

Użycie wszystkich trzech metod zaprezentowano na listingu 1.23.

Na stronie zostały umieszczone trzy przyciski. Pierwszy wyświetla zdjęcie (metoda `.show()`), drugi je ukrywa (metoda `.hide()`), trzeci zaś przełącza widok zdjęcia z ukrytego na wyświetlone i na odwrót (metoda `.toggle()`). Opcjonalnie po nazwie metody podaje się w nawiasie czas trwania efektu (w milisekundach).

**Listing 1.23. jQuery — efekty. Użycie metod `.show()`, `.hide()` oraz `.toggle()`**

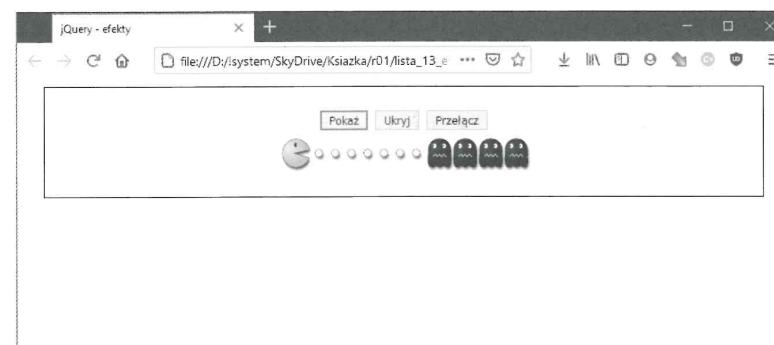
```
<html>
<head>
  <title>jQuery - efekty</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" type="text/css" href="styl.css" />
  <script src="jquery-3.6.0.js"></script>
  <script type="text/javascript">

    $(function() {

      $('#pokaz').on('click', function() {
        $('#img').eq(0).show(1000);
      });
      $('#ukryj').on('click', function() {
        $('#img').hide(500);
      });
      $('#przelacz').on('click', function() {
        $('#img').toggle();
      });
    });
  </script>
</head>

<body>
  <div id="container">
    <div id="page">
      <button id="pokaz">Pokaż</button>
      <button id="ukryj">Ukryj</button>
      <button id="przelacz">Przełącz</button>
      <br></img>
    </div>
  </div>
</body>
</html>
```

Rysunek 1.18 przedstawia kod z listingu 1.23.



**Rysunek 1.18.** jQuery — efekty. Użycie metod `.show()`, `.hide()` oraz `.toggle()`

Na listingu 1.24 jest pokazane użycie efektów jQuery typu **slide** (wślizg).

**Listing 1.24. jQuery — efekty. Użycie metod `.slideUp()`, `.slideDown()` oraz `.slideToggle()`**

```
<html>
<head>
  <title>jQuery - efekty</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" type="text/css" href="styl.css" />
  <style>
    .prostokat {
      border: 1px solid black;
      width: 200px;
      height: 150px;
      background: blue;
      margin: 0 20px;
      float: left;
    }
  </style>
  <script src="jquery-3.6.0.js"></script>
  <script type="text/javascript">

    $(function() {
      $('#wysun').on('click', function() {
        $('#pierwszy').toggle().slideDown(1000);
      });
      $('#wysun2').on('click', function() {
        $('#drugi').toggle().slideUp(2000);
      });
      $('#wysun3').on('click', function() {
        $('#trzeci').slideToggle('slow');
      });
    });
  </script>
```

```

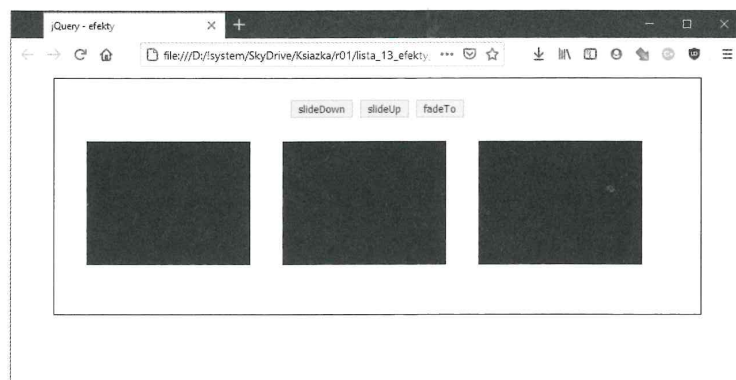
</script>
</head>

<body>
  <div id="container">
    <div id="page" style="height: 250px">
      <button id="wysun">slideDown</button>
      <button id="wysun2">slideUp</button>
      <button id="wysun3">fadeTo</button><br><br>

      <div id="pierwszy" class="prostokat"></div>
      <div id="drugi" class="prostokat"></div>
      <div id="trzeci" class="prostokat"></div>
      <div style="clear: both"></div>
    </div>
  </div>
</body>
</html>

```

Kliknięcie przycisku z nazwą efektu uruchamia animację prostokąta (rysunek 1.19).



**Rysunek 1.19.** jQuery — efekty. Użycie metod `.slideUp()`, `.slideDown()` oraz `.slideToggle()`

## Użycie animacji

jQuery umożliwia **animowanie stylów CSS**. Warunek jest jeden: wartość stylu **musi być wyrażana w postaci liczby**, np. `width`, `padding`, `font-size`. Animowane nie mogą być te właściwości, których **wartość to tekst**, takie jak `text-align` czy `font-family`.

Ogólna składnia metody `.animate()` wygląda następująco:

```

.animate({animowany_styl} <, czas_trwania> <, przebieg_animacji> <,
funkcja_wywołania_zwrotnego>),

```

Parametr `czas_trwania` określa długość animacji (w ms), dozwolone jest użycie słów kluczowych `fast` i `slow`. Użycie zapisu `+=` doda odpowiednią wielkość do aktualnej, a `-=` odejmie. Opis i przeznaczenie pozostałych parametrów zostały zamieszczone w dalszej części rozdziału.

Kod z listingu 1.25 symuluje wyścig dwóch kulek. Animowane są takie właściwości CSS jak `left` i `top`.

Animacje są wyzwalane przez przyciski. Pierwszy, *Start niebieska*, włącza animację kulki niebieskiej, drugi, *Start czerwona* — czerwonej, a wybranie przycisku *Start obie* spowoduje jednoczesny start obu kulek. Start obu kulek jest możliwy dzięki użyciu metody `.add()`, która pozwala dodać element do istniejącej grupy elementów. Alternatywnym sposobem tworzenia grupy jest wyliczenie jej elementów, przy czym nazwy oddzielamy przecinkiem, np. `$('#start1, #start2').click();`.

Po analizie kodu można wywnioskować, że wyścig jako pierwsza ukończy kulka niebieska (na pokonanie odległości 700 px potrzebuje ona 600 ms, czerwona zaś — 800 ms), dlatego po zakończeniu animacji kulki niebieskiej uruchamiana jest funkcja (tzw. **funkcja wywołania zwrotnego**) wyświetlająca tekst *Pierwszy!!!*.

Jeśli jednocześnie są animowane dwie właściwości CSS, muszą one znajdować się w **nawiasie klamrowym i być oddzielone od siebie znakiem przecinka**.

### WSKAZÓWKA

Pomiędzy czasem animacji a funkcją wywołania zwrotnego może się znajdować parametr `easing`, który może przyjmować dwie wartości: `linear` i `swing`. Użycie pierwszej sprawi, że prędkość animacji pozostanie stała, a zastosowanie drugiej zwiększy prędkość animacji w środku i jednocześnie zmniejszy na początku i końcu trwania przejścia. Efekt użycia obu przełączników został zademonstrowany na stronie <https://jqueryui.com/easing/>.

Wciśnięcie przycisku *Od nowa* spowoduje wczytanie ustawień początkowych — właściwości CSS `left` i `top` zostaną wyzerowane.

Ostatni przycisk, *Stop*, zatrzymuje animację. Wykorzystana metoda `.clearQueue()` usuwa z kolejki wszystkie efekty, które nie zostały jeszcze uruchomione.

### WSKAZÓWKA

W kodzie z listingu 1.25 zostały użyte właściwości CSS, których nazwy składają się z jednego słowa. W przypadku właściwości, których nazwy składają się z wielu słów oddzielonych od siebie znakiem łącznika, np. `border-radius`, w połączeniu z metodą `.animate()` używamy notacji **camelCase** — pierwsze słowo rozpoczynamy od **małej litery**, a każde następne od **dużej**. Dla właściwości CSS `border-radius` prawidłowym zapisem będzie `borderRadius`.

**Listing 1.25. jQuery — animacja**

```

<html>

<head>
  <title>jQuery - zdarzenia</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" type="text/css" href="styl.css" />
  <style>
    #pierwszy {
      background: blue;
      height: 60px;
      width: 60px;
      border-radius: 100px;
      left: 50px;
      top: 200px;
      position: absolute;
    }

    #drugi {
      background: red;
      height: 60px;
      width: 60px;
      border-radius: 100px;
      left: 50px;
      top: 300px;
      position: absolute;
    }
  </style>

  <script src="jquery-3.6.0.js"></script>
  <script type="text/javascript">

    $(function() {

      $('#start1').on('click', function() {
        const div = $('#pierwszy');
        div.animate({ 'left': '+=300px' }, 'slow');
        div.animate({ 'top': '+=50px' }, 'slow');
        div.animate({ 'left': '+=700px' }, 600);
      });

      $('#start2').on('click', function() {
        const div = $('#drugi');

```

```

        div.animate({ 'top': '+=50px' }, 'slow');
        div.animate({ 'left': '+=700px' }, 800);
      });

      $('#start3').on('click', function() {
        $('#start1, #start2').click();
      });

      $('#reset').on('click', function() {
        $('#log').text('');
        $('#pierwszy, #drugi').css({ left: '', top: '' });
      });

      $('#stop').click(function() {
        const kulka = $('.kulka');
        kulka.clearQueue();
        kulka.stop();
      });
    });
  </script>
</head>

<body>
  <div id="container">
    <div id="page">
      <button id="start1">&raquo; Start niebieska</button>
      <button id="start2">&raquo; Start czerwona</button>
      <button id="start3">&raquo; Start obie</button>
      <button id="stop">&raquo; Stop</button>
      <button id="reset">&raquo; Od nowa</button>

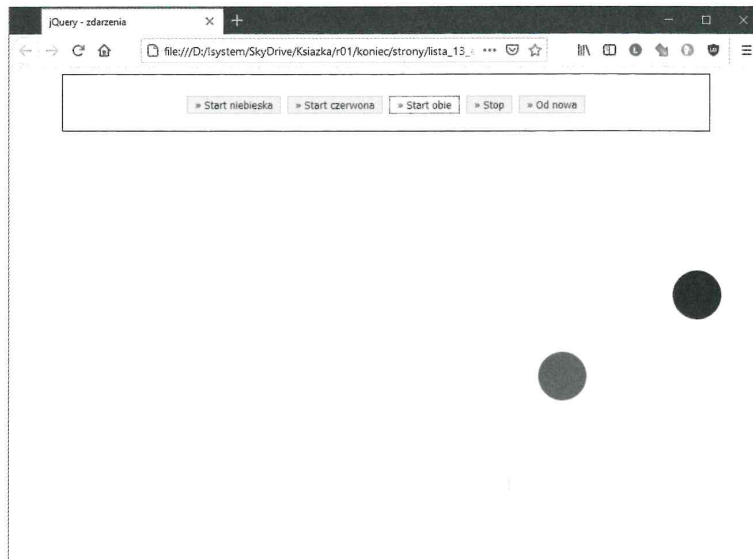
      <div id="pierwszy" class="kulka"></div>
      <div id="drugi" class="kulka"></div>
      <div id="log"></div>
    </div>
  </div>
</body>

</html>

```

Kod z listingu 1.25 jest zilustrowany na rysunku 1.20.





Rysunek 1.20. jQuery — animacja

**Zadanie 1.6.**

Utwórz akapit z wpisanym własnym imieniem. Za pomocą kodu CSS ustaw szerokość akapitu, tło oraz obramowanie. Pod akapitem dodaj dwa przyciski: *Start* i *Przywróć*. Wykonaj animację akapitu zmieniającą szerokość akapitu, wielkość użytej czcionki oraz wielkość obramowania. Animacja jest uruchamiana po wciśnięciu przycisku *Start*, a resetowana po naciśnięciu przycisku *Przywróć*.

**Zadanie 1.7.**

Przygotuj stronę, która będzie wyświetlać dwa przyciski: *W prawo* i *W lewo*. Pod przyciskami ma zostać umieszczony kwadrat (utworzony za pomocą znacznika `<div>`). Napisz skrypt, który przesunie kwadrat o wartość 80 px. Kierunek przesunięcia zależy od tego, który przycisk zostanie wybrany.

**1.1.5. Formularz**

W jQuery nie zapomniano o obsłudze **formularzy** — szczególnie ważnego elementu każdej strony internetowej. Dlatego udostępniono metody, które to ułatwiają.

**Elementy, zdarzenia i metody formularza sieciowego**

jQuery zapewnia szereg selektorów, które pozwolą uchwycić żądany element formularza. Stosując się do połączenia z *nazwą elementu CSS* bądź *jQuery*

Do najczęściej wykorzystywanych selektorów należą:

- `:button` — elementy `<button>` i `<input>` z atrybutem `type`, który ma wartość `button`;
- `:checkbox` — element `<input>`, którego atrybut `type` ma wartość `checkbox`;
- `:disabled/:enabled` — elementy, które zostały wyłączone/włączone;
- `:input` — elementy `<button>`, `<input>`, `<select>` oraz `<textarea>`;
- `:password` — element `<input>`, którego atrybut `type` ma wartość `password`;
- `:radio` — element `<input>`, którego atrybut `type` ma wartość `radio`;
- `:submit` — element `<input>` bądź `<button>`, którego atrybut `type` ma wartość `submit`;
- `:text` — element `<input>`, którego atrybut `type` ma wartość `text`.

W połączeniu z przedstawionymi selektorami najczęściej wykorzystywanymi metodami będą `.val()`, której zadaniem jest pobranie wartości z pola formularza, oraz `.is()`, która sprawdzi, czy zaznaczono bądź wybrano pole formularza.

**Praca z formularzem**

Listing 1.26 pokazuje szereg pól formularza, od typu `text`, poprzez pole listy i pole wielokrotnego wyboru, po pola typu `radio` i `checkbox`.

Pierwszym polem formularza jest pole tekstowe. Skrypt pobiera zawartość pola tekstowego po każdym zwolnieniu klawisza i wyświetla ją w akapicie umieszczonym pod polem (rysunek 1.21).

**Pole formularza input typu text:**

przykładowy tekst

Rysunek 1.21. jQuery — formularz, pole tekstowe

Drugim polem formularza jest lista jednokrotnego wyboru. Wybranie z listy pozycji *Stron internetowych* spowoduje wyświetlenie formularza z polami typu *radio*. Nazwa wybranej opcji zostanie wyświetlona w wierszu pod listą (rysunek 1.22).

**Pole jednokrotnego wyboru:**

Języki programowania wykorzystujesz do tworzenia:

Twój wybór to: Stron internetowych

---

PHP ☐ JavaScript ☐

Rysunek 1.22. jQuery — formularz, lista jednokrotnego wyboru oraz pola typu radio