Najlepszy podręcznik do jQuery!

Rusz głową!

jQuery

Szybko implementuj złożone formularze HTML

Buduj aplikacje internetowe bez stosowania irytujących wtyczek

Twórz własne, niestandardowe funkcje jQuery

Dodawaj interaktywność do swoich stron w zaledwie kilku wierszach kodu

Korzystaj z animacji i Ajaksa na swoich stronach

O'REILLY®

Ryan Benedetti, Ronan Cranley

Helion

Tytuł oryginału: Head First jQuery

Tłumaczenie: Ireneusz Jakóbik

ISBN: 978-83-246-3864-2

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

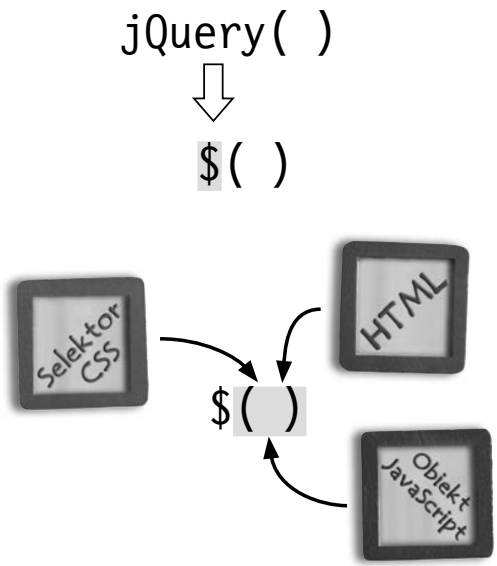
Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/jquerg
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

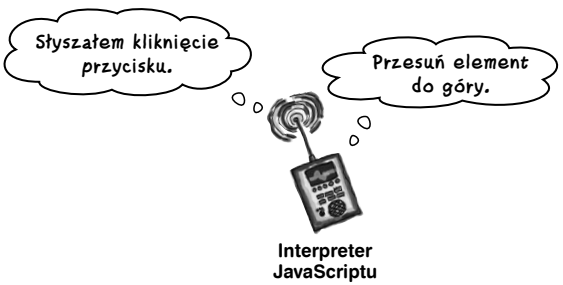
Printed in Poland.

1 Zaczynamy z jQuery

Akcja na stronach internetowych

Chcesz więcej dla swoich stron. Masz w zanadrzu HTML oraz CSS i chciałbyś wzbogacić swoje umiejętności o pisanie skryptów, ale nie masz zamiaru spędzić całego życia, pisząc je wiersz po wierszu. Potrzebujesz biblioteki skryptów, która pozwoli Ci zmieniać strony internetowe w locie. A skoro wyrażamy już nasze życzenia, to może dodać jeszcze współpracę z Ajaksem i PHP? Czy da się zrobić w trzech wierszach to, do czego większość języków po stronie klienta potrzebuje piętnastu? Pobożne życzenia? W żadnym wypadku! Musisz zapoznać się z jQuery.

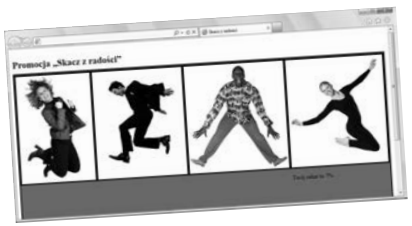




2 Selektory i metody

Chwytaj i w drogę

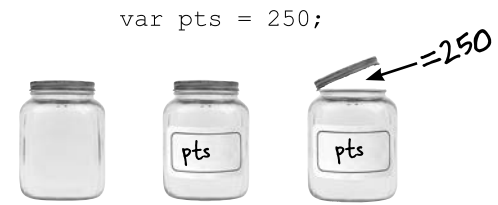
jQuery pomaga chwytać elementy strony internetowej i robić z nimi różne rzeczy. W tym rozdziale zagłębimy się w selektory i metody jQuery. Dzięki selektorom możemy chwytać elementy na stronie, a metody pozwalają na robienie z nimi rozmaitych sztuczek. Podobnie jak wielka księga zaklęć magicznych, biblioteka jQuery pozwala nam zmieniać masę rzeczy w locie. Możemy sprawiać, że obrazy rozpływają się w powietrzu i pojawiają znikąd. Możemy wybrać określony fragment tekstu i w animowany sposób zmienić rozmiar jego czcionki. Rozpocznijmy więc pokaz — złapmy kilka elementów strony i do roboty!



3 Zdarzenia i funkcje jQuery

Niech się coś dzieje na Twojej stronie

jQuery ułatwia dodawanie akcji oraz interaktywności do każdej strony internetowej. W tym rozdziale przyjrzymy się temu, jak sprawić, żeby strona reagowała, kiedy użytkownicy wchodzą z nią w interakcję. Sprawienie, że kod będzie reagował na działania użytkownika, wyniesie Twoją witrynę na zupełnie nowy poziom. Zajmiemy się też tworzeniem funkcji wielokrotnego użycia, dzięki czemu kod napiszesz raz, ale już korzystać z niego będziesz wielokrotnie.



4 Manipulowanie stroną internetową za pomocą jQuery

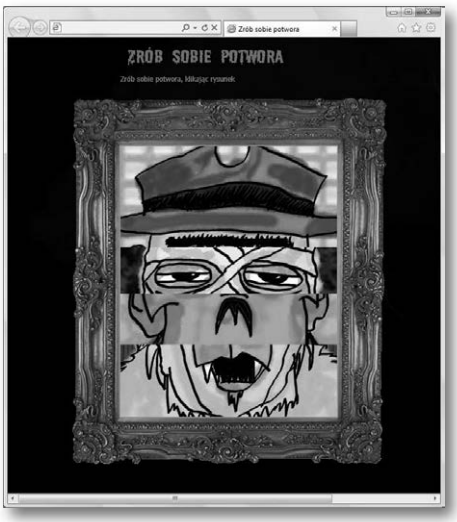
Zmodyfikuj drzewo DOM

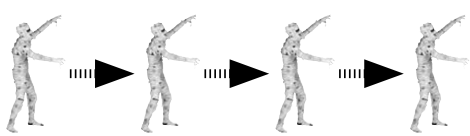
To, że strona skończyła się pobierać, nie znaczy od razu, że musi ona zachowywać tę samą strukturę. W rozdziale 1. zobaczyliśmy, jak podczas wczytywania strony, w celu ustalenia jej struktury, jest konstruowane drzewo DOM. W bieżącym rozdziale dowiemy się, jak poruszać się w górę i w dół struktury drzewa DOM i jak pracować z hierarchią elementów oraz relacjami rodzic – dziecko, aby za pomocą jQuery zmieniać w locie strukturę strony.

5 Efekty i animacja w jQuery

Nieco wdzięku na Twojej stronie

Sprawianie, że na Twojej stronie będą się dziać różne rzeczy, to niezły pomysł, ale tylko do czasu. Jeśli nie będzie ona wyglądać atrakcyjnie, ludzie nie zechcą z niej korzystać. Dlatego właśnie przydadzą Ci się efekty i animacja w jQuery. W bieżącym rozdziale nauczysz się, co zrobić, aby wraz z upływem czasu elementy zmieniały swoje miejsce na stronie, jak pokazać lub ukryć określony element albo jak go powiększyć lub zmniejszyć, a wszystko to na oczach użytkownika. Dowiesz się też, jak zakładać harmonogramy animacji, aby uruchamiały się one w różnych odstępach czasu, co nada Twojej stronie bardzo dynamiczny wygląd.

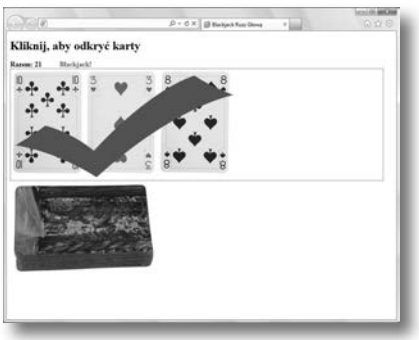


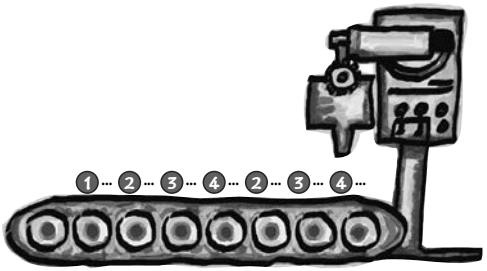


jQuery i JavaScript

6 ~~Luke~~ jQuery, jestem twoim ojcem!

Sama biblioteka jQuery nie da sobie ze wszystkim rady. Chociaż jest ona biblioteką JavaScriptu, nie potrafi niestety robić wszystkiego, co jej język ojczysty. W tym rozdziale przyjrzymy się niektórym aspektom JavaScriptu, których będziesz potrzebował do tworzenia naprawdę atrakcyjnych stron, oraz temu, jak można z nich korzystać w jQuery w celu tworzenia niestandardowych list i obiektów, a także jak przechodzić przez nie w pętlach, aby ułatwić sobie życie.

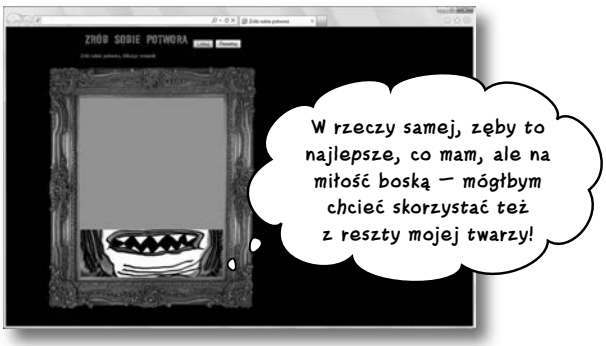


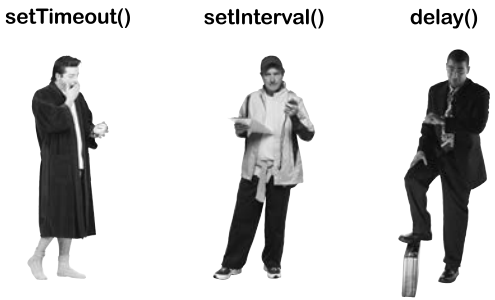


7 Niestandardowe funkcje dla niestandardowych efektów

Co ostatnio dla mnie zrobiłeś?

Kiedy połączysz niestandardowe efekty jQuery z funkcjami JavaScriptu, możesz sprawić, że Twój kod — a tym samym Twoja aplikacja internetowa — będzie wydajniejszy, skuteczniejszy i *lepszy*. W tym rozdziale zagłębisz się jeszcze bardziej w ulepszanie efektów jQuery poprzez obsługę **zdarzeń przeglądarki**, pracę z **funkcjami czasu** oraz poprawę **organizacji i możliwości wielokrotnego użycia** Twoich niestandardowych funkcji w JavaScripcie.





8 jQuery i Ajax

Proszę podać dane

Używanie jQuery w celu robienia sprytnych sztuczek z CSS i drzewem DOM to fajna sprawa, ale już wkrótce będziesz musiał odczytywać informacje (albo dane) z serwera i je wyświetlać. Może nawet przyjdzie Ci zaktualizować fragmenty strony informacjami z serwera bez potrzeby jej odświeżania. Poznaj Ajax. W połączeniu z jQuery i JavaScriptem może on właśnie coś takiego robić. W tym rozdziale dowiemy się, jak jQuery radzi sobie z namawianiem Ajaksa do wysyłania zapytań do serwera i co potem robi z otrzymanymi informacjami.

9 Obsługa danych JSON

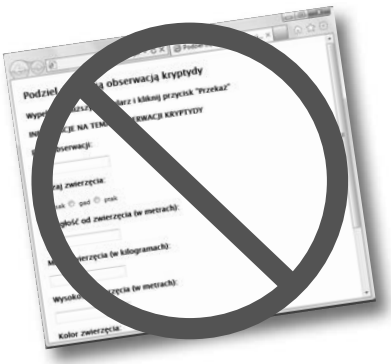
Kliencie, oto serwer

Chociaż czytanie danych z pliku XML jest bardzo przydatne, to jednak nie zawsze wystarcza. Bardziej interaktywny format wymiany danych (JavaScript Object Notation, znany też jako JSON) ułatwia pobieranie danych z serwera. JSON jest też łatwiejszy do wygenerowania i czytania niż format XML. Używając jQuery, PHP oraz SQL-a, nauczysz się, jak utworzyć bazę danych przechowującą informacje, które będziesz mógł później odczytać za pomocą formatu JSON i wyświetlić na ekranie przy użyciu jQuery. Tak objawia się prawdziwa moc aplikacji internetowych!

10 Interfejs użytkownika w jQuery

Diametralna zmiana wyglądu

Sieć WWW żyje i umiera dzięki swoim użytkownikom i ich danym. Zbieranie danych od użytkowników to poważne zajęcie, które może być czasochłonnym wyzwaniem dla programisty. Widziałeś już, jak Ajax, PHP i MySQL mogą pomóc w czynieniu efektywniejszym działania aplikacji internetowych. Teraz przyjrzymy się, jak jQuery może nam pomóc w utworzeniu interfejsów użytkownika dla formularzy zbierających dane. Po drodze otrzymasz sporą dawkę jQuery UI — oficjalnej biblioteki interfejsu użytkownika języka jQuery.



11

jQuery i API

Obiekty, wszędzie obiekty

Niezależnie od tego, jak bardzo utalentowanym programistą jesteś, sam nie dasz sobie rady ze wszystkim... Zobaczyliśmy, jak można dołączać wtyczki jQuery, takie jak jQuery UI, albo nawigację z wykorzystaniem kart, aby bez większego wysiłku wzbogacać programy pisane w jQuery. W celu wyniesienia naszych aplikacji na jeszcze wyższy poziom, dodania do nich paru z naprawdę niezłych narzędzi dostępnych w Internecie oraz skorzystania z informacji udostępnionych przez prawdziwych potentatów — takich jak Google, Twitter albo Yahoo! — będziemy jednak potrzebować czegoś... więcej. Firmy te, i wiele innych, dostarczają API (*application programming interfaces*, czyli interfejsy programowania aplikacji) dedykowane dla swoich usług, z których możesz korzystać w swojej witrynie. W tym rozdziale przyjrzymy się podstawom API i zastosujemy niezwykle popularny interfejs, jakim jest Google Maps API.

A

Ostatki

Dziesięć najważniejszych rzeczy (o których nie napisaliśmy)

Nawet po tym wszystkim ciągle jest jeszcze wiele rzeczy, do których nie dotarliśmy. Istnieje mnóstwo dobrodziejstw jQuery i JavaScriptu, których nie udało nam się wcisnąć do tej książki. Byłoby nieuczciwością nie wspomnieć o nich, a tak będziesz lepiej przygotowany na każdy inny aspekt jQuery, który możesz napotkać podczas swoich podróży.

B

Konfiguracja środowiska programistycznego

Przygotuj się do przyszłych sukcesów

Potrzebujesz miejsca na ćwiczenie nowo nabytych umiejętności PHP bez narażania swoich danych na niebezpieczeństwa czyhające w sieci. Posiadanie bezpiecznego miejsca na opracowywanie aplikacji PHP przed wypuszczeniem jej w szeroki świat sieci internetowej zawsze jest dobrym pomysłem. Dodatek ten zawiera instrukcje dotyczące instalacji serwera WWW, MySQL i PHP, dzięki czemu będziesz mieć do dyspozycji bezpieczne miejsce do pracy i ćwiczeń.

Nieco wdzięku na Twojej stronie

Sprawianie, że na Twojej stronie będą się dziać różne rzeczy, to niezły pomysł, ale tylko do czasu. Jeśli nie będzie ona wyglądać atrakcyjnie, ludzie nie zechcą z niej korzystać. Dlatego właśnie przydadzą Ci się efekty i animacja w jQuery. W bieżącym rozdziale nauczysz się, co zrobić, aby wraz z upływem czasu elementy zmieniały swoje miejsce na stronie, jak pokazać lub ukryć określony element albo jak go powiększyć lub zmniejszyć, a wszystko to na oczach użytkownika. Dowiesz się też, jak zakładać harmonogramy animacji, aby uruchamiały się one w różnych odstępach czasu, co nada Twojej stronie bardzo dynamiczny wygląd.

Gryzmołki potrzebują aplikacji sieciowej

Gryzmołki zaopatrują webowickie dzieci w atrakcyjne materiały artystyczne do zabawy. Kilka lat temu firma uruchomiła popularną witrynę internetową, w której udostępniła interaktywne aplikacje dla dzieci. Społeczność jej fanów rośnie w takim tempie, że Gryzmołki zaczęły mieć problemy z radzeniem sobie na bieżąco z napływającymi prośbami.

Aby zaspokoić potrzeby nowej, większej klienteli Gryzmołków, dyrektor ds. projektów internetowych chciałaby utworzyć aplikację, która nie korzystałaby z Flasha ani z żadnych innych wtyczek przeglądarkowych.

Zrób sobie potwora

Oto plan aplikacji przekazany przez dyrektor ds. projektów internetowych wraz z plikami graficznymi od projektanta przeznaczonymi do użycia w programie:

Projekt Zrób sobie potwora

Zadaniem aplikacji Zrób sobie potwora jest zabawianie dzieci w docelowej grupie wiekowej przez umożliwienie im zrobienia sobie potwora dzięki mieszaniu 10 różnych głów, oczu, nosów i ust. Przejścia między poszczególnymi częściami twarzy potwora powinny być animowane.

Interfejs użytkownika

Pojemnik

Ramka

Obszar głowy
Kliknij, aby przewinąć głowę potwora.

Obszar oczu
Kliknij, aby przewinąć oczy potwora.

Obszar nosa
Kliknij, aby przewinąć nos potwora.

Obszar ust
Kliknij, aby przewinąć usta potwora.

img - lightning1
img - lightning2
img - lightning3

Po dziewięciu kliknięciach każdy pasek powinien „przewinąć się" na początek.

Animacja

Symulacja obrazująca, jak powinna się zmieniać część twarzy potwora.

Symulacja obrazująca, jak powinna wyglądać animacja błyskawicy.

Obrazy błyskawicy powinny pojawiać się i znikać szybko, tak jakby rzeczywiście się błyskało.

Pliki graficzne

frame.png
szerokość: 545 pikseli
wysokość: 629 pikseli

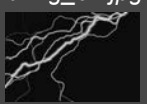
headsstrip.png ***szerokość: 3670 pikseli, wysokość: 172 piksele***

eyessstrip.png ***szerokość: 3670 pikseli, wysokość: 79 pikseli***

nosessstrip.png ***szerokość: 3670 pikseli, wysokość: 86 pikseli***

mouthsstrip.png ***szerokość: 3670 pikseli, wysokość: 117 pikseli***

lightning_01.jpg



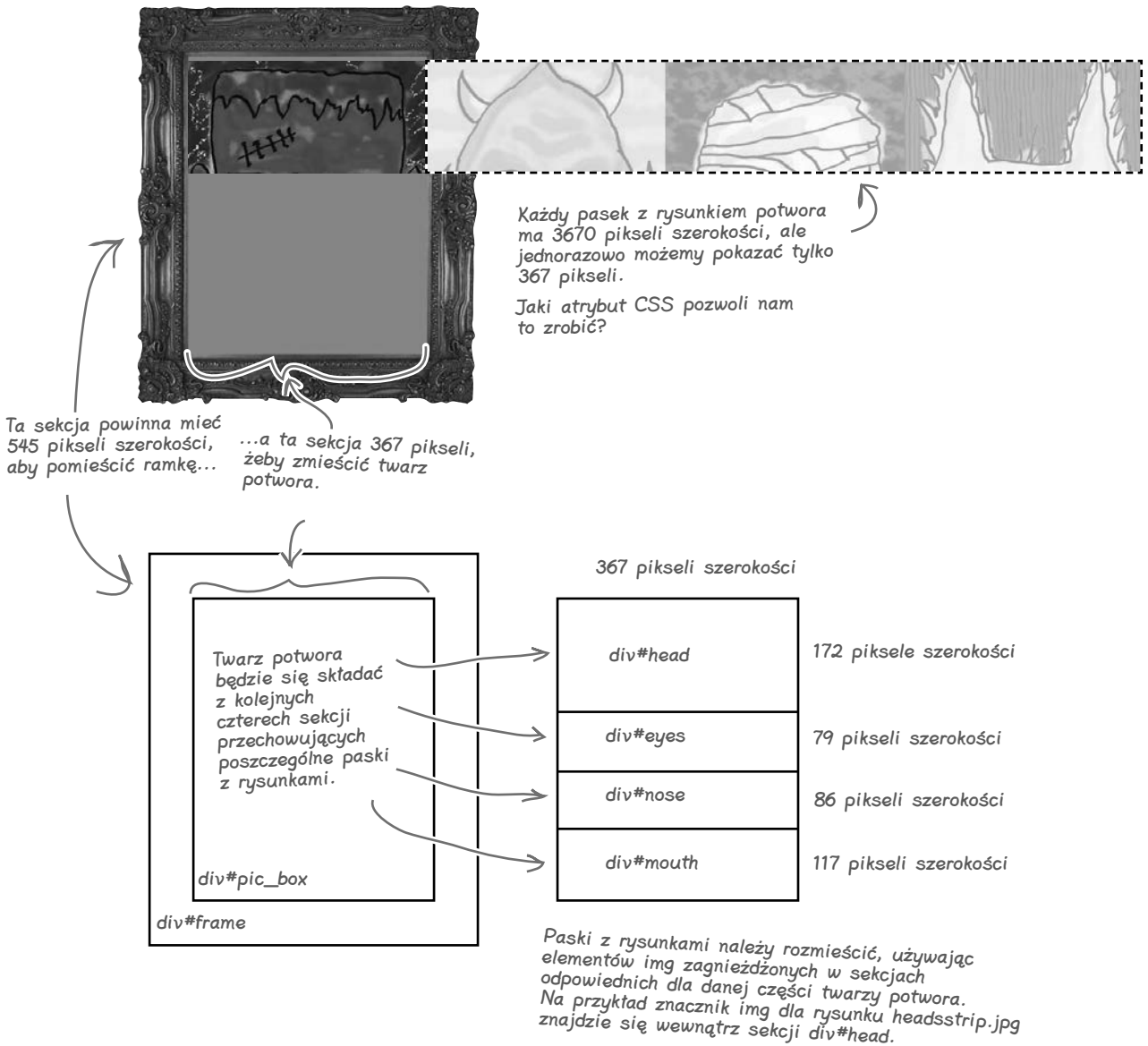
lightning_02.jpg

lightning_03.jpg

Masz mnóstwo szczegółów dotyczących wymagań projektu oraz potrzebne pliki graficzne, ale grafik nie napisał żadnego HTML-a ani CSS, a od tego właśnie musisz zacząć. Co powinieneś zrobić, aby utworzyć i skonfigurować te pliki?

Zrób sobie potwora — potrzebujemy układu i położenia na stronie

Bez wątpienia sporo mówiliśmy do tej pory na temat konieczności zdefiniowania już na samym początku struktury i stylu — jeszcze zanim napiszesz cokolwiek w jQuery. Teraz jest to jeszcze ważniejsze: jeżeli z góry nie określisz układu i położenia elementów na stronie, Twoje efekty oraz animacje mogą okazać się nieudane, i to *szybko*. Nie ma nic gorszego, niż gapić się w swój kod jQuery i zastanawiać się, dlaczego nie robi on w przeglądarce tego, co chciałeś. Dobrym rozwiązaniem jest naszkicowanie swoich pomysłów i przemyślenie tego, co się będzie dziać na ekranie.



W każde z pustych miejsc w plikach HTML i CSS wpisz identyfikator CSS, właściwość albo ustawienie, które pomogą Ci skonfigurować układ i położenie elementów aplikacji Zrób sobie potwora. W przypadku wątpliwości poszukaj wskazówek na dwóch poprzednich stronach. Kilka z pustych miejsc wypełniliśmy za Ciebie.

```
<body>
<header id="top"><img src="images/Monster_Mashup.png" />
<p>Zrób sobie twarz potwora, klikając obrazek.</p></header>

<div id="frame">
    <div id="pic_box" >
        <div id=..... class="face"><img src="images/headsstrip.jpg"></div>
        <div id=..... class="face"><img src="images/eyesstrip.jpg"></div>
        <div id=..... class="face"><img src="images/nosesstrip.jpg"></div>
        <div id=..... class="face"><img src="images/mouthsstrip.jpg"></div>
    </div>
</div>
    <script type="text/javascript" src="scripts/jquery-1.7.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

Rozwiązanie ćwiczenia

W każde z pustych miejsc w plikach HTML i CSS wpisz identyfikator CSS, właściwość albo ustawienie, które pomogą Ci skonfigurować układ i położenie elementów aplikacji Zrób sobie potwora. W przypadku wątpliwości poszukaj wskazówek na dwóch poprzednich stronach. Kilka z pustych miejsc wypełniliśmy za Ciebie.

```
<body>
<header id="top"><img src="images/Monster_Mashup.png" />
<p>Zrób sobie twarz potwora, klikając obrazek.</p></header>

<div id="frame">
    <div id="pic-box">
        <div id="head" class="face"><img src="images/headsstrip.jpg"></div>
        <div id="eyes" class="face"><img src="images/eyesstrip.jpg"></div>
        <div id="nose" class="face"><img src="images/nosesstrip.jpg"></div>
        <div id="mouth" class="face"><img src="images/mouthsstrip.jpg"></div>
    </div>
</div>
    <script type="text/javascript" src="scripts/jquery-1.7.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;
}
```

Kiedy animujemy pozycję elementów, powinniśmy używać położenia absolutnego albo względnego.

Nadanie właściwości overflow wartości „hidden" umożliwi nam ukrycie części paska z rysunkiem, która wykracza poza obszar sekcji pic_box.

W tym celu można skorzystać też z właściwości CSS „clip".

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

#head{
    height:172px;
}

#eyes{
    height:79px;
}

#nose{
    height:86px;
}
#mouth{
    height:117px;
}
```

my_style.css

Trochę więcej struktury i stylu

W następnej kolejności czekają nas zmiany strukturalne w plikach HTML i CSS. Do plików *index.html* oraz *my_style.css* dodaj poniższy kod. Pliki graficzne możesz pobrać z *ftp://ftp.helion.pl/przyklady/jquerg.zip*.

Zrób to!

Dodaj kontener i zagnieźdź w nim rysunki z błyskawicą.

```
<div id="container">
    <img class="lightning" id="lightning1" src="images/lightning-01.jpg" />
    <img class="lightning" id="lightning2" src="images/lightning-02.jpg" />
    <img class="lightning" id="lightning3" src="images/lightning-03.jpg" />
     <div id="frame">
        <div id="pic_box">
           <div id="head" class="face"><img src="images/headsstrip.jpg"></div>
           <div id="eyes" class="face"><img src="images/eyesstrip.jpg"></div>
           <div id="nose" class="face"><img src="images/nosesstrip.jpg"></div>
           <div id="mouth" class="face"><img src="images/mouthsstrip.jpg"></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;
}
```

Chcemy, żeby rysunki z błyskawicą były początkowo niewidoczne.

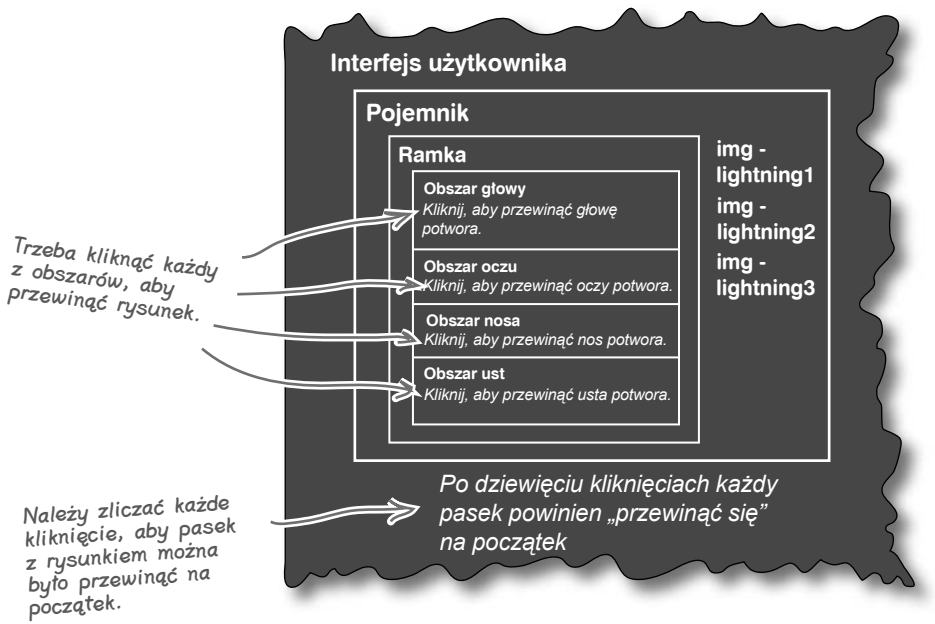
Kiedy chcemy animować elementy, powinniśmy nadać ich właściwości position wartość absolute, fixed albo relative.

```
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;
}
```

my_style.css

Uaktywniamy interfejs

Teraz, kiedy opracowaliśmy już wizualnie układ strony Zrób sobie potwora, skonfigurujmy resztę sekcji interfejsu użytkownika zgodnie z otrzymanym planem. Część ta sprowadza się do klikania w celu wywoływania zdarzeń. Zajmujesz się tym już od czterech rozdziałów, więc konfiguracja powinna Ci pójść jak z płatka.



Nie istnieją głupie pytania

P: Nie za bardzo już pamiętam, o co chodzi z tym pozycjonowaniem w CSS. Dlaczego jest nam to potrzebne do efektów i animacji w jQuery?

O: `position` to właściwość CSS, która kontroluje, jak i gdzie umieści elementy silnik przeglądarki odpowiedzialny za układ strony. jQuery realizuje wiele ze swoich efektów, korzystając z właściwości `position`. Jeśli już trochę pozapominałeś i potrzebujesz odświeżenia wiadomości, zajrzyj do znakomitych wyjaśnień znajdujących się w centrum deweloperskim Mozilli: *http://developer.mozilla.org/en/CSS/position#Relative_positioning*

P: Dlaczego musimy właściwości CSS `position` nadawać wartość `absolute`, `fixed` albo `relative`, kiedy chcemy animować elementy?

O: Jeśli pozostawimy właściwości CSS `position` jej domyślną wartość (czyli `static`), nie będziemy mogli zastosować położenia górnego, prawego, lewego ani dolnego. Kiedy zaczniemy pracować z funkcją `animate`, będziemy potrzebowali ustawiać takie położenia, a wartość `static` po prostu na to nie pozwala. Pozostałe ustawienia właściwości `position` — `absolute`, `fixed` oraz `relative` — to umożliwiają.

P: Wspomniałeś o silniku przeglądarki odpowiedzialnym za układ strony. Co to takiego?

O: Silnik przeglądarki to jej centralna część, która interpretuje kod HTML i CSS i wyświetla go w oknie roboczym przeglądarki (czyli oknie, które pokazuje użytkownikowi zawartość strony). Google Chrome i Safari korzystają z silnika Webkit, Firefox używa silnika Gecko, a Microsoft Internet Explorer silnika o nazwie Trident.

Magnesiki z kodem jQuery

Ułóż magnesiki z kodem w odpowiednim porządku, aby otrzymać aktywny element `div#head`. Upewnij się, że zmienne oraz instrukcje warunkowe zachowują właściwą kolejność, dzięki czemu będziesz mógł wykryć osiągnięcie przez zmienną `headclix` wartości 9, która oznacza dziesiąte kliknięcie.

```
headclix
});
}
if (headclix
= 0;
else {
headclix
= 0;
}
});
+= 1;
$(document).ready(function(){
< 9){
var headclix
$("#head").click(function(){
```

Magnesiki z kodem jQuery — rozwiązanie

Ułóż magnesiki z kodem w odpowiednim porządku, aby otrzymać aktywny element `div#head`. Upewnij się, że zmienne oraz instrukcje warunkowe zachowują właściwą kolejność, dzięki czemu będziesz mógł wykryć osiągnięcie przez zmienną `headclix` wartości 9, która oznacza dziesiąte kliknięcie.

```
$(document).ready(function(){
    var headclix = 0;
    $("#head").click(function(){
        if (headclix < 9){
            headclix += 1;
        }
        else {
            headclix = 0;
        }
    });
});
```

Zaczynasz od 0, ponieważ jeszcze nic nie zostało kliknięte.

Ten warunek ogranicza użytkownika do dziesięciu kliknięć.

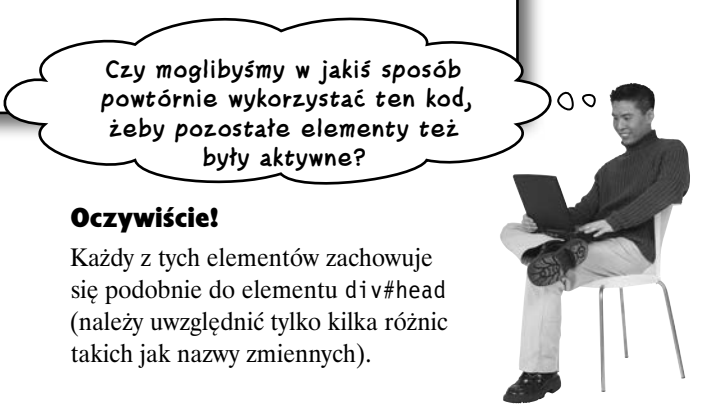
W to miejsce trafi kod animujący.

Nadaj zmiennej headclix wartość, jaką miała wcześniej, plus jeden.

Jeżeli zmienna headclix jest większa lub równa 9, zrób to.

W tym miejscu umieścisz kod przewijający pasek z rysunkiem.

Po dziesiątym kliknięciu zresetuj zmienną headclix.



Oczywiście!

Każdy z tych elementów zachowuje się podobnie do elementu `div#head` (należy uwzględnić tylko kilka różnic takich jak nazwy zmiennych).

Zaostrz ołówek

Uzupełnij poniższy skrypt jQuery, aby elementy oczu, nosa i ust były aktywne. Za chwilę do każdego kliknięcia dodamy odpowiednią funkcjonalność. Upewnij się, że zmienne oraz instrukcje warunkowe zachowują właściwą kolejność, dzięki czemu będziesz mógł wykryć dziesiąte kliknięcie.

```
$(document).ready(function(){

        $("#head").click(function(){
            if (headclix < 9){
                headclix += 1;
            }
            else{
                headclix = 0;
            }
        });









});
```

my_scripts.js

Zaostrz ołówek

Rozwiązanie

Tak więc uaktywniłeś elementy oczu, nosa i ust, ustawiając w odpowiedniej kolejności zmienne oraz instrukcje warunkowe, dzięki czemu będziesz mógł wykryć dziesiąte kliknięcie.

```
$(document).ready(function(){
        var headclix = 0, eyeclix=0, noseclix= 0, mouthclix = 0;
          $("#head").click(function(){
                if (headclix < 9){
                        headclix += 1;
                }
                else{
                        headclix = 0;
                }
          });
          $("#eyes").click(function() {
                if (eyeclix < 9){
                        eyeclix += 1;
                }
                else{
                        eyeclix = 0;
                }
          });
          $("#nose").click(function() {
                if (noseclix < 9){
                        noseclix += 1;
                }
                else{
                        noseclix = 0;
                }
           });
           $("#mouth").click(function() {
                if (mouthclix < 9){
                        mouthclix += 1;
                }
                else{
                        mouthclix = 0;
                }
          });
});
```

Możemy zadeklarować wiele zmiennych i nadać im wartości, rozdzielając te zmienne przecinkami.

Każda część twarzy potwora jest teraz aktywna i skonfigurowana w taki sposób, aby przed przewinięciem paska z rysunkiem pozwolić tylko na dziewięć kliknięć.

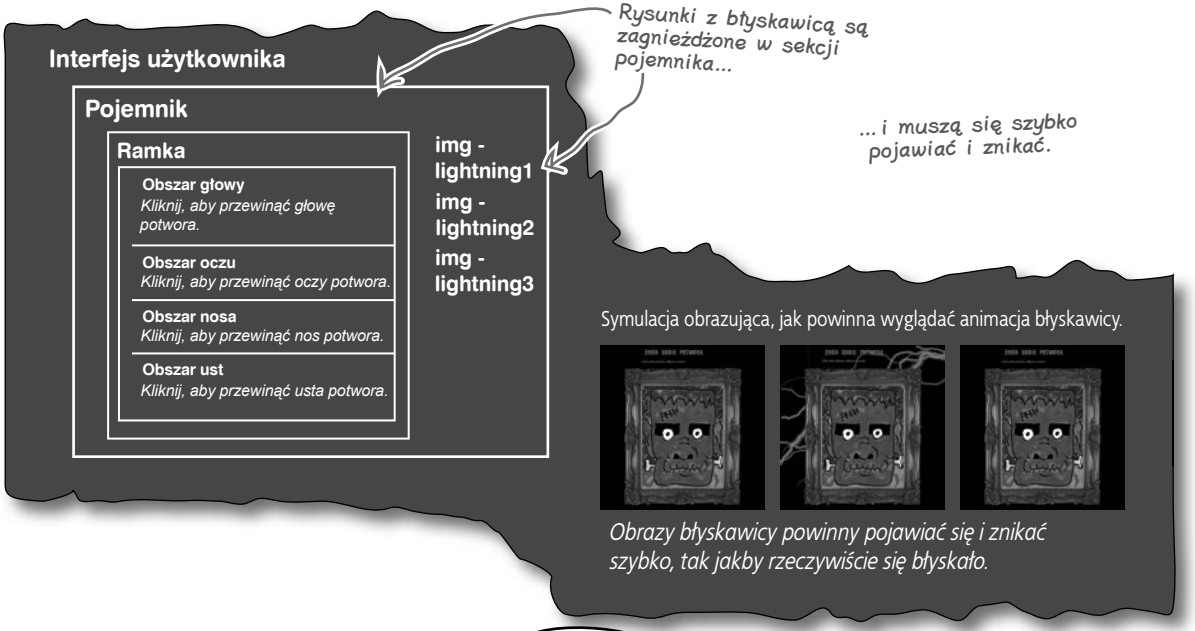
Czy zwróciłeś uwagę, że — pomimo drobnych różnic — każda funkcja wewnątrz metody click ma podobną strukturę? To mógłby być dobry przypadek do wielokrotnego wykorzystania kodu.

Cierpliwości, mój pasikoniku, dojdziemy do tego w rozdziale 7.

my_scripts.js

Robimy efekt błyskawicy

Następny w kolejności jest efekt błyskawicy. Zanim spróbujemy zrealizować jego działanie, sprawdźmy, co na ten temat jest w planie.



Zrobiliśmy już coś podobnego w rozdziale 1. z wysuwaniem się rysunku i nabieraniem przez niego barw. Czy nie moglibyśmy skorzystać z tego samego, żeby Zrób sobie potwora zadziałało?

Potencjalnie tak, ale być może istnieje lepszy sposób.

Przyjrzeliśmy się gotowym efektom jQuery w rozdziale 1., ale teraz zagłębimy się w nie trochę bardziej.

W jaki sposób jQuery animuje elementy?

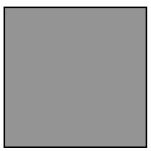
Kiedy przeglądarka pobiera plik CSS, ustawia wizualne właściwości elementów na stronie. Korzystając z wbudowanych efektów jQuery, interpreter JS zmienia te właściwości i animuje elementy na Twoich oczach. Nie jest to jednak magia... wszystko sprowadza się do właściwości CSS. Spójrzmy jeszcze raz na kilka efektów, które już widziałeś.

hide, show i toggle zmieniają właściwość CSS display

hide

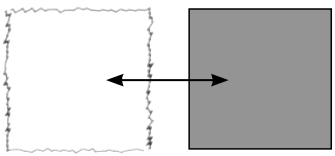
Interpreter JS zmienia właściwość CSS `display` wybranego elementu na `none` i usuwa go z układu strony.

show



Interpreter JS tak zmienia właściwość CSS `display` wybranego elementu, aby stał się on widoczny.

toggle



Jeśli element jest ukryty, interpreter JS go pokazuje, i odwrotnie.

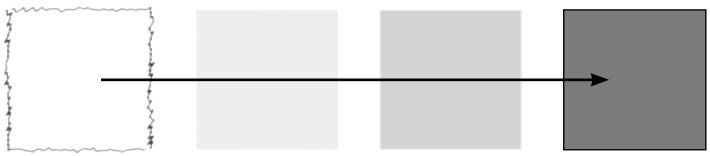
Efekty jQuery zmieniają właściwości CSS w locie, wprowadzając zmiany na stronie na oczach użytkownika.

WYSIL SZARE KOMÓRKI

`hide`, `show` i `toggle` zmieniają właściwość `display`. Potrzebujemy jednak przewijać części twarzy oraz w tym samym czasie włączać i wyłączać błyskawice. Którą właściwość CSS zmienia Twoim zdaniem jQuery w celu uzyskania tych efektów?

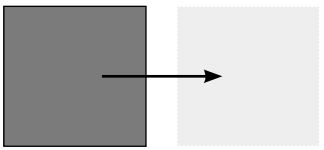
Efekty wygaszania zmieniają właściwość CSS opacity

fadeIn



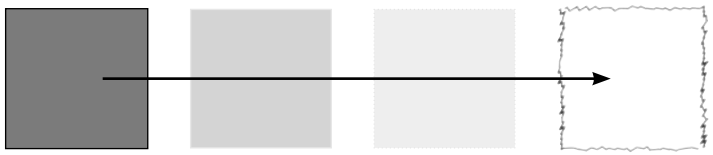
Za pomocą `fadeIn` interpreter JavaScriptu zmienia właściwość CSS `opacity` wybranego elementu od 0 do 100.

fadeTo



`fadeTo` umożliwia animowaną zmianę właściwości `opacity` wybranego elementu do określonej wartości procentowej.

fadeOut



Za pomocą `fadeOut` interpreter JavaScriptu zmienia właściwość CSS `opacity` wybranego elementu od 100 do 0 i pozostawia na stronie miejsce dla tego elementu.

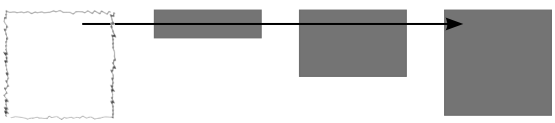
Porady maniaka

Właściwość CSS `opacity` nie działa tak samo na różnych przeglądarkach. Na szczęście jQuery zadba o to w naszym imieniu i tak naprawdę to wszystko, co musimy o tym wiedzieć!

Wysuwanie sprowadza się do zmiany wysokości

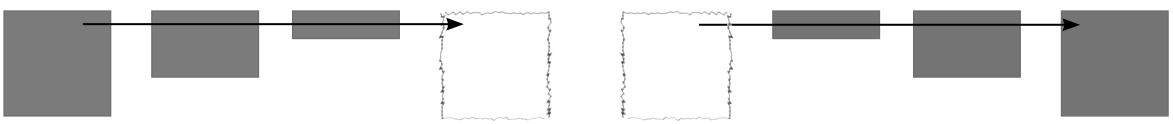
Interpreter JavaScriptu nakazuje drzewu DOM zmienić właściwość CSS `height` wybranego elementu (lub elementów) na 0, a następnie nadać właściwości `display` wartość `none`. W zasadzie jest to ukrycie za pomocą wsunięcia.

slideDown



Interpreter JavaScriptu pokazuje wybrany element (lub elementy), animując ich wysokość od 0 do wartości ustawionej w stylu CSS.

slideToggle



Interpreter JavaScriptu sprawdza, czy obraz ma pełną wysokość, czy zerową, i przełącza efekt wysunięcia w zależności od wyniku. Jeżeli element ma wysokość 0, interpreter wysuwa go w dół, a jeśli ma pełną wysokość, wsuwa go w górę.

Mogę więc wysuwać elementy wyłącznie w górę i w dół? A co, jeśli bym chciała przesunąć coś w lewo albo w prawo?

jQuery udostępnia gotowe efekty tylko do wysuwania elementów w górę albo w dół.

W bibliotece tej nie znajdziesz metody `slideRight` ani `slideLeft` (a przynajmniej nie było ich w chwili pisania tej książki). Nie przejmuj się jednak — zajmiemy się tym nieco później...

W jQuery nie znajdziesz metody slideRight ani slideLeft.

Zaostrz ołówek

Które z gotowych efektów jQuery przydadzą się w aplikacji Zrób sobie potwora? Dla każdej grupy efektów napisz, czy będą one pomocne, i wyjaśnij, dlaczego tę grupę wybrałeś lub nie.

Efekt	Czy możemy go użyć?	Dlaczego?
Pokazywanie/ ukrywanie		
Wysunięcia		
Wygaszenia		

Zaostrz ołówek

Rozwiązanie

Które z gotowych efektów jQuery przydadzą się w aplikacji Zrób sobie potwora?

Efekt	Czy możemy go użyć?	Dlaczego?
Pokazywanie/ ukrywanie	*Nie*	*Efekty pokazywania i ukrywania (show i hide) nie pomogą nam przy tworzeniu aplikacji Zrób sobie potwora, ponieważ nie musimy animować właściwości display żadnego elementu.*
Wysunięcia	*Nie*	*O mały włos! Pasek z rysunkiem musimy przesunąć w lewo, a slideUp, slideDown i efekty wysunięć pozwalają na zmienianie tylko właściwości height. Potrzebujemy czegoś, co zmieniałoby właściwość left.*
Wygaszenia	*Tak*	*Możemy skorzystać z efektów wygaszania, aby sprostać wymogom planu, które stwierdzają, że rysunki błyskawic powinny pojawiać się i wygaszać szybko, jakby naprawdę się błyskało.*

Zatrudnij efekty wygaszania

W założeniach jest mowa o tym, że rysunki błyskawic powinny pojawiać się i wygaszać, ale ma się to dziać szybko, aby wyglądało tak, jakby naprawdę się błyskało. Zagłębmy się trochę w efekty wygaszania i zobaczmy, jak możemy sprawić, że błyskanie będzie działać.

Tutaj jest identyfikator pierwszego elementu graficznego.

Tutaj wstawiamy parametr duration decydujący o czasie trwania animacji. Kontroluje on to, jak długo będzie trwał efekt.

```
$("#lightning1").fadeIn("fast");
```

Możesz użyć jednego z parametrów tekstowych: slow, normal lub fast...

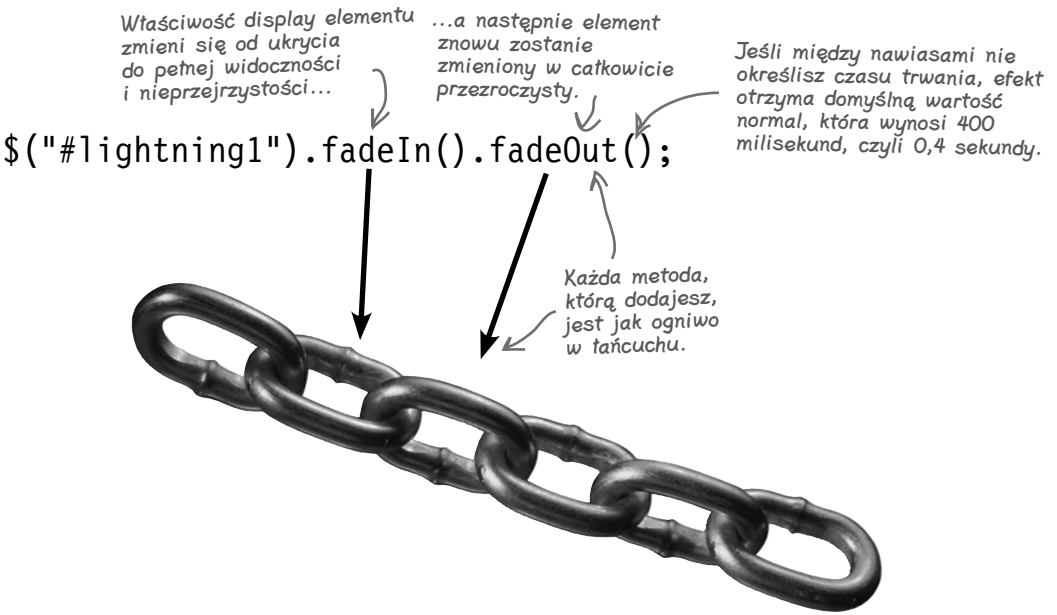
Przyklej to sobie

...albo możesz podać wartość w milisekundach. Mógłbyś na przykład wstawić 1000 i animacja efektu trwałaby jedną sekundę.

```
$("#lightning1").fadeIn(1000);
```

Łącz efekty za pomocą łańcuchów metod

Błyskawice muszą pojawiać się i wygaszać wciąż na nowo. Zamiast rozpisywać te efekty oddzielnie, możemy skorzystać z łańcucha metod, z którego korzystaliśmy przez chwilę w rozdziale 4., kiedy musieliśmy wspinać się po drzewie DOM. Łańcuchy metod to cecha jQuery pozwalająca na łączenie ze sobą metod, które będą działać na zwróconym zbiorze elementów. Dzięki nim efekty błyskawic będą łatwiejsze do napisania, a zatem przyjrzyjmy się im bliżej.



Ćwiczenie

Napisz wiersze kodu, które zrealizują każdy z etapów przedstawionych poniżej.

1. Pokaż stopniowo element `#lightning1` w czasie jednej czwartej sekundy.

   .....

2. Dodaj kolejny efekt, który wygasi element `#lightning1` w czasie jednej czwartej sekundy.

   .....

Rozwiązanie ćwiczenia

Napisz wiersze kodu, które zrealizują każdy z etapów przedstawionych poniżej.

1. Pokaż stopniowo element `#lightning1` w czasie jednej czwartej sekundy.

```
$("#lightning1").fadeIn("250");
```

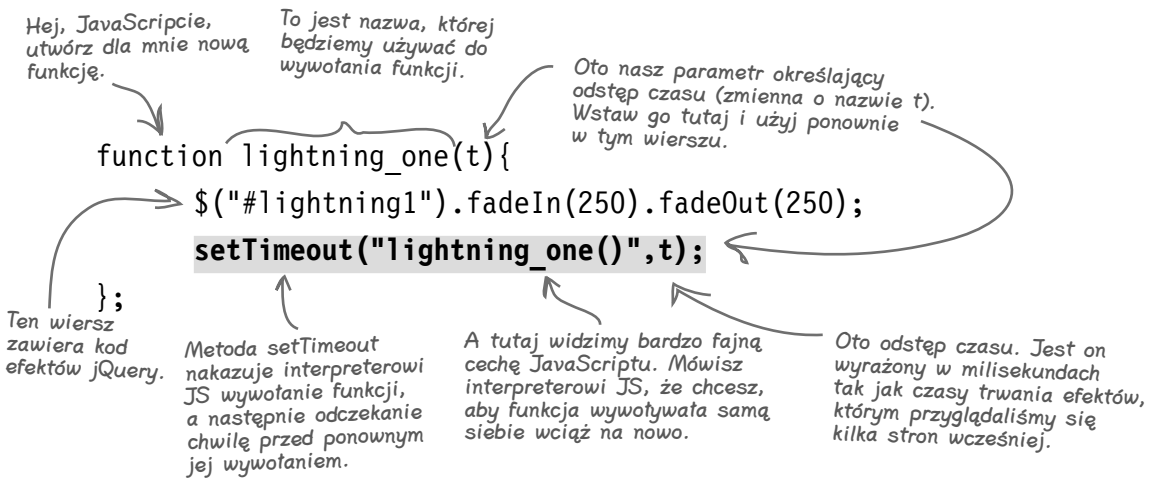
2. Dodaj kolejny efekt, który wygasi element `#lightning1` w czasie jednej czwartej sekundy.

```
$("#lightning1").fadeIn("250").fadeOut("250");
```

Kontratakujemy za pomocą funkcji czasowych

Masz już więc błyskawicę, która pojawia się i znika, ale wymogi projektu mówią, że błyskawice powinny uderzać przez cały czas. Prawdziwa błyskawica strzela przez niebo, po czym zwykle następuje przerwa, po której na niebie pojawia się kolejna. Potrzebujemy zatem *powtarzać* nasz efekt.

Powróć myślami do poprzednich rozdziałów, gdzie musiałeś wykonywać powtarzalne zadania. Z czego korzystałeś? Zgadza się: z funkcji! Po raz pierwszy pojawiły się one w rozdziale 3., gdy tworzyliśmy funkcję wielokrotnego użytku obsługującą kliknięcia oraz generator liczb losowych. Teraz możemy użyć funkcji do wywoływania wygaszeń, poczekać chwilę, a następnie znowu je wywoływać w różnych odstępach czasu. Dzięki temu uzyskamy dla Zrób sobie potwora udany efekt wielu błyskawic. Spójrzmy na funkcję realizującą taki efekt.



W zaledwie trzech wierszach kodu uzyskałeś zsynchronizowaną w czasie funkcję błyskającą pierwszym rysunkiem pioruna. Spróbuj teraz napisać funkcje dla pozostałych dwóch rysunków z piorunami.

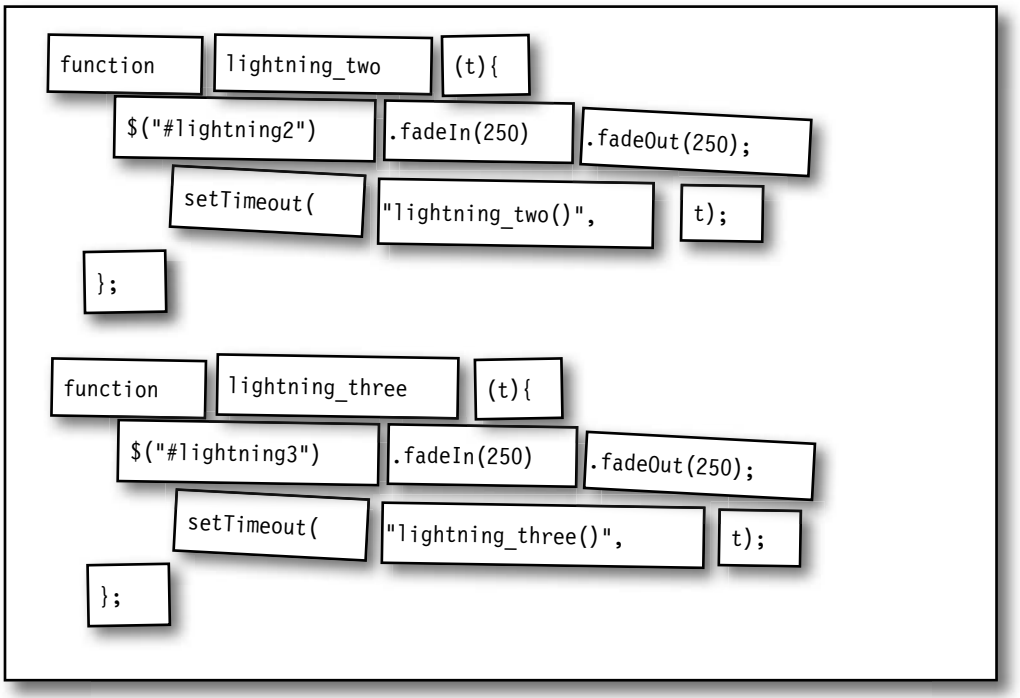
Magnesiki jQuery

Poukładaj magnesiki z kodem we właściwej kolejności, aby uzyskać funkcje animujące błyskawice dla pozostałych dwóch elementów z piorunami.

```
function
lightning_two
(t){

};

function
lightning_three
(t){

};

.fadeIn(250)
t);
setTimeout(
$("#lightning2")
.fadeIn(250)
"lightning_two()",
.fadeOut(250);
"lightning_three()",
.fadeOut(250);
setTimeout(
t);
$("#lightning3")
```

Magnesiki jQuery — rozwiązanie

Poukładaj magnesiki z kodem we właściwej kolejności, aby uzyskać funkcje animujące błyskawice dla pozostałych dwóch elementów z piorunami.



Nie istnieją głupie pytania

P: Czy `fadeIn().fadeOut()` nie jest tym samym co `toggle`?

O: Dobre pytanie! To nie jest to samo. Metoda `toggle` jest pojedynczą metodą, która po prostu przełącza wybrany element ze stanu ukrytego na widoczny i na odwrót w zależności od bieżącego stanu tego elementu. Połączenie `fadeIn` i `fadeOut` w łańcuch utworzy sekwencyjny efekt, który najpierw stopniowo pokaże wybrany element lub elementy, po czym — gdy to zostanie już osiągnięte — ponownie je wygasi.

P: Metoda `setTimeout` jest dla mnie nowa. Czy pochodzi ona z jQuery, czy z JavaScriptu?

O: Metoda `setTimeout` pochodzi tak naprawdę z JavaScriptu i możesz z niej korzystać w celu kontrolowania pewnych aspektów animacji w jQuery. Funkcją `setTimeout` zajmiemy się dokładniej w dalszych rozdziałach, a zwłaszcza w rozdziale 7.

Jeśli chciałbyś poczytać o niej już teraz, odwiedź Mozilla Developer's Center pod adresem *https://developer.mozilla.org/en/window.setTimeout*, a jeżeli naprawdę chcesz pogłębić swoją wiedzę, sięgnij po znakomitą książką Davida Flanagana *JavaScript: The Definitive Guide* (O'Reilly; *http://oreilly.com/catalog/9780596805531*).

P: Kiedy używam efektu `hide`, element po prostu znika. Jak mogę to spowolnić?

O: Aby spowolnić efekt `hide`, `show` albo `toggle`, dodaj między nawiasami parametr `duration` określający czas trwania animacji. Oto, jak moglibyśmy zrealizować efekt ukrywania w rozdziale 1.:

```
$("#picframe").hide(500);
```

Dodaj do swojego skryptu funkcje błyskawic

Zrób to!

Korzystając z kodu, który utworzyłeś w ćwiczeniu na poprzedniej stronie, zaktualizuj plik ze skryptem dla aplikacji Zrób sobie potwora.

W tych wierszach są wywoływane funkcje zdefiniowane na samym dole pogrubioną czcionką.

```
$(document).ready(function(){
    var headclix = 0, eyeclix = 0, noseclix = 0, mouthclix = 0;
    lightning_one(4000);
    lightning_two(5000);
    lightning_three(7000);

    $("#head").click(function(){
        if (headclix < 9){headclix+=1;}
        else{headclix = 0;}
    });

    $("#eyes").click(function(){
        if (eyeclix < 9){eyeclix+=1;}
        else{eyeclix = 0;}
    });

    $("#nose").click(function(){
        if (noseclix < 9){noseclix+=1;}
        else{noseclix = 0;}
    });

    $("#mouth").click(function(){
        if (mouthclix < 9){mouthclix+=1;}
        else{mouthclix = 0;}
    });

});//koniec funkcji end doc.onready
```

Liczby między nawiasami to parametry w milisekundach, które zostaną przekazane do metody setTimeout. Za ich pomocą możesz zmieniać czasy błyskania piorunów.

Dla oszczędności miejsca usunęliśmy niektóre podziały wierszy. Nie przejmuj się, jeśli w Twoim skrypcie wyglądają one inaczej.

To są definicje funkcji błyskawic.

```
function lightning_one(t){
    $("#container #lightning1").fadeIn(250).fadeOut(250);
    setTimeout("lightning_one()",t);
};
function lightning_two(t){
    $("#container #lightning2").fadeIn("fast").fadeOut("fast");
    setTimeout("lightning_two()",t);
};
function lightning_three(t){
    $("#container #lightning3").fadeIn("fast").fadeOut("fast");
    setTimeout("lightning_three()",t);
};
```

my_scripts.js

Jazda próbna

Otwórz stronę w swojej ulubionej przeglądarce, aby sprawdzić, czy Twój efekt błyskawicy działa.

Udało Ci się osiągnąć efekt wygaszania błyskawic, łącząc go z metodą JavaScriptu setTimeout.

Pioruny pojawiają się i znikają szybko w różnych odstępach czasu, symulując prawdziwe błyskawice.

Jak dotąd udało Ci się opracować funkcje obsługujące kliknięcia, a trzy rysunki błyskawic pojawiają się i wygaszają w różnych odstępach czasu. Spójrzmy na plan, aby sprawdzić, co pozostało jeszcze do zrobienia.

Projekt Zrób sobie potwora

Zadaniem aplikacji Zrób sobie potwora jest zabawianie dzieci w docelowej grupie wiekowej przez umożliwienie im zrobienia sobie potwora dzięki mieszaniu 10 różnych głów, oczu, nosów i ust. Przejścia między poszczególnymi częściami twarzy potwora powinny być animowane.

Animacja

Symulacja obrazująca, jak powinna się zmieniać część twarzy potwora.

Symulacja obrazująca, jak powinna wyglądać animacja błyskawicy.

Ta część planu stanowi nasze ostatnie wyzwanie w projekcie.

Dotarliśmy więc do miejsca, w którym potrzebujemy przesunąć rysunek w lewo, a żaden z gotowych efektów przesuwania tego nie robi. Czy istnieje jakaś inna metoda, której moglibyśmy użyć?

Gotowe efekty są wspaniałe, ale nie pozwalają Ci robić wszystkiego, na co tylko masz ochotę.

Nadeszła pora na opracowanie niestandardowego efektu, który będzie przesuwać części twarzy potwora w lewo.

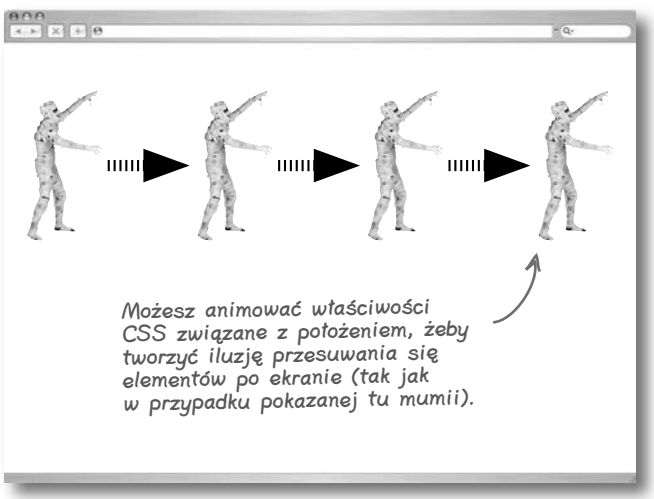
Efekty własnej roboty wykorzystujące metodę animate

Tak więc w jQuery nie ma efektu `slideRight` ani `slideLeft`, a dokładnie tego potrzebujesz na tym etapie realizacji projektu. Czy to oznacza, że nasza strona Zrób sobie potwora poległa?

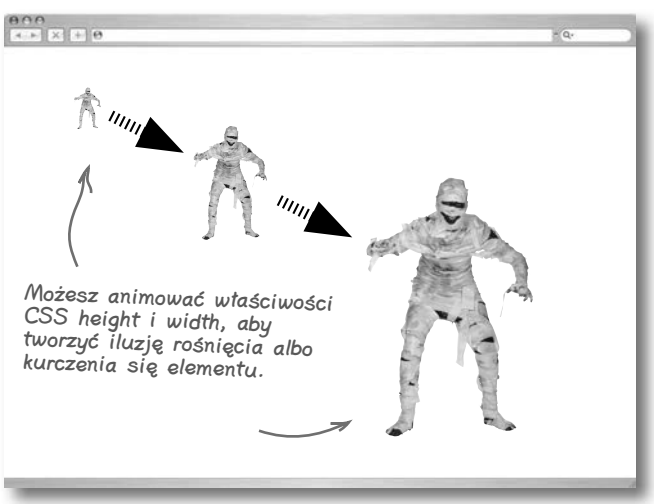
Bez obaw. jQuery oferuje metodę `animate` służącą do tworzenia własnych efektów. Za pomocą `animate` możesz tworzyć niestandardowe animacje, które będą potrafiły robić o wiele więcej niż gotowe efekty. Metoda ta umożliwia animowanie właściwości CSS wybranego elementu lub elementów, a także animowanie wielu właściwości w tym samym czasie.

Rzućmy okiem na niektóre z efektów, jakie możesz osiągnąć za pomocą metody `animate`.

Efekty ruchu



Efekty skali



WYSIL SZARE KOMÓRKI

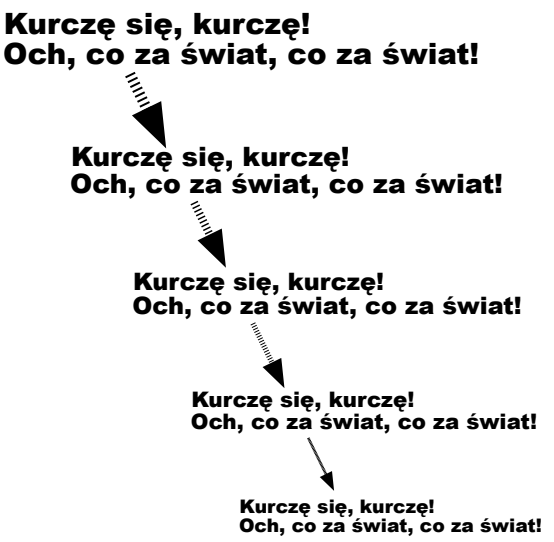
Którą właściwość CSS będziesz musiał animować, aby przy każdym kliknięciu części twarzy potwora przesuwały się w lewo?

Co można, a czego nie można animować?

Za pomocą metody `animate` możesz dynamicznie zmieniać właściwości czcionek, aby tworzyć efekty tekstowe. W jednym wywołaniu animacji możesz także animować wiele właściwości CSS jednocześnie, co poszerza paletę ciekawych rzeczy, które potrafi robić Twoja aplikacja sieciowa.

Chociaż metoda `animate` jest naprawdę niezła, ma ona swoje ograniczenia. Gdzieś tam w głębi animacja wykorzystuje mnóstwo matematyki (czym na szczęście nie musisz się przejmować), a zatem jesteś ograniczony do pracy tylko z tymi właściwościami CSS, których ustawienia są *numeryczne*. Znaj swoje ograniczenia, ale popuść wodze wyobraźni — metoda `animate` oferuje pełne spektrum elastyczności i rozrywki.

Efekty tekstowe



Możesz animować właściwości CSS czcionek, aby tworzyć iluzję lecącego, rosnącego albo malejącego tekstu.

To zaledwie kilka przykładów. Potrzebowalibyśmy o wiele, wiele więcej stron w książce, żeby zaprezentować wszystkie możliwości.

Metoda `animate` będzie działać wyłącznie z właściwościami CSS, które w swoich ustawieniach korzystają z liczb:

- obramowanie, margines, dopełnienie;
- wysokość elementu, wysokość minimalna i maksymalna;
- szerokość elementu, szerokość minimalna i maksymalna;
- rozmiar czcionki;
- pozycja dolna, lewa, prawa i górna;
- pozycja tła;
- odstępy między literami i wyrazami;
- wcięcie tekstu;
- wysokość linii.

Metoda animate z bliska

Z zewnątrz metoda `animate` działa tak jak inne metody, z którymi już pracowałeś.

Wybierz element lub elementy, z którymi chcesz pracować.

Wywołaj metodę animate.

Pierwszy parametr pozwala Ci wybrać właściwość CSS, którą chcesz animować.

Drugi parametr to czas trwania w milisekundach. Pozwala Ci on określić, jak długo będzie trwać animacja.

```
$("#my_div").animate({left:"100px"},500);
```

W tym przykładzie animujemy właściwość CSS left...

...i ustawiamy ją na 100 pikseli.

Pierwszy parametr jest wymagany — musisz go tu umieścić, aby animacja zadziałała. Drugi parametr jest opcjonalny.

Jedną z największych zalet metody `animate` jest jednak możliwość zmieniania wielu właściwości wybranego elementu w tym samym czasie.

```
$("#my_div").animate({
    opacity: 0,
    width: "200",
    height: "800"
}, 5000);
```

W tym przykładzie animujemy jednocześnie przezroczystość i wielkość elementu.

Obejrzyj to!

Parametry właściwości CSS należy ustawiać zgodnie ze standardem DOM, a nie ze standardem CSS.

WYTĘŻ UMYSŁ

Jak myślisz — co takiego dzieje się za kulisami w przeglądarce, co pozwala metodzie `animate` zmieniać elementy na oczach użytkownika?

Metoda animate zmienia w czasie styl

Efekty wizualne oraz animacja, które oglądasz na ekranie w kinie albo w telewizji, korzystają ze złudzenia ruchu. Specjaliści od efektów i animatorzy biorą sekwencję obrazów i odtwarzają je **w określonym tempie po jednym obrazie**, aby to złudzenie osiągnąć. Spotkałeś się zapewne z opracowanymi tym prostym sposobem książkami, w których ten sam efekt jest osiągany podczas ich wertowania.

To samo dzieje się w oknie przeglądarki, z tym że nie mamy sekwencji obrazów do pokazania. Zamiast tego interpreter JavaScriptu **wielokrotnie wywołuje funkcję, która zmienia styl animowanego obiektu**. Przeglądarka przenosi te zmiany na ekran. Użytkownik widzi iluzję ruchu albo zmianę elementu, kiedy modyfikowany jest jego styl.

1. Kiedy działa metoda `animate`, interpreter JavaScriptu ustawia licznik na czas trwania animacji.

2. Interpreter JavaScriptu mówi silnikowi przeglądarki, żeby zmienił właściwość CSS określoną w parametrze metody `animate`. Silnik przeglądarki wyświetla tę właściwość na ekranie.

3. Interpreter JavaScriptu wielokrotnie wywołuje funkcję zmieniającą właściwość CSS elementu do chwili, w której licznik ustawiony w punkcie 1 wyzeruje się. Przy każdym wywołaniu funkcji zmiana jest pokazywana na ekranie.

4. Użytkownik widzi iluzję ruchu, kiedy przeglądarka renderuje zmiany elementu.

Połącz fragmenty kodu zawierające metodę `animate` z opisem tego, co robi on na ekranie.

```
$("#my_div").animate({top: "150px"}, "slow")
```

Jednocześnie animuje zmiany lewego i prawego marginesu wszystkich paragrafów.

```
$("p").animate({
        marginLeft:"150px",
        marginRight:"150px"
});
```

Animuje zmianę prawej pozycji `#my_div` do zera w ciągu pół sekundy.

```
$("#my_div").animate({width: "30%"}, 250)
```

Animuje zmianę odstępu między literami we wszystkich sekcjach w domyślnym czasie 400 milisekund.

```
$("#my_div").animate({right: "0"}, 500)
```

Jednocześnie animuje zmianę dopełnienia i szerokości `#my_div`.

```
$("p").animate({letterSpacing:"15px"});
```

Powoli animuje zmianę górnej pozycji `#my_div`.

```
$("#my_div").animate({
        padding: "200px",
        width: "30%"
}, "slow")
```

Szybko animuje zmianę wysokości wszystkich obrazów.

```
$("img").animate({height: "20px"}, "fast")
```

Animuje zmianę szerokości `#my_div` w czasie jednej czwartej sekundy.

KTO CO ROBI?

Rozwiązanie

Połącz fragmenty kodu zawierające metodę `animate` z opisem tego, co robi on na ekranie.

```
$("#my_div").animate({top: "150px"}, "slow")
```

Jednocześnie animuje zmiany lewego i prawego marginesu wszystkich paragrafów.

```
$("p").animate({
        marginLeft:"150px",
        marginRight:"150px"
});
```

Animuje zmianę prawej pozycji `#my_div` do zera w ciągu pół sekundy.

```
$("#my_div").animate({width: "30%"}, 250)
```

Animuje zmianę odstępu między literami we wszystkich sekcjach w domyślnym czasie 400 milisekund.

```
$("#my_div").animate({right: "0"}, 500)
```

Jednocześnie animuje zmianę dopełnienia i szerokości `#my_div`.

```
$("p").animate({letterSpacing:"15px"});
```

Powoli animuje zmianę górnej pozycji `#my_div`.

```
$("#my_div").animate({
        padding: "200px",
        width: "30%"
}, "slow")
```

Szybko animuje zmianę wysokości wszystkich obrazów.

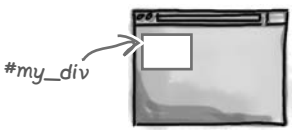
```
$("img").animate({height: "20px"}, "fast")
```

Animuje zmianę szerokości `#my_div` w czasie jednej czwartej sekundy.

Dokładnie skąd dokąd?

Ważną rzeczą do zapamiętania na temat metody `animate` jest fakt, że zmienia ona *aktualną* właściwość CSS na właściwość, którą ustawisz *w pierwszym parametrze*. Aby Twoje niestandardowe animacje były skuteczne, musisz się mocno zastanowić, jakie wartości są aktualnie ustawione w CSS. W poprzednim przykładzie zmieniliśmy lewe położenie `#my_div` na 100 pikseli. To, co wydarzy się na ekranie, w całości zależy od bieżącej wartości właściwości CSS `left` elementu `#my_div`.

Bieżąca wartość właściwości CSS




```
#my_div{
  left: 20px;
}
```

Wartość właściwości CSS w animate

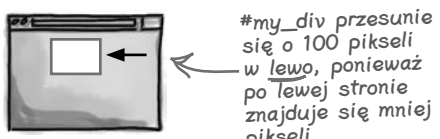

```
$("#my_div").animate({left:"100px"});
```

Jeśli właściwość ma aktualnie inną wartość, otrzymamy inny wynik.

Bieżąca wartość właściwości CSS


```
#my_div{
  left: 200px;
}
```

Wartość właściwości CSS w animate




```
$("#my_div").animate({left:"100px"});
```

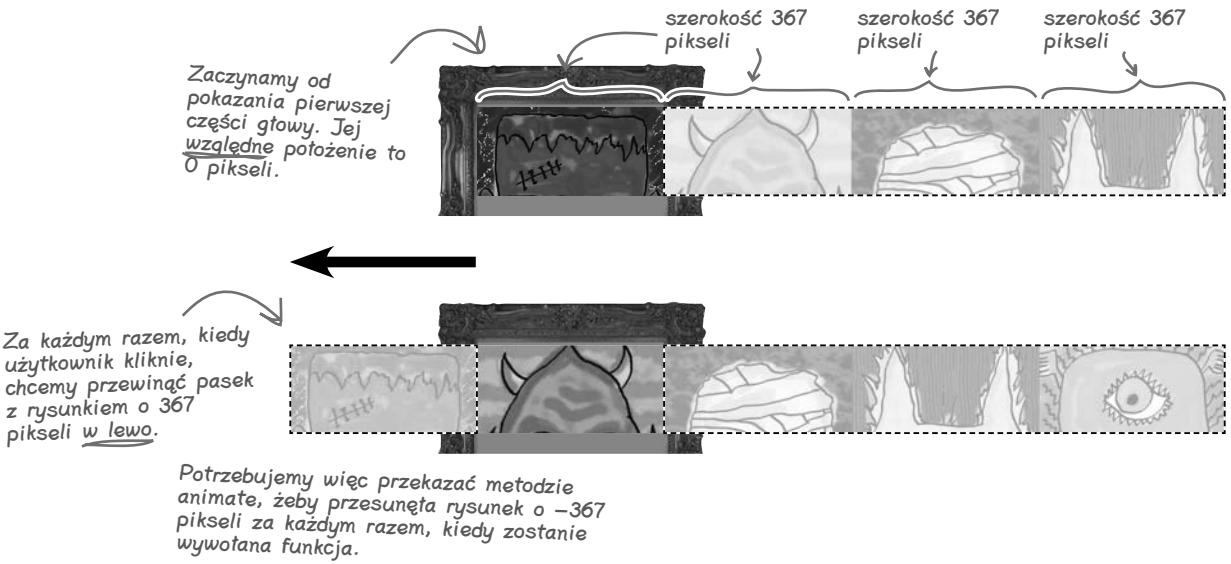
Fascynujące! Tylko jak możemy z tego skorzystać, żeby zadziałało nasze Zrób sobie potwora?

Wszystko jest względne.

Aby części twarzy potwora w Zrób sobie potwora przesuwały się w stronę, w którą chcemy, musimy zastanowić się, jakie są ich *bieżące pozycje* i jak chcemy je zmienić względem *pozycji*, które zajęły po tym, kiedy ostatnio zmieniła je metoda `animate`.

Ruch bezwzględny obiektów a ich ruch względny

Zapewne pamiętasz, że paski z rysunkami, które chcemy pokazać, zagnieździliśmy wewnątrz sekcji o identyfikatorze `#pic_box`. Obecnie w CSS właściwość `left` elementu `#pic_box` ma wartość `91px`. Zastanówmy się nad tym, jak chcielibyśmy przesuwać paski z rysunkami, aby uzyskać efekt przesunięcia w lewo, na którym nam zależy.



Zastanów się nad przykładem animacji bezwzględnej z poprzedniej strony.

Tutaj mówimy metodzie animate, żeby nadała lewej pozycji sekcji #my_div wartość równą dokładnie 100 pikseli.

```
$("#my_div").animate({left:"100px"});
```

Ale jak powiedzieć jej, żeby przesunęła element o −367 pikseli za każdym razem, kiedy metoda animate zostanie wywołana?

```
$("#head").animate({left:"???"});
```

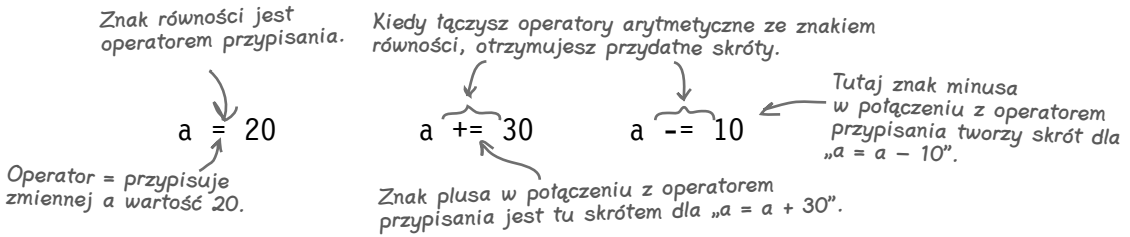
Animacja względna = przesuń to za każdym razem właśnie o tyle

Za pomocą *animacji bezwzględnej* przesuwasz element na bezwzględną pozycję siatki ekranu. Za pomocą animacji względnej przesuwasz element *względem* pozycji, którą zajmował *ostatnio* po zakończeniu animacji, która go tam umieściła.

Ale jak przesuwamy element względnie za pomocą metody `animate`?

Przesuwaj elementy względnie dzięki łączeniu operatorów

Istnieją specjalne operatory JavaScriptu, które przesuwają element lub elementy o tę samą wartość przy każdym wywołaniu metody `animate`. Są one znane jako *operatory przypisania*, ponieważ są zwykle używane do przypisywania wartości zmiennej w taki sposób, że do jej bieżącej wartości jest dodawana nowa wartość. Brzmi to o wiele bardziej skomplikowanie, niż w istocie jest.



Takie połączenia operatorów pomagają w tworzeniu animacji względnej, umożliwiając ustawianie nowej wartości na wartość bieżącą *plus* albo *minus* pewna liczba pikseli.

Ten zapis przesunie element o identyfikatorze box o 20 pikseli za każdym razem, kiedy zostanie wywołana metoda animate.

```
$("#box").animate({left:"+=20"});
```

Oto, co się stanie z elementem `#box` przy każdym wywołaniu powyższej metody `animate`.

CELNE SPOSTRZEŻENIA

Oto dwa inne zestawienia operatorów przypisania:

- a `*=` 5 jest skrótem dla przemnożenia bieżącej wartości a przez 5 i przypisania uzyskanego wyniku do a.
- a `/=` 2 jest skrótem dla podzielenia bieżącej wartości a przez 2 i przypisania uzyskanego wyniku do a.

Ćwiczenie

Napisz wiersz kodu w jQuery, który zrealizuje następujące czynności:

1. Przesunięcie elementu `#head` o 367 pikseli w lewo przy każdym wywołaniu metody `animate`. Czas trwania ustaw na pół sekundy.

.....

2. Przesunięcie elementu `#head` na pozycję wyjściową (`left:0px`). Czas trwania ustaw na pół sekundy.

.....

Rozwiązanie ćwiczenia

Napisz wiersz kodu w jQuery, który zrealizuje następujące czynności:

1. Przesunięcie elementu #head o 367 pikseli w lewo przy każdym wywołaniu metody `animate`. Czas trwania ustaw na pół sekundy.

```
$("#head").animate({left:"-=367px"},500);
```

2. Przesunięcie elementu #head na pozycję wyjściową (`left:0px`). Czas trwania ustaw na pół sekundy.

```
$("#head").animate({left:"0px"},500);
```

Ta bezwzględna animacja resetuje głowę potwora w sposób przypominający przewinięcie taśmy.

Nie istnieją głupie pytania

P: Niektóre osoby nie chcą, żeby animacje przeszkadzały im w oglądaniu strony. Co powinienem zrobić, żeby pozwolić użytkownikowi na wyłączenie animacji?

O: To doskonałe spostrzeżenie. Animacje mogą irytować i powodować problemy z dostępnością. Jeśli chcesz, żeby użytkownicy mogli je wyłączać, możesz utworzyć przycisk powiązany z następującym kodem (wiesz już, jak to zrobić):

```
$.fx.off = true;
```

Innym przydatnym sposobem na wyłączenie animacji jest użycie metody jQuery o nazwie `stop`. Możesz się dowiedzieć więcej na temat obu tych sposobów w witrynie jQuery:

http://api.jquery.com/jQuery.fx.off/

http://api.jquery.com/stop/

P: Powiedziałeś: „Parametry właściwości CSS należy ustawiać zgodnie ze standardem DOM, a nie ze standardem CSS". Co to do diaska znaczy?

O: Świetne pytanie! Metoda `animate` pobiera parametry napisane w standardzie DOM (zwanym też notacją DOM) zamiast w notacji CSS.

Oto konkretny przykład ilustrujący tę różnicę. Aby ustawić szerokość obramowania sekcji w notacji CSS, napisałbyś coś takiego:

```
div {
border-style:solid;
border-width:5px;
}
```

Załóżmy teraz, że chciałbyś animować tę szerokość obramowania. W jQuery właściwość szerokości krawędzi ustawia się za pomocą notacji DOM, tak jak tutaj:

```
$("div").animate({borderWidth:30
},"slow");
```

Zauważ, że w notacji CSS dla właściwości szerokości piszesz `border-width`, natomiast w notacji DOM dla tej samej właściwości używasz zapisu `borderWidth`.

Jeśli chcesz poczytać więcej o różnicach między tymi dwoma notacjami, zapoznaj się z poniższym artykułem:

http://www.oxfordu.net/webdesign/dom/straight_text.html

P: A co powinienem zrobić, jeśli chciałbym animować zmianę koloru?

O: Aby animować przejścia między kolorami, musisz skorzystać z jQuery UI, które zawiera więcej efektów, niż zostało dołączonych do jQuery. jQuery UI opiszemy w rozdziale 10., ale z pominięciem efektów. Kiedy już dowiesz się, jak pobierać i ustawiać motywy oraz dołączać jQuery UI do swoich aplikacji internetowych, animowanie zmian koloru stanie się bardzo proste.

Dodaj funkcje animujące do swojego skryptu

Korzystając z kodu, który ułożyłeś w ćwiczeniu na poprzedniej stronie, zaktualizuj swój plik ze skryptem dla aplikacji Zrób sobie potwora.

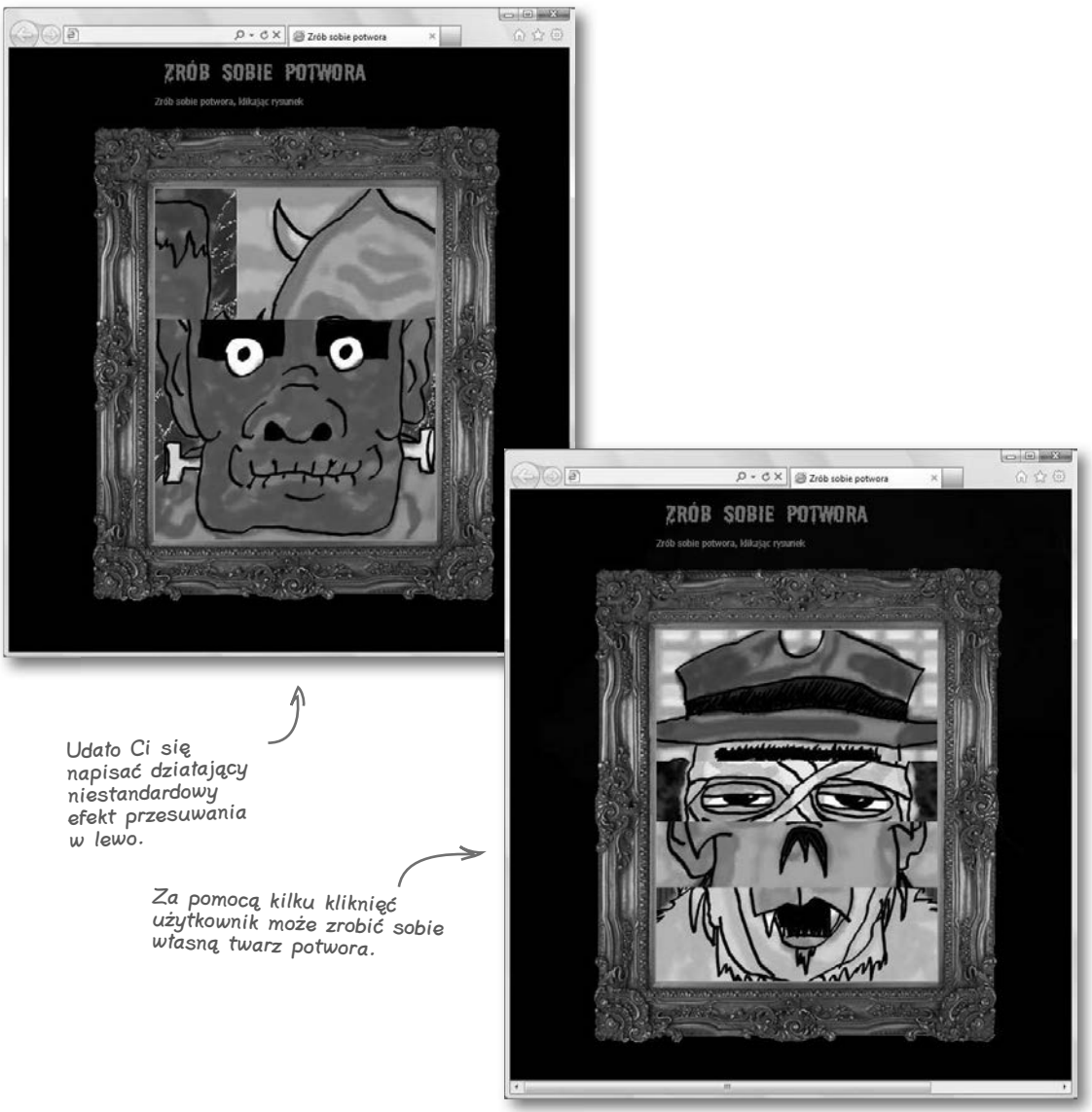
```
$("#head").click(function(){
        if (headclix < 9){
                $(this).animate({left:"-=367px"},500);
                headclix+=1;
        }
        else{
                $(this).animate({left:"0px"},500);
                headclix = 0;
        }
});

$("#eyes").click(function(){
        if (eyeclix < 9){
                $(this).animate({left:"-=367px"},500);
                eyeclix+=1;
        }
        else{
                $(this).animate({left:"0px"},500);
                eyeclix = 0;
        }
});

$("#nose").click(function(){
        if (noseclix < 9){
                $(this).animate({left:"-=367px"},500);
                noseclix+=1;
        }
        else{
                $(this).animate({left:"0px"},500);
                noseclix = 0;
        }
});

$("#mouth").click(function(){
        if (mouthclix < 9){
                $(this).animate({left:"-=367px"},500);
                mouthclix+=1;
        }
        else{
                $(this).animate({left:"0px"},500);
                mouthclix = 0;
        }
```

Możemy tu skorzystać ze słowa kluczowego „this”, ponieważ znajdujemy się wewnątrz funkcji obsługującej element, który kliknęliśmy.

my_scripts.js

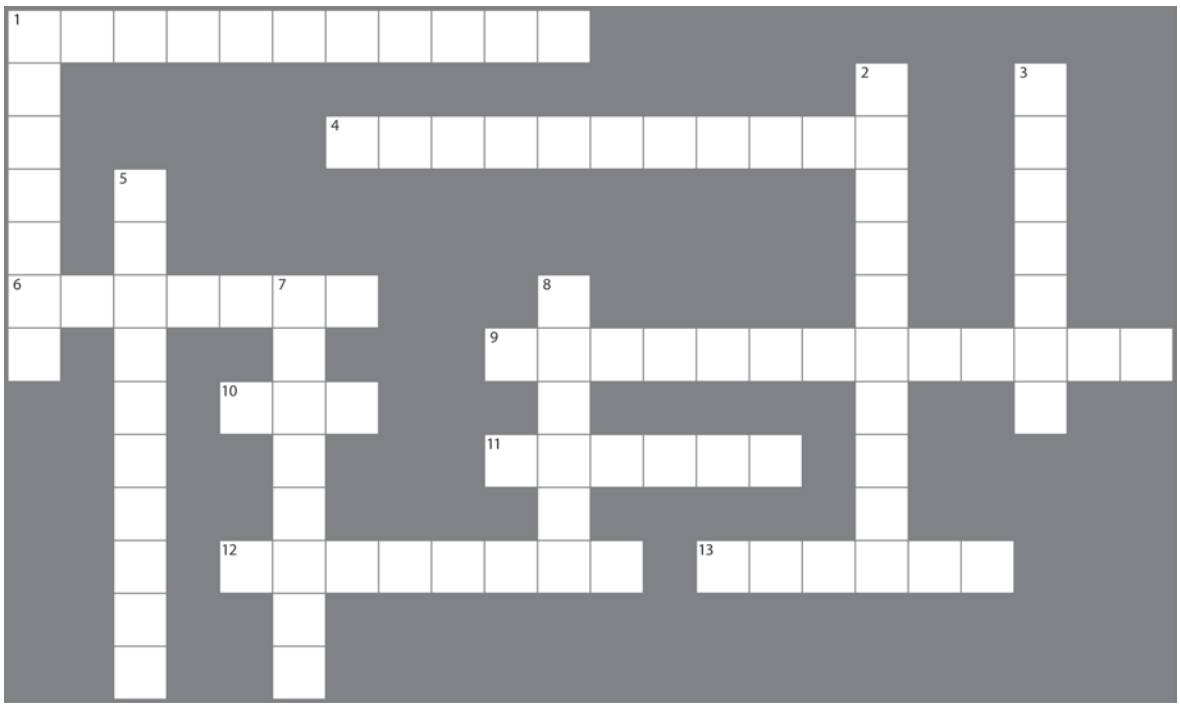
Jazda próbna

Otwórz stronę w swojej ulubionej przeglądarce, aby upewnić się, że wszystko działa.



Krzyżówka jQuery

Nadszedł czas, aby wygodnie usiąść i dać swojej lewej półkuli jakieś zajęcie. Będzie to zwykła krzyżówka. Wszystkie wyrazy i wyrażenia stanowiące hasła pochodzą z bieżącego rozdziału.



Poziomo

1. Metoda powodująca następujące działanie: jeżeli wskazany element ma wysokość 0, interpreter JS wysuwa go w dół, jeśli natomiast wskazany element ma pełną wysokość, interpreter JS wsuwa go do góry.

4. Tworzą iluzję elementu przesuwającego się po ekranie.

6. `hide`, `show` i `toggle` animują tę właściwość CSS.

9. Funkcjonalność jQuery umożliwiająca łączenie ze sobą metod, które chcesz wywołać dla danego zbioru elementów.

10. Efekty jQuery polegają na manipulowaniu nim w locie.

11. Tę właściwość CSS oraz właściwość `width` możesz animować w celu stworzenia iluzji rosnącego albo malejącego elementu.

12. Parametr kontrolujący czas trwania efektu.

13. Kiedy wywołasz ten efekt jQuery, interpreter JS zmieni przezroczystość wskazanego elementu od 0 do 100.

Pionowo

1. 1000 milisekund.

2. Metoda `animate` będzie działać tylko dla właściwości CSS, które mają takie wartości.

3. Biblioteka jQuery oferuje tę metodę, jeśli zechcesz tworzyć efekty niestandardowe.

5. Efekt uzyskany za pomocą animowania właściwości `height` elementu.

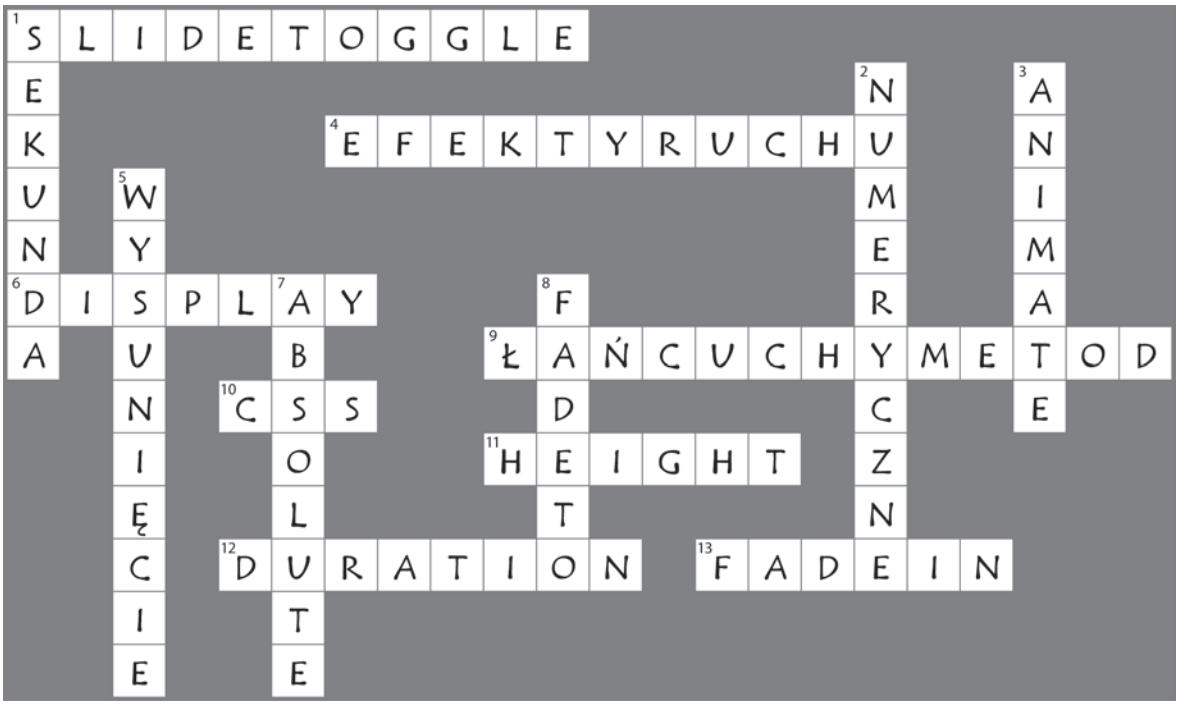
7. Jedna z wartości, obok `fixed` i `relative`, jaką można nadać właściwości określającej położenie elementu w celu jego animowania.

8. Metoda nadająca danemu elementowi w animowany sposób określony stopień przezroczystości.

Patrz, mamo! Bez Flasha!

Dyrektor ds. projektów internetowych jest zadowolona z wyników Twojej pracy nad aplikacją Zrób sobie potwora. Skorzystałeś z gotowych efektów jQuery w połączeniu z własnymi niestandardowymi efektami skrojonymi na miarę potrzeb klienta.

Rozwiązanie krzyżówki jQuery



Twój niezbędnik jQuery

Masz już za sobą rozdział 5., a do swojego niezbędnika dodałeś efekty wygaszania kolorów oraz wysuwania, a także niestandardowe animacje.

Efekty wygaszania

Zmieniają właściwość CSS opacity wskazanego elementu:

fadeIn

fadeOut

fadeTo

Efekty wysuwania

Zmieniają właściwość CSS height wskazanego elementu:

slideUp

slideDown

slideToggle

animate

Umożliwia tworzenie niestandardowych animacji, kiedy gotowe efekty jQuery już nie wystarczają.

Animuje właściwości CSS.

Działa tylko z takimi właściwościami CSS, które przyjmują wartości liczbowe.

Elementy można przesuwać względnie albo bezwzględnie.

Połączenia operatorów (=, +, −) w znacznym stopniu ułatwiają animację względną.

Skorowidz

K

L

Ł

M

N

O

P

Władaj stronami z jQuery!

jQuery. Rusz głową!

Jeżeli Twoja witryna internetowa ma się wyróżniać na tle konkurencji, musi być interaktywna, przyjemna dla oka i wygodna w użyciu. Jeżeli nie spełnia tych warunków, będzie jej niezwykle trudno zdobyć popularność. Ale zawsze możesz wykorzystać JavaScript! Zastosowanie tego języka programowania pozwala na istotne wzbogacenie strony. Z kolei użycie biblioteki jQuery sprawia, że wykorzystanie potencjału JavaScriptu jeszcze nigdy nie było tak proste. Jeżeli dodać do tego liczne gotowe do użycia efekty specjalne, otrzymamy niemalże perfekcyjne rozwiązanie.

Kolejna książka z serii *Rusz głową!* to najlepszy sposób na opanowanie możliwości biblioteki jQuery. Atrakcyjna forma graficzna oraz nowoczesna metodologia nauczania sprawiają, że przyswajanie wiedzy jest przyjemne i efektywne. Autorzy nie wymagają od Ciebie znajomości języka JavaScript — dzięki temu możesz z marszu zacząć poznawać jQuery. Nie ma na co czekać! Sprawdź, jak używać selektorów, reagować na zdarzenie, modyfikować drzewo DOM oraz używać efektów specjalnych. *jQuery. Rusz głową!* to Twoja przepustka do tworzenia angażujących, interaktywnych witryn WWW, które wyglądają i działają tak jak prawdziwe aplikacje!

- **Wykorzystaj efekty specjalne**
- **Bez trudu modyfikuj drzewo DOM**
- **Nasłuchuj zdarzeń**
- **Zaskocz swoich użytkowników**

Poznaj jQuery i ulepsz swoją stronę internetową!

helion.pl księgarnia internetowa

Nr katalogowy: 8634

Księgarnia internetowa: http://helion.pl

Zamówienia telefoniczne: 0 801 339900, 0 601 339900

Helion

Sprawdź najnowsze promocje: http://helion.pl/promocje
Książki najchętniej czytane: http://helion.pl/bestsellery
Zamów informacje o nowościach: http://helion.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

Informatyka w najlepszym wydaniu

sięgnij po WIĘCEJ
KOD KORZYŚCI

ISBN 978-83-246-3864-2

9 788324 638642

Cena 77,00 zł