть полосы изображения, выходящую за границы области pic\_box.

Для этого также можно воспользоваться свойством CSS clip.



my\_style.css

## Еще немного структуры и стиля

Далее необходимо разобраться со структурными изменениями в файлах HTML и CSS. Включите приведенный ниже код в файлы `index.html` и `my_style.css`. Графические файлы можно загрузить по адресу [www.thinkjquery.com/chapter05](http://www.thinkjquery.com/chapter05).



Добавляем контейнер и вкладываем в него изображение с молниями.

```
<div id="container">
  
  
  
  <div id="frame">
    <div id="pic_box">
      <div id="head" class="face"></div>
      <div id="eyes" class="face"></div>
      <div id="nose" class="face"></div>
      <div id="mouth" class="face"></div>
    </div>
  </div>
</div>
```



index.html

```
#container{
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: 0;
}

.lightning{
  display: none;
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: 0;
}

body{
  background-color: #000000;
}

p{
  color: #33FF66;
  font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif;
  font-size: 12px;
}

#text_top {
  position: relative;
  z-index: 4;
}
```

В исходном состоянии молнии должны быть невидимы.

Чтобы использовать анимацию для элемента, следует задать его свойству `position` значение `absolute`, `fixed` или `relative`.



my\_style.css

## Проработка интерфейса

Итак, мы разобрались с визуальным макетом приложения «Собери монстра». Давайте проработаем интерактивную часть пользовательского интерфейса, представленную в эскизе. Нам предстоит запрограммировать реакцию разных элементов страницы на щелчки. Собственно, этим мы занимаемся уже четыре главы, так что сейчас это уже пара пустяков.



### Часто задаваемые вопросы

**В:** Что это за свойство CSS `position`? Почему оно необходимо для анимации и эффектов jQuery?

**О:** Свойство CSS `position` управляет тем, где и как браузерный движок размещает элементы. Многие эффекты jQuery реализуются с использованием свойства `position`. Если вы забыли, как работает это свойство, обращайтесь к превосходному объяснению в центре для разработчиков Mozilla:

[http://developer.mozilla.org/en/CSS/position#Relative\\_positioning](http://developer.mozilla.org/en/CSS/position#Relative_positioning)

**В:** Почему для анимации элементов свойству CSS `position` необходимо задать значение `absolute`, `fixed` или `relative`?

**О:** Если оставить свойству CSS `position` значение по умолчанию (т. е. `static`), то для элемента будет невозможно задать позицию сторон (`top`, `right`, `left` и `bottom`). При использовании функции `animate` возможность изменения этих позиций необходима, а в режиме `static` это невозможно. Со значениями `absolute`, `fixed` и `relative` — такой проблемы нет.

**В:** Вы упомянули какой-то «браузерный движок». А это что такое?

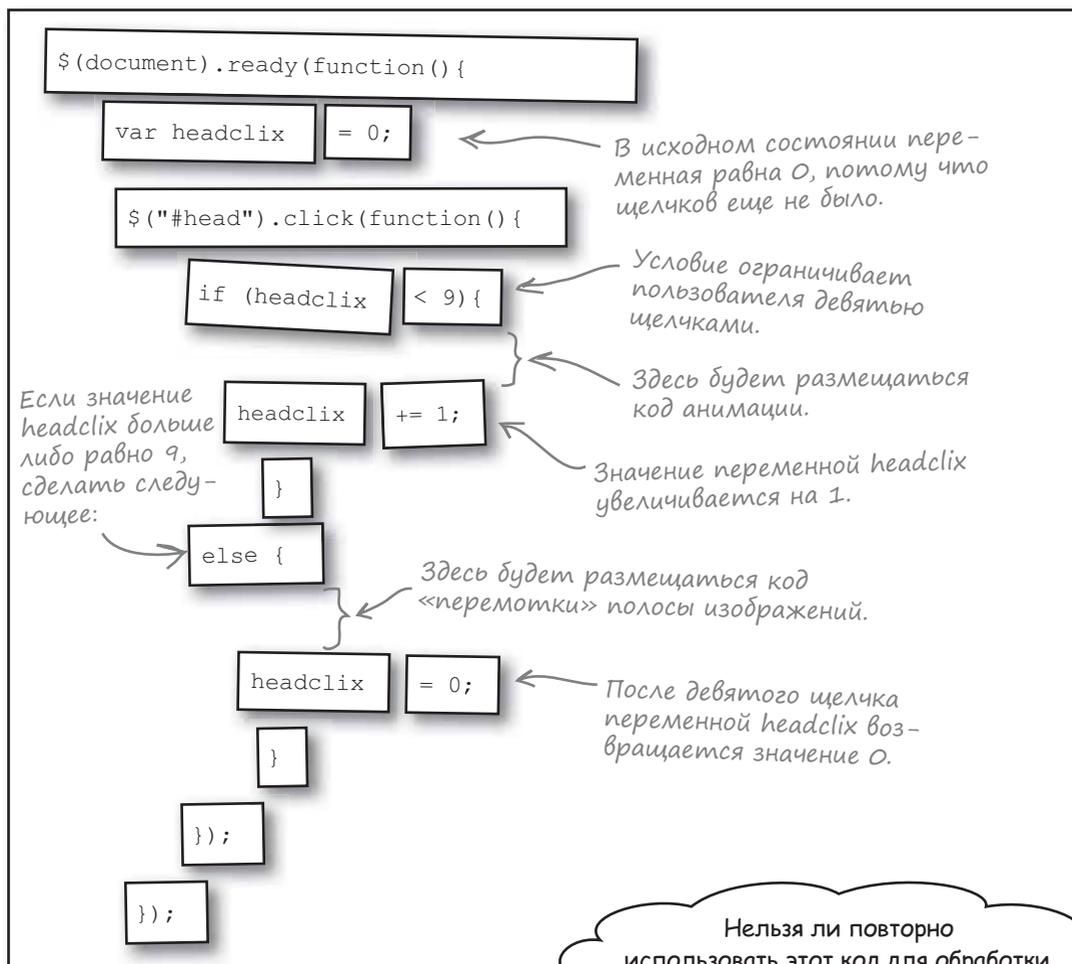
**О:** Браузерный движок визуализации — один из важнейших компонентов браузера, который интерпретирует разметку HTML и код CSS и отображает результат в окне просмотра браузера. Google Chrome и Safari используют движок визуализации Webkit; Firefox использует Gecko, а Microsoft Internet Explorer — движок, который называется Trident.





## Развлечения с магнитами. Решение

Расставьте магниты в правильном порядке, чтобы элемент `div#head` реагировал на щелчки. Проследите за тем, чтобы переменные и условные конструкции следовали в правильном порядке для обнаружения девятого щелчка.



Нельзя ли повторно использовать этот код для обработки щелчков на других областях?

### Конечно, можно!

Все элементы работают по той же схеме, что и элемент `div#head` (с небольшими изменениями вроде имени переменной).

