

Biblioteka jQuery

Biblioteki języka JavaScript umożliwiają szybkie i łatwe tworzenie strony internetowej. Do najpopularniejszych należą: jQuery, Angular, Knockout, React, Dojo. Są to biblioteki open source. Każda z nich zawiera specyficzny dla siebie zestaw funkcji usprawniających pracę z modelem DOM.



4.1. Opis biblioteki

Biblioteka jQuery to framework zawierający różnorodne funkcje, za pomocą których można realizować często spotykane zadania (na przykład animacje, obsługę zdarzeń, tworzenie efektów wizualnych na stronie internetowej). Jej głównymi zaletami są łatwość użycia, niezależność od przeglądarki i małe rozmiary otrzymanego kodu. Biblioteka ułatwia pracę nad stroną internetową i umożliwia tworzenie czytelnego kodu.

4.1.1. Zasady korzystania z biblioteki jQuery

Jest kilka sposobów na rozpoczęcie pracy z biblioteką jQuery. Najpopularniejsze to:

- pobranie biblioteki jQuery ze strony <http://www.jquery.com>,
- dołączenie biblioteki jQuery z sieci dostarczania treści, na przykład Google.

Pobieranie jQuery

Aby korzystać z biblioteki jQuery, należy pobrać jej plik ze strony <http://www.jquery.com>. Na stronie dostępne są dwie wersje biblioteki jQuery, wersja nieskompresowana i skompresowana.

Wersja nieskompresowana przeznaczona jest do programowania i testowania.

W celu skrócenia czasu ładowania w wersji skompresowanej usunięte są niepotrzebne komentarze, podziały wierszy, spacje i znaki tabulacji. Zmniejszenie objętości pliku uzyskano kosztem jego czytelności, dlatego ta wersja pliku nie nadaje się do testowania i analizy kodu.

Dołączenie jQuery z sieci

Druga metoda to korzystanie z biblioteki za pośrednictwem sieci dostarczania treści. Rozwiązania hostingowe są udostępniane przez firmy o dużych sieciach, takie jak Google, Microsoft. Gdy użytkownik otworzy wybraną stronę na swoim komputerze, biblioteka jQuery zostanie przesłana przez serwer znajdujący się najbliżej niego. Dzięki temu skróci się czas ładowania strony.

Biblioteka jQuery jest pojedynczym plikiem języka JavaScript, do którego można odwoływać się za pomocą znacznika `<script>`. Znacznik ten powinien znajdować się w sekcji `<head>`.

Po wybraniu metody korzystania z biblioteki należy dołączyć jQuery w kodzie projektowanej strony internetowej.

W sekcji `<head>` dokumentu HTML powinien zostać wstawiony wpis:

```
<script src="jquery-3.4.1.min.js"></script>
```

Podany kod dodaje do skryptu bibliotekę jQuery w wersji 3.4.1.

Aby zapewnić poprawność przetwarzania strony, arkusze stylów CSS powinny znaleźć się w dokumencie HTML przed biblioteką jQuery. W tym dokumencie powinien zostać zdefiniowany również element `doctype`. Jego brak może spowodować błędne zachowania przeglądarki lub biblioteka jQuery może zostać niepoprawnie przetworzona.

Sposób dołączania biblioteki jQuery pobranej na lokalny komputer i zapisanej w folderze *js* pod nazwą *jquery.js* został pokazany w przykładzie 4.1.

Przykład 4.1

```
<!DOCTYPE html>

<html>

<head>

<title>Dołączenie biblioteki jQuery</title>

<meta charset="UTF-8">

<link href="css/style.css">

<script src="js/jquery.js"></script>

<script>
// treść skryptu
</script>

</head>

<body>

...

</body>

</html>
```

Alternatywnie biblioteka jQuery może zostać dołączona u dołu strony przed znacznikiem domykającym `</body>`.

Sposób dołączania biblioteki jQuery udostępnionej w ramach hostingu bibliotek przez firmę Google został pokazany w przykładzie 4.2.

Przykład 4.2

```
<!DOCTYPE html>

<html>

<head>

<title>Dołączenie biblioteki jQuery</title>

<meta charset="UTF-8">

<link href="css/style.css">

<script src="https://ajax.googleapis.com/ajax/
libs/jquery/3.4.1/jquery.min.js">

</script>

<script>
// treść skryptu
</script>

</head>

<body>

...

</body>

</html>
```

Dołączanie biblioteki jQuery udostępnionej przez firmę Microsoft wygląda następująco:

```
<script src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.4.1.min.js"
></script>
```

Z kolei dołączanie biblioteki jQuery udostępnionej na serwerze jQuery wygląda tak:

```
<script src="http://code.jquery.com/jquery-3.4.1.min.js"></script>
```

Podstawowym sposobem korzystania z biblioteki jQuery jest użycie funkcji `$()`. Wybiera ona określone elementy dokumentu drzewa DOM i pozwala wchodzić w różnego rodzaju interakcje. Przykładowo zapis `$(document)` powoduje pobranie odnośnika do dokumentu HTML. Druga metoda to użycie zapisu jQuery w postaci: `jQuery(document)`.

Istnieje kilka sposobów na dołączanie funkcji, która zostanie uruchomiona, gdy struktura drzewa DOM otwieranej witryny zostanie załadowana do przeglądarki. Dla przykładu:

```
$(document).ready(function() {
    // tutaj znajdzie się wykonywany kod
});
```

lub

```
$(function() {
    // tutaj znajdzie się wykonywany kod
});
```

Od wersji jQuery 3.0 zalecane jest stosowanie tylko tej drugiej wersji składni.

4.1.2. Selektory

Selektory to podstawowe elementy biblioteki jQuery. Są używane we wszystkich operacjach dotyczących modelu DOM. Za ich pomocą wybieramy te elementy strony, które będą przetwarzane. Biblioteka jQuery korzysta z selektorów arkuszy CSS i kilku niestandardowych selektorów jQuery. Najczęściej używane selektory CSS zostały przedstawione w tabeli 4.1.

Tabela 4.1. Wybrane selektory CSS

Selektor	Znaczenie
<code>\$('div')</code>	pobiera wszystkie bloki <code>div</code> z dokumentu
<code>\$('.blok')</code>	pobiera z dokumentu wszystkie elementy, które mają klasę <code>blok</code>
<code>\$('#id')</code>	pobiera z dokumentu jeden element o identyfikatorze <code>id</code>
<code>\$('p')</code>	pobiera z dokumentu wszystkie akapity
<code>\$('*')</code>	pobiera wszystkie elementy modelu DOM lub innego elementu
<code>\$(this).hide()</code>	ukrywa bieżący element
<code>\$('p').hide()</code>	ukrywa wszystkie elementy <code><p></code>
<code>\$('#id').hide()</code>	ukrywa element o identyfikatorze <code>id</code>

Ogólny sposób użycia selektora jest taki:

```
$('.selektor').metoda()
```

Jak widać, selektor składa się z **aliasu** biblioteki jQuery (`jQuery` lub `$`), wybranego elementu modelu DOM zapisanego w cudzysłowach lub apostrofach i ujętego w nawiasy okrągłe oraz z metody biblioteki jQuery. Metodą biblioteki może być na przykład definicja stylów albo animowanie elementów na stronie.

Przykład 4.3

```
$(function() {
    $('p').css('font-family', 'Arial, Verdana');
});
```

W podanym przykładzie zapis `$('p')` to selektor jQuery dla wszystkich elementów `p` na stronie. `$()` to funkcja, która zwraca nowy obiekt jQuery. Wywołana funkcja `css()` to metoda tego obiektu. Powoduje ona ustawienie właściwości `font-family` dla wszystkich elementów `p` na stronie.

Poniższy przykład pokazuje wykorzystanie biblioteki do ukrywania i pokazywania tekstu po kliknięciu napisu jQuery.

Przykład 4.4

Kod HTML:

```
<!DOCTYPE HTML>

<html>

<head>

<title>jQuery</title>

<meta charset="UTF-8">

<style type="text/css">

div {

    width: 400px;

    background-color: #dff;

    display: none;

}

p {

    background-color: #aaa;

    position: absolute;

    left: 130px;

    font-weight: bold;

    padding: 5px;

}

</style>

<script src="js/jquery-3.4.1.js"></script>

<script src="js/skrypt4.js"></script>

</head>
```

```

<body>

<p id="bibl">jQuery – Kliknij!</p><br><br><br>

<div id="bibl_info">

Biblioteka jQuery zawiera różnorodne funkcje dające możliwość realizacji
często spotykanych zadań (animacje, obsługa zdarzeń, tworzenie efektów
na stronie internetowej). Jej głównymi zaletami są: łatwość użycia,
niezależność od przeglądarki i małe rozmiary otrzymanego kodu.

</div>

</body>

</html>

```

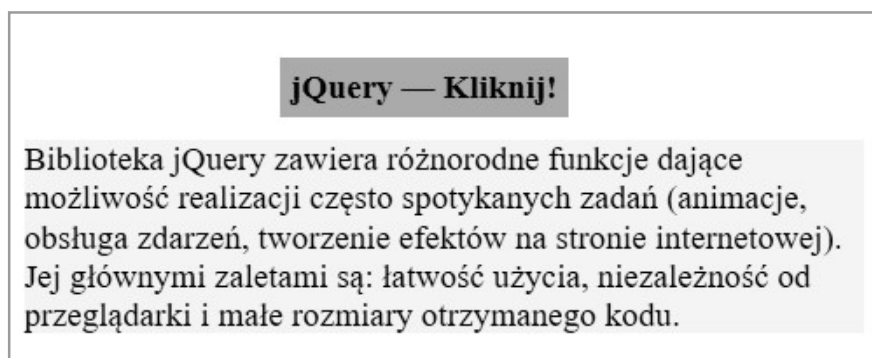
Plik ze skrypcem *skrypt4.js*:

```

$(function() {
    $('#bibl').click(function() {
        $('#bibl_info').toggle("slow");
    });
});

```

W podanym wyżej przykładzie biblioteka pobrana ze strony została zapisana w folderze *js*. W pliku *skrypt4.js* został zapisany skrypt utworzony zgodnie z podanym wcześniej szablonem. Po załadowaniu drzewa DOM zostanie wywołana funkcja anonimowa wykonująca polecenia na elementach dokumentu HTML. W wyniku wykonania kodu po kliknięciu napisu *jQuery — Kliknij!* wyświetli się opis wybranego hasła, a po ponownym kliknięciu napisu opis zostanie ukryty (rysunek 4.1).



Rysunek 4.1. Po kliknięciu napisu jQuery — Kliknij! został wyświetlony opis tego hasła

W kodzie HTML zostały zdefiniowane identyfikatory dla akapitu `id="bibl"` oraz dla znacznika `div id="bibl_info"`. Zapis funkcji `$('#bibl')` oznacza pobranie elementu o identyfikatorze `bibl` i po wystąpieniu zdarzenia `click` wywołanie funkcji obsługującej to zdarzenie. W wyniku wystąpienia zdarzenia dla elementu o identyfikatorze `bibl_info` zostanie wykonana metoda `toggle()`, która na przemian ukrywa i pokazuje wybrany element. Metodę `toggle()` można uzupełnić o parametr określający szybkość zwijania i rozwijania tekstu. Parametr ten może przyjąć jedną z wartości:

- `slow` — wolne przewijanie:

```
$('#bibl_info').toggle("slow");
```

- `normal` — normalna szybkość przewijania,
- `fast` — szybkie przewijanie,
- *dowolna wartość* — szybkość przewijania w milisekundach:

```
$('#bibl_info').toggle(3000);
```

Zmiana zawartości pliku *skrypt4.js* spowoduje zmianę działania strony.

Przykład 4.5

```
$(function() {
    $('#bibl').click(function() {
        $('*').css('font-family', 'Arial, Verdana');
    });
});
```

W podanym skrypcie za pomocą metody `css()` ustawione zostały właściwości `font-family` dla wszystkich elementów strony.

Przykład 4.6

```
$(function() {
    $('#bibl').click(function() {
        $('p').css('background', 'red');
    });
});
```

W podanym skrypcie przy użyciu metody `css()` został zmieniony kolor tła dla wszystkich znaczników `<p>`.

Ćwiczenie 4.1

Zaprojektuj na stronie internetowej blok zawierający kilka haseł związanych z czwartą rewolucją przemysłową. Hasła mają być widoczne na stronie, natomiast opis danego hasła ma się pojawiać po kliknięciu tego hasła. Zdefiniuj odpowiednio style dla wyświetlanych bloków informacji.

4.1.3. Filtry selektorów jQuery

Za pomocą selektorów wybieramy określone elementy modelu DOM. Jeżeli użyjemy filtrów, otrzymamy selektory, które zwracają elementy spełniające wskazane kryteria.

Filtry jQuery rozpoczynają się od znaku `:`, po którym występuje nazwa zastosowanego filtra. Wybrane filtry jQuery zostały przedstawione w tabeli 4.2.

Tabela 4.2. Filtry biblioteki jQuery

Selektor	Znaczenie
:first	pierwszy element z wyszukiwanego zbioru
:last	ostatni element z wyszukiwanego zbioru
:eq(indeks)	wybrany element wyszukiwanego zbioru (indeksowanie rozpoczyna się od 0)
:odd	nieparzyste elementy wyszukiwanego zbioru
:even	parzyste elementy wyszukiwanego zbioru
:contains(tekst)	element zawierający podany tekst
:hidden	elementy ukryte oraz pola typu <code><input type="hidden"></code>
:visible	elementy widoczne

Jedną z metod ułatwiających odczyt wierszy w tabeli lub elementów listy jest naprzemienne ich kolorowanie. Polega to na nadaniu innego koloru wierszom parzystym i nieparzystym. Do tego celu mogą zostać użyte filtry `:even` i `:odd`.

Przykład 4.7

Kod HTML:

```
<!DOCTYPE HTML>
<html>
<head>
<title>Filtr even-odd</title>
<meta charset="UTF-8">
<script src="js/jquery-3.4.1.js"></script>
<script src="js/skrypt7.js"></script>
</head>
<body>
<table>
<tr><td width=100> Produkt</td><td width=250> Opis</td><td width=50> Cena</td></tr>
<tr><td> Olej</td><td> Olej rzepakowy</td><td> 6 zł</td></tr>
<tr><td> Cukier</td><td> Cukier brązowy</td><td> 4 zł</td></tr>
<tr><td> Mąka</td><td> Mąka krupczatka</td><td> 3 zł</td></tr>
<tr><td> Sól</td><td> Sól warzona</td><td> 2 zł</td></tr>
<tr><td> Masło</td><td> Masło śmietankowe</td><td> 4 zł</td></tr>
```



```

<tr><td> Chleb</td><td> Chleb razowy</td><td> 3 zł</td></tr>

</table>

</body>

</html>

```

Kod jQuery:

```

$(function() {

    $('tr:even').css('background', '#B0B0B0');

    $('tr:odd').css('background', '#EEE');

});

```

Wynik wykonania kodu został pokazany na rysunku 4.2.

Produkt	Opis	Cena
Olej	Olej rzepakowy	6 zł
Cukier	Cukier brązowy	4 zł
Mąka	Mąka krupczatka	3 zł
Sól	Sól warzona	2 zł
Masło	Masło śmietankowe	4 zł
Chleb	Chleb razowy	3 zł

Rysunek 4.2. Użycie filtrów `:even` i `:odd`

Tę samą metodę możemy zastosować również dla listy nieuporządkowanej. Standardem jest wykorzystywanie listy nieuporządkowanej do tworzenia menu na stronę internetową. Odróżnianie elementów listy poprzez ich naprzemienne kolorowanie może pomóc użytkownikom w poruszaniu się po menu.

Przykład 4.8

Kod HTML:

```

<!DOCTYPE HTML>

<html>

<head>

<title>Menu</title>

<meta charset="UTF-8">

<style type="text/css">

.tlo1 {

    background: #EEE;

}

.tlo2 {

    background: #B0B0B0;

}

```

```

#menu {
    width: 100px;
}
ul {
    display: block;
    list-style-type: none;
    margin: 0;
    padding: 0;
}
</style>
<script src="js/jquery-3.4.1.js"></script>
<script src="js/skrypt8.js"></script>
</head>
<body>
<div id="menu">
<p>Menu</p>
<ul>
    <li> Start</li>
    <li> Aktualności</li>
    <li> Szkolenia</li>
    <li> Usługi</li>
</ul>
</div>
</body>
</html>

```

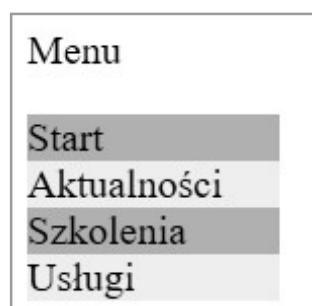
Kod jQuery (*skrypt8*):

```

$(function() {
    $("ul li:even").addClass('tlo2');
    $("ul li:odd").addClass('tlo1');
});

```

W kodzie jQuery zostały pobrane wszystkie odnośniki do elementów listy. Elementy parzyste ("ul li:even") zostały za pomocą metody `addClass('tlo2')` przypisane do klasy `tlo2`, a elementy nieparzyste do klasy `tlo1`. Klasy `tlo1` i `tlo2` zostały zdefiniowane w arkuszach stylów CSS. Metoda `addClass()` pozwala na dodanie wybranych selektorów do istniejącej już klasy. W wyniku wykonania kodu elementy parzyste listy zostaną dodane do klasy `tlo2`, a nieparzyste do klasy `tlo1` (rysunek 4.3).



Rysunek 4.3.
Naprzemienne
kolorowanie
elementów menu

Metody związane z używaniem stylów CSS zostały pokazane w tabeli 4.3.

Tabela 4.3. Metody CSS

Nazwa metody	Opis
<code>.css()</code>	umożliwia pobranie lub ustawienie właściwości dla dowolnego elementu
<code>.addClass()</code>	umożliwia dodanie klasy CSS do dowolnego elementu
<code>.hasClass()</code>	umożliwia testowanie, czy element ma klasę
<code>.removeClass()</code>	umożliwia usunięcie klasy CSS z dowolnego elementu
<code>.toggleClass()</code>	umożliwia dodanie i usunięcie klas CSS z dowolnego elementu

Gdyby w dokumencie HTML istniały jeszcze inne listy, to wszystkie elementy tych list zostałyby przypisane do klas `tlo1` lub `tlo2`. Aby tego uniknąć, w skrypcie można dodać informację, że przypisanie do klas dotyczy tylko listy, dla której został zdefiniowany identyfikator `id="menu"`. Natomiast w kodzie HTML musi nastąpić przypisanie tego identyfikatora do wybranej listy.

Przykład 4.9

```
$(function() {
    $("ul#menu li:even").addClass('tlo2');
    $("ul#menu li:odd").addClass('tlo1');
});
```

Ćwiczenie 4.2

Utwórz menu dla strony internetowej, w którym pozycja menu wybrana przez kliknięcie myszą będzie podświetlana. Wykorzystaj kod z poprzednich przykładów oraz zdefiniowane przez siebie style elementów menu.



4.2. Zdarzenia biblioteki jQuery

4.2.1. Opis zdarzeń

Obsługa zdarzeń to jedno z podstawowych działań wykonywanych na stronie internetowej. Biblioteka jQuery umożliwia użycie wszystkich zdarzeń języka JavaScript. Jedyna różnica polega na tym, że z nazw zdarzeń został usunięty prefiks `on`.

Przykład 4.10

```
$(function() {
    $('.blok').click(function() {
        alert("Uwaga!");
    });
});
```

Wybrane zdarzenia biblioteki jQuery zostały przedstawione w tabeli 4.4.

Tabela 4.4. Zdarzenia biblioteki jQuery

Nazwa zdarzenia	Opis
<code>click()</code>	występuje po kliknięciu elementu myszą
<code>mouseout()</code>	występuje po usunięciu kursora myszy z wybranego elementu
<code>mouseover()</code>	występuje po umieszczeniu kursora myszy w obrębie wybranego elementu
<code>hover()</code>	występuje po ustawieniu kursora myszy nad wybranym elementem

4.2.2. Zdarzenia myszy

Przy użyciu zdarzeń obsługiwanych przez bibliotekę jQuery można animować menu utworzone na stronie internetowej, na przykład przez podświetlenie elementu, nad którym pojawi się kursor myszy (rysunek 4.4). Po usunięciu kursora zostanie poprzedni sposób wyświetlania elementu. Do tego celu używa się zdarzenia `hover()`. Po najechnaniu myszą na element listy nas elementu (`addClass()`), a po przesunięciu myszy poza element z elementu (`removeClass()`).

Przed przygotowaniem skryptu, który będzie powodował podświetlenie elementu, w dokumencie HTML w arkuszu stylów należy zdefiniować nową klasę (na przykład `li.wybor`). Po najechnaniu myszą na element klasa `li.wybor` zostanie do niego dodana, a po odsunięciu myszy zostanie usunięta.

Przykład 4.11

Kod HTML:

```
<!DOCTYPE HTML>
<html>
<head>
<title>Menu</title>
<meta charset="UTF-8">
<style type="text/css">
li.wybor {
    background: gray;
    color: #FFF;
}
#menu {
    width: 120px;
```

Menu

Start

Aktualności

Szkolenia

Usługi

Rysunek 4.4. Podświetlenie elementu, nad którym pojawia się kursor myszy

```

    }
    ul {
        display: block;
        list-style-type: none;
        margin: 0;
        padding: 0;
    }
</style>
<script src="js/jquery-3.4.1.js"></script>
<script src="js/skrypt11.js"></script>
</head>
<body>
<p>Menu</p>
<ul id="menu">
    <li> Start</li>
    <li> Aktualności</li>
    <li> Szkolenia</li>
    <li> Usługi</li>
</ul>
</body>
</html>

```

Kod jQuery:

```

$(function() {
    $("ul#menu li").hover(function() {
        $(this).addClass('wybor');
    }, function() {
        $(this).removeClass('wybor');
    });
});

```

Pierwsza funkcja w podanym kodzie dodaje klasę do elementu, druga ją usuwa. Użyte w kodzie słowo kluczowe `this` stanowi odwołanie do bieżącego obiektu (`"ul#menu li"`) wykonywanej funkcji.

4.2.3. Zdarzenia formularza

Biblioteka jQuery oferuje grupę zdarzeń, które są związane z obsługą formularza. Mogą one zostać wykorzystane na przykład do sprawdzania poprawności wypełnienia formularza, do powiadomienia użytkownika o jakimś działaniu lub do określonej akcji, gdy



użytkownik opuści pole. Zdarzenia formularza biblioteki jQuery zostały przedstawione w tabeli 4.5.

Tabela 4.5. Zdarzenia formularza

Zdarzenie	Opis
<code>change()</code>	występuje w momencie edytowania zawartości pola
<code>focus()</code>	występuje w momencie aktywowania pola tekstowego (wybranie pola klawiszem <i>Tab</i> lub jego zaznaczenie)
<code>focusin()</code>	występuje w momencie aktywowania pola tekstowego znajdującego się wewnątrz elementu
<code>focusout()</code>	występuje w momencie dezaktywowania elementu
<code>blur()</code>	występuje w momencie dezaktywowania elementu na rzecz innego elementu
<code>select()</code>	występuje w momencie zaznaczenia tekstu wewnątrz elementu
<code>submit()</code>	występuje w momencie wysłania formularza po kliknięciu przycisku typu <code>submit</code>
<code>reset()</code>	występuje w momencie zresetowania formularza po kliknięciu przycisku typu <code>reset</code>

4.3. Zastosowanie biblioteki jQuery na stronie internetowej

Jednym ze sposobów tworzenia dynamicznych elementów stron internetowych jest wykorzystanie biblioteki jQuery napisanej w języku JavaScript. Biblioteka pozwala na prostą manipulację elementami na stronie, pracę ze zbiorami elementów oraz udostępnia wiele efektów wizualnych.

4.3.1. Pokazywanie i ukrywanie treści

Tak zwana harmonijka to dobry sposób na prezentowanie lub ukrywanie treści na stronie internetowej. Może zostać również wykorzystana jako rozwijane menu nawigacyjne (rysunek 4.5). W tak opracowanym narzędziu hasła dla treści, które będą wyświetlane lub ukrywane, są zorganizowane w postaci menu. Kliknięcie wybranej opcji pozwala wyświetlić ukrytą treść. Po kliknięciu innej opcji wyświetlana treść zostaje ukryta i wyświetla się inna (rysunek 4.6).

Język HTML
Podstawowym elementem języka HTML jest znacznik. Znaczniki są poleceniami umieszczonymi w nawiasach ostrych. Informują one przeglądarkę o wyglądzie otwieranej strony oraz o strukturze tekstu umieszczonego na niej.
Kaskadowe arkusze stylów CSS
Systemy zarządzania treścią – CMS
JavaScript
jQuery

Rysunek 4.5.

Menu nawigacyjne

Język HTML
Kaskadowe arkusze stylów CSS
Systemy zarządzania treścią – CMS
JavaScript
JavaScript odgrywa coraz większą rolę w projektowaniu interaktywnych stron internetowych. Jego możliwości są bardzo duże, od prostej manipulacji danymi, przez dynamiczne modelowanie struktury strony, obsługę rozszerzeń multimedialnych, aż po tworzenie odrębnych aplikacji.
jQuery

Rysunek 4.6.

Wyświetlenie treści ukrytej w menu

Przykład 4.12

Kod HTML:

```

<!DOCTYPE HTML>

<html>
<head>
<meta charset="UTF-8">
<title>Rozwijany blok</title>
<script src="js/jquery-3.4.1.js"></script>
<script src="js/skrypt12.js"></script>
<style type="text/css">
body {
    margin: 10px auto;
    width: 600px;
}
.blok {
    width: 450px;
    border-bottom: solid 1px #b0b0b0;
}
.blok h3 {
    background: #c6d2f2;
    padding: 8px 15px;
    margin: 0;
    font: bold 12pt Arial, sans-serif;
    border: solid 1px #b0b0b0;
    border-bottom: none;
    cursor: pointer;
}

```

```
.blok h3:hover {
    background-color: #88a3e8;
}

.blok h3.aktywny {
    background-color: #880000;
}

.blok p {
    background: #e1e6f4;
    margin: 0;
    padding: 10px 15px 20px;
    border-left: solid 1px #c4c4c4;
    border-right: solid 1px #c4c4c4;
    font: 10pt Arial, sans-serif;
    display: none;
}
</style>
</head>
<body>
<div class="blok">
    <h3>Język HTML</h3>

    <p>Podstawowym elementem języka HTML jest znacznik.
Znaczniki są poleceniami umieszczonymi w nawiasach ostrych.
Informują one przeglądarkę o wyglądzie otwieranej strony
oraz o strukturze tekstu umieszczonego na niej. </p>

    <h3>Kaskadowe arkusze stylów CSS</h3>

    <p>Język CSS określa układ graficzny dokumentu HTML: parametry
czcionki, wysokość i szerokość obrazków, ich położenie,
rodzaj tła itp. Wszystkie polecenia dotyczące formatowania
powinny zostać umieszczone w oddzielnym pliku (arkuszu) i być
powiązane z elementami zdefiniowanymi w kodzie HTML.</p>

    <h3>Systemy zarządzania treścią-CMS</h3>

    <p>Systemy zarządzania treścią (CMS, ang. Content Management
System) to aplikacje internetowe, które pozwalają na łatwe tworzenie
i modyfikowanie oraz rozbudowę strony internetowej. Systemy CMS generują
przy wykorzystaniu odpowiednich szablonów gotowe strony internetowe.</p>

    <h3>JavaScript</h3>

    <p>JavaScript odgrywa coraz większą rolę w projektowaniu
interaktywnych stron internetowych. Jego możliwości
są bardzo duże, od prostej manipulacji danymi, przez
```



```

dynamiczne modelowanie struktury strony, obsługę rozszerzeń
multimedialnych, aż po tworzenie odrębnych aplikacji.</p>

<h3>jQuery</h3>

<p>Biblioteka jQuery to framework zawierający różnorodne
funkcje, za pomocą których można realizować często spotykane
zadania (na przykład animacje, obsługę zdarzeń, tworzenie efektów
na stronie internetowej). Jej głównymi zaletami są: łatwość użycia,
niezależność od przeglądarki i małe rozmiary otrzymanego kodu.</p>
</div>
</body>
</html>

```

Kod jQuery:

```

$(function() {
    $(".blok h3").eq(0).addClass("aktywny");
    $(".blok p").eq(0).show();

    $(".blok h3").click(function() {
        $(this).next("p").slideToggle("slow")
        .siblings("p:visible").slideUp("slow");
        $(this).toggleClass("aktywny");
        $(this).siblings("h3").removeClass("aktywny");
    });
});

```

W podanym wyżej przykładzie kodu HTML zawartość strony zorganizowano w postaci bloku `<div class="blok">`. Opcje menu zostały zdefiniowane w postaci nagłówka `<h3>`, a wyświetlana treść w postaci akapitu `<p>`.

Następnie zdefiniowane zostały style: dla klasy `.blok` przypisanej do bloku `<div>`, dla nagłówka `h3` (`.blok h3`), dla akapitu (`.blok p`) oraz dla nagłówka `h3`, gdy będzie wskazany myszą (`.blok h3:hover`).

Natomiast zapis w skrypcie `js $(".blok h3").eq(0).addClass("aktywny")` odpowiada za pobranie z klasy `.blok` elementu `h3` o indeksie 0 oraz dodanie go do klasy `aktywny`. Metoda `eq(indeks)` wybiera element, który jest zgodny z podanym indeksem. W tym wypadku jest to pierwszy nagłówek `h3`, który po przypisaniu mu klasy `aktywny` zmieni swój wygląd. Instrukcja `$(".blok p").eq(0).show()` pobierze pierwszy element `p` i wykona na nim metodę `show()`. Jeżeli element jest ukryty, metoda ta zmienia wartość atrybutu `display` na `"block"`. Jeśli wybrany element został już wyświetlony, metoda nie będzie powodowała żadnego działania.

Kolejny blok instrukcji odnosi się do sytuacji, gdy po kliknięciu określonej opcji powinien zostać wyświetlony tekst znajdujący się za wybranym nagłówkiem.



Polecenie `$(".blok h3").click(function() {})` mówi, że po kliknięciu wybranego nagłówka powinna zostać wywołana funkcja, która będzie wyświetlała sąsiadujący z nim akapit: `$(this).next("p").slideToggle("slow").siblings("p:visible").slideUp("slow").` Metoda `next()` pobiera następny po wskazanym pasujący element typu rodzeństwo: `siblings()`. Metoda `slideToggle("slow")` wyświetla lub ukrywa wybrany element. Parametr "slow" określa szybkość ukrywania lub pokazywania elementu.

W wyniku działania skryptu dla wskazanego elementu (h3) zostanie pobrany następny element p ("p:visible"), który się wyświetli. Widoczny siostrzany (`siblings()`) element p zostanie ukryty (`slideUp()`). Do wybranego elementu h3 zostanie dodana klasa aktywny: `$(this).toggleClass("aktywny")`, a siostrzany element h3 utraci klasę aktywny.

Pominięcie w skrypcie *js* bloku

```
$(".blok h3").eq(0).addClass("aktywny");  
$(".blok p").eq(0).show();
```

spowoduje wyświetlenie menu w postaci pokazanej na rysunku 4.7.

Język HTML
Kaskadowe arkusze stylów CSS
Systemy zarządzania treścią – CMS
JavaScript
jQuery

Rysunek 4.7. Menu nawigacyjne bez bloku z treścią

4.3.2. Proste animacje

Strony internetowe z pokazami slajdów, animowanymi menu lub innymi animacjami często są tworzone w języku JavaScript. Biblioteka jQuery korzysta z efektów dostępnych w tym języku, a konfigurowanie ich przy użyciu jQuery jest dużo prostsze.

Do pokazywania i ukrywania elementów często wykorzystuje się metody `show()` oraz `hide()`. Można je wywoływać z parametrem określającym czas trwania akcji. Parametrem może być wartość `slow`, `normal` lub `fast`.

Kolejna metoda to `slideToggle()`. Jest ona odpowiednikiem metod `show()` oraz `hide()`. Automatycznie sprawdza, czy wybrany element jest ukryty, czy nie, i wyświetla go lub ukrywa.

Następne metody to `fadeIn()` oraz `fadeOut()`. Działają prawie tak samo jak `show()` oraz `hide()`. Różnica polega na tym, że ukazują i ukrywają element, zmieniając stopniowo jego przezroczystość.

Metody `slideDown()` oraz `slideUp()` też pokazują lub ukrywają elementy, ale efektem animacji jest zsuwanie i rozsuwanie elementu. Przykład wykorzystania tych metod do animowania treści strony został pokazany poniżej, a wynik interpretacji kodu widać na rysunku 4.8.

Przykład 4.13

```
<!DOCTYPE HTML>
<html>
<head>
<title>Ukrywanie napisu</title>
<meta charset="UTF-8">
<style>
.blok {
    background-color: #c0c0c0;
    color: #000;
    margin: 15px;
    padding: 20px;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 40px;
    width: 300px;
    font: bold 24pt Arial, sans-serif;
}
</style>
<script src="js/jquery-3.4.1.js"></script>
<script>
$(function() {
    $("button").click(function() {
        $("#p1").css("color", "red")
        .slideUp(2000)
        .slideDown(2000);
    });
});
</script>
</head>
```



Rysunek 4.8.

Pokazanie i ukrywanie napisu

```
<body>

<div id="blok" class="blok">

<p id="p1">Biblioteka jQuery!</p>

</div>

<button>Kliknij</button>

</body>

</html>
```

Ćwiczenie 4.3

Wstaw na stronę internetową zdjęcie. Utwórz animację, która będzie powodowała, że gdy użytkownik najedzie myszą na zdjęcie, zniknie ono w sposób podobny do tego, jak znikał tekst w przykładzie 4.13. Po odsunięciu myszy zdjęcie będzie powoli pojawiało się na stronie.

4.3.3. Animacje zaawansowane

Zmiana rozmiaru elementu

Najbardziej zaawansowaną metodą służącą do animowania elementów strony internetowej jest metoda `animate()`. Metoda ta działa na właściwościach stylów CSS, ale tylko tych, które przyjmują wartości liczbowe (na przykład `margin`, `padding`, `top`). Pierwszy parametr metody to tablica asocjacyjna z właściwościami stylów, czyli animowane właściwości. Drugi to czas wykonywania animacji (parametr opcjonalny). Określany jest wartościami `slow`, `normal`, `fast` lub wartościami podanymi w milisekundach. Następny parametr `easing` (opcjonalny) określa sposób przejścia animacji (liniowy — `linear` — lub wahadłowy — `swing`). Domyślnie w animacjach stosowane jest przejście liniowe. Ostatni parametr (opcjonalny) to funkcja zwrotna, czyli funkcja wywoływana po zakończeniu animacji.

Przykład 4.14

```
<!DOCTYPE HTML>

<html>

<head>

<style type="text/css">

#blok {

    width: 100px;

    height: 100px;

    background: #c6d2f2;

    position: absolute;

}

</style>
```

```

<script src="js/jquery-3.4.1.js"></script>
<script>
$(function(){
    $("button").click(function(){
        $("#blok").animate({
            width: "500px", height: "500",
        }, 1500);
    });
});
</script>
</head>
<body>
<button>Start animacji</button>
<div id="blok">
</div>
</body>
</html>

```

W podanym wyżej przykładzie funkcja `animate()` została wywołana dla elementu `<div>` o identyfikatorze `id="blok"`. Zmieniającymi się parametrami są wysokość i szerokość elementu. Czas trwania animacji to 1500 milisekund.

Po kliknięciu przycisku *Start animacji* wynikiem wykonania kodu będzie zmiana rozmiaru bloku o identyfikatorze `id="blok"` (rysunki 4.9 i 4.10).



Rysunek 4.9.

Początkowy
rozmiar bloku



Rysunek 4.10. Zmieniony rozmiar bloku

Ćwiczenie 4.4

Umieść na stronie internetowej dowolny tekst. Wykorzystując metodę `animate()`, utwórz animację, która będzie powodowała, że gdy użytkownik najedzie myszą na tekst, tekst ten kilkakrotnie się powiększy (stopniowo), a po odsunięciu myszy wróci do początkowego rozmiaru.

Zmiana rozmiaru i położenia elementu

Domyślnie każdy element dokumentu HTML ma określoną pozycję, która nie może być zmieniona. Aby móc nią manipulować, należy najpierw ustawić właściwość `position` na wartość `relative`, `fixed` lub `absolute`. Jednocześnie można manipulować wieloma właściwościami elementu.

Przykład 4.15

```
<!DOCTYPE HTML>

<html>

<head>

<style type="text/css">

#blok {

    width: 100px;

    height: 100px;

    background: #c6d2f2;

    position: absolute;

    padding: 10px;

}

</style>

<script src="js/jquery-3.4.1.js"></script>

<script>

$(function(){

    $("#blok").mouseover(function() {

        $(this).animate({

            width: "+=300px", height: "+=300px", left: "250px", opacity: "0.5",

        }, 1500);

    });

    $("#blok").mouseout(function() {

        $(this).animate({

            width: "100px", height: "100px", left: "10px", opacity: "1",

        }, 1500);

    });

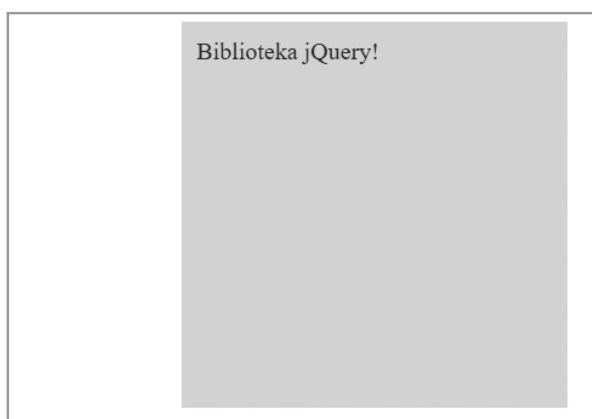
});
```

```

});
</script>
</head>
<body>
<div id="blok">
Biblioteka jQuery!
</div>
</body>
</html>

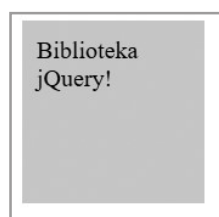
```

W podanym przykładzie wielkość bloku zmienia się w zależności od tego, czy kursor myszy znajduje się nad nim. Metoda `animate()` została zadeklarowana dwukrotnie i zostanie wykonana w zależności od występującego zdarzenia. Gdy najedziemy kursorem myszy na element, zmieni on swoje położenie oraz rozmiar (rysunek 4.11). Po odsunięciu kursora myszy element ten powróci do pierwotnego położenia i rozmiaru (rysunek 4.12). Przy definiowaniu właściwości wykorzystana została możliwość ustawienia wartości względnych (`width: "+=300px"`, `height: "+=300px"`). Gdy podaje się wartości względne, przed podaną wartością należy umieścić `+=` lub `-=`.



Rysunek 4.11.

Zmiana rozmiaru i położenia bloku



Rysunek 4.12.

Powrót do pierwotnego położenia i rozmiaru

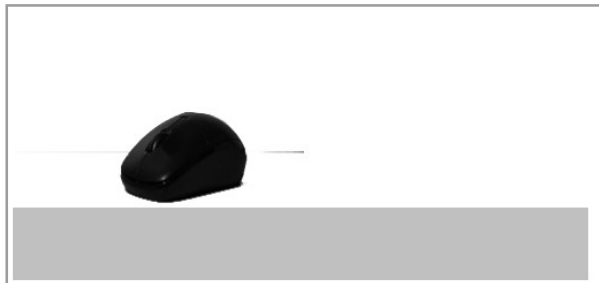
jQuery tworzy kolejki dla animacji. Dlatego jeżeli w kodzie występuje wiele wywołań tej samej metody, zostanie utworzona kolejka wywołań. Metody uruchamiane są jedna po drugiej. W podanym przykładzie kilkakrotne najechanie kursorem myszy na obiekt i odsunięcie kursora spowoduje kilkakrotne powtórzenie animacji.

UWAGA

Przy użyciu metody `animate()` można przetwarzać większość właściwości CSS. Należy pamiętać, że nazwy właściwości nie mogą zawierać znaku `-` (łącznik). Nazwa taka powinna mieć postać na przykład `paddingleft` (a nie `padding-left`).

Ćwiczenie 4.5

Umieść na stronie internetowej prostą grafikę, na przykład obrazek myszy. Wykorzystując metodę `animate()`, utwórz animację, która będzie powodowała, że po najechaniu myszą na grafikę obrazek zostanie przesunięty w inne miejsce, a następnie wróci do położenia początkowego (rysunki 4.13 i 4.14).



Rysunek 4.13.

Początkowe położenie elementu animowanego



Rysunek 4.14.

Końcowe położenie elementu animowanego

Ćwiczenie 4.6

Utwórz animacje wybranych elementów swojej strony internetowej. W pracy wykorzystaj metodę `animate()` oraz zmianę atrybutów stylów CSS.

Wyświetlanie podpowiedzi do treści

Wyświetlanie podpowiedzi w postaci dymków informacyjnych nad odnośnikami lub elementami formularza pozwala na łatwiejsze poruszanie się w obrębie strony. Podpowiedzi mogą być instrukcjami, jak dalej postępować, lub mogą wyjaśniać pewne terminy i zawierać istotne informacje.

Przykład 4.16

Kod HTML:

```
<!DOCTYPE HTML>

<html>

<head>

<meta charset="UTF-8">

<title>Dymki</title>

<script src="js/jquery-3.4.1.js"></script>

<script src="js/skrypt16.js"></script>

<style type="text/css">

body {

    background: #c0c0c0;

    margin: 10px auto;

    width: 570px;

    font: 75%/120% Arial, sans-serif;
```



```
}  
.menu {  
  margin: 100px 0 0;  
  padding: 0;  
  list-style: none;  
}  
.menu li {  
  padding: 0;  
  margin: 0 2px;  
  float: left;  
  position: relative;  
  text-align: center;  
}  
.menu a {  
  padding: 14px 10px;  
  display: block;  
  color: #000000;  
  width: 144px;  
  text-decoration: none;  
  font-weight: bold;  
  background: url(menu.png) no-repeat center;  
}  
.menu li p {  
  background: url(opis.png) no-repeat;  
  width: 118px;  
  height: 54px;  
  position: absolute;  
  top: -85px;  
  left: -10px;  
  text-align: center;  
  padding: 10px 10px 10px;  
  font-style: normal;  
  z-index: 2;  
  display: none;  
}  
</style>  
</head>
```

```
<body>
<ul class="menu">
  <li>
    <a href="http://www.sejm.gov.pl/">Sejm Rzeczypospolitej Polskiej</a>
    <p>Wszelka władza społeczności ludzkiej początek swój bierze z woli
narodu</p>
  </li>
  <li>
    <a href="http://www.men.gov.pl/">Ministerstwo Edukacji Narodowej</a>
    <p>Takie będą Rzeczypospolite, jakie ich młodzieży chowanie</p>
  </li>
  <li>
    <a href="http://www.cke.edu.pl/">Centralna Komisja Egzaminacyjna</a>
    <p>Edukacja skuteczna, przyjazna i nowoczesna</p>
  </li>
</ul>
</body>
</html>
```

Kod jQuery:

```
$(function(){
  $(".menu a").hover(function() {
    $(this).next("p").animate({opacity: "show", top: "-75"}, "slow");
  }, function() {
    $(this).next("p").animate({opacity: "hide", top: "-85"}, "fast");
  });
});
```

Na stronie internetowej umieszczono kilka elementów menu (rysunek 4.15). Po wybraniu elementu pojawi się okienko z podpowiedzią (rysunek 4.16).



Rysunek 4.15. Menu umieszczone na stronie internetowej



Rysunek 4.16. Wyświetlenie podpowiedzi do menu

Działanie kodu jQuery jest następujące: po wskazaniu myszą elementu menu `$ (" .menu a") .hover` zostanie wywołana funkcja. Funkcja ta wybierze następny po wskazanym element `p` i uruchomi dla niego metodę `animate ()`. Metoda ta wyświetli zawartość wybranego elementu `p` w miejscu przesuniętym w górę w stosunku do jego pierwotnego położenia i ukryje wcześniej wyświetlany element `p`.

Ćwiczenie 4.7

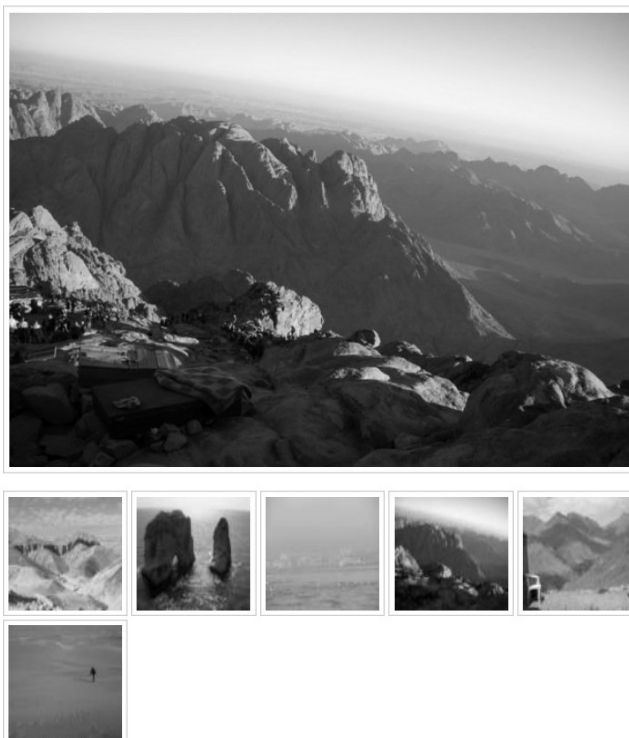
Wykorzystując przykład 4.16 i poznane narzędzia projektowania stron internetowych, zaprojektuj dla swojej strony internetowej menu z podpowiedziami w postaci dymków pojawiających się po ustawieniu myszy nad wybraną opcją tego menu.

4.3.4. Pokaz zdjęć

Z wykorzystaniem biblioteki jQuery można w bardzo prosty sposób tworzyć animowane pokazy slajdów. Najczęściej takie pokazy zawierają galerię miniatur zdjęć — wybrane przez użytkownika zdjęcie jest wyświetlane w pełnych rozmiarach. Sposobów prezentowania zdjęć jest wiele. Na rysunku 4.17 został przedstawiony jeden z nich. W dolnej części znajduje się galeria miniatur obrazów, w górnej wyświetlany jest aktualnie wybrany obraz.

Do utworzenia podobnego pokazu potrzebne są miniatury zdjęć oraz zdjęcia w oryginalnej wielkości. Aby je przygotować, można wykorzystać na przykład program Web Album Generator, który generuje miniatury zdjęć oraz tworzy zdjęcia o określonych rozmiarach, najczęściej mniejszych od rozmiarów zdjęć oryginalnych.

Pokaz zdjęć (Zdjęcie 4)



Rysunek 4.17.
Pokaz zdjęć na stronie internetowej

Po przygotowaniu zdjęć i umieszczeniu ich w folderze *img* istotne jest takie zdefiniowanie stylów CSS, aby cały pokaz dobrze prezentował się na stronie internetowej.

Przykład 4.17

```
<!DOCTYPE HTML>

<head>

<meta charset="UTF-8">

<title>Pokaz zdjęć</title>

<script src="js/jquery-3.4.1.js"></script>

<script>

$(function(){

    $("h2").append('<em></em>')

    $(".galeria a").click(function(){

        var do_zdj = $(this).attr("href");

        var do_tyt = $(this).attr("title");

        $("#zdjecie_d").attr({ src: do_zdj, alt: do_tyt });

        $("h2 em").html(" (" + do_tyt + ")"); return false;

    });

});

</script>

<style type="text/css">

body {

    margin: 20px auto;

    padding: 0;

    width: 580px;

    font: 75%/120% Arial, Helvetica, sans-serif;

}

h2 {

    font: bold 190%/100% Arial, Helvetica, sans-serif;

    margin: 0 0 .2em;

}

h2 em {

    font: normal 80%/100% Arial, Helvetica, sans-serif;

    color: #999999;

}

}
```

```

#zdjecie_d {
    border: solid 1px #ccc;
    width: 550px;
    height: 400px;
    padding: 5px;
}

.galeria img {
    border: solid 1px #ccc;
    width: 100px;
    height: 100px;
    padding: 4px;
}

.galeria img:hover {
    border-color: #FF9900;
}

</style>
</head>
<body>
<h2>Pokaz zdjęć</h2>
<p></p>
<p class="galeria">
    <a href="img/zdj1.jpg" title="Zdjęcie 1"></a>
    <a href="img/zdj2.jpg" title="Zdjęcie 2"></a>
    <a href="img/zdj3.jpg" title="Zdjęcie 3"></a>
    <a href="img/zdj4.jpg" title="Zdjęcie 4"></a>
    <a href="img/zdj5.jpg" title="Zdjęcie 5"></a>
    <a href="img/zdj6.jpg" title="Zdjęcie 6"></a>
</p>
</body>
</html>

```

W skrypcie jQuery została użyta metoda `append()`, która do nagłówka `h2` będzie dołączała tytuł wybranego zdjęcia. Metoda ta dodaje na końcu wybranego elementu kodu HTML zawartość podaną jako jej parametr. W skrypcie zostały również utworzone dwie zmienne. Pierwsza, `do_zdj`, przechowuje ścieżkę dostępu do pliku z wybranym zdjęciem. Druga, `do_tyt`, przechowuje tytuł zdjęcia umieszczony w odnośniku do niego. Zastosowana metoda `attr()` dla wybranego elementu pobiera wartość wskazanego

atributu i przypisuje go do zmiennej. W tym wypadku pobranymi atrybutami są `href` i `title`. Następnie elementowi `id="zdjecie_d"` (blok wyświetlający duże zdjęcie) zostały przy użyciu tej samej metody przypisane atrybuty: `src` równy zmiennej `do_zdj` oraz `alt="do_tyt"`. Ostatnia linia kodu odpowiada za pobranie za pomocą metody `html()` zawartości wskazanego elementu i umieszczenie za nim wartości zmiennej `do_tyt`, czyli tytułu wybranego zdjęcia.



4.4. Walidacja formularzy

Dla tworzonych na stronie internetowej formularzy powinna być stosowana walidacja. Walidacja pozwoli sprawdzić, czy użytkownik wypełnił wymagane pola formularza oraz czy wpisał prawidłowo dane. Tworzenie walidacji formularza z wykorzystaniem języka JavaScript zostało przedstawione w podrozdziale 3.11. Stanie się ono łatwiejsze dzięki zastosowaniu biblioteki jQuery oraz wtyczki Validation opracowanej dla tej biblioteki. Plik wtyczki można pobrać ze strony <https://jqueryvalidation.org/>. Po jego rozpakowaniu do walidacji formularza potrzebne będą tylko pliki: *jquery.validate.js*, zawierający walidator, oraz *messages_pl.js*, zawierający standardowe komunikaty walidacji zapisane w języku polskim. Pliki te kopiujemy i umieszczamy w dowolnym folderze z kodem strony. Na przykład w folderze *js* zawierającym bibliotekę jQuery tworzymy folder *validate* i w nim umieszczamy potrzebne pliki.

Informację o tym, że w kodzie HTML będzie wykorzystywana pobrana wtyczka, wpisujemy w sekcji `<head>` dokumentu, najlepiej po deklaracji pobrania biblioteki jQuery. Następnie dodajemy skrypt uruchamiający wtyczkę dla tworzonego formularza.

Przykład 4.18

```
<!DOCTYPE HTML>

<html>

<head>

<meta charset="UTF-8">

<title>Walidacja formularza</title>

<script src="js/jquery-3.4.1.js"></script>

<script src="js/validate/jquery.validate.js"></script>

<script src="js/validate/messages_pl.js"></script>

<script>

jQuery(function() {

    jQuery("#formularz").validate();

});

</script>

<style type="text/css">
```

```
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    font-weight: normal;
}

#formularz {
    width: 450px;
    background-color: #c0c0c0;
}

</style>
</head>
<body>

<form id="formularz" action="" method="post">
<fieldset>
<b><font size=4>Dane osobowe</font></b><br><br>
<b>Nazwisko:</b>
<div><input id="nazw" name="nazw" type="text" class="required" minlength=
"3"></div>
<b>Imię:</b>
<div><input id="imie" name="imie" type="text" class="required" minlength=
"2"></div>
<b>Zawód:</b>
<div><input id="zaw" name="zaw" type="text" class="required" minlength=
"3"></div>
<b>Adres e-mail:</b>
<div><input id="email" name="email" type="text" class="required email"
></div>
<b>Telefon:</b>
<div><input id="tel" name="tel" type="text" class="number" minlength="9"
></div><br>
<label><b> Pracuję: </b><input type="checkbox" name="opcje" id="opcje"
maxlength="1">
</label><br><br>
<b>Wiadomość:</b><br>
<div><textarea id="inf" cols="30" name="inf" rows="5" class="required"
minlength="5"></textarea></div><br><br>
```



```

 &nbsp; &nbsp; &nbsp;

</fieldset>
</form>
</body>
</html>

```

Efekt działania powyższego kodu pokazany jest na rysunku 4.18. Komunikaty pojawiają się automatycznie po błędnym wypełnieniu pól formularza.

Dane osobowe

Nazwisko:
 Proszę o podanie przynajmniej 3 znaków.

Imię:

Zawód:
 To pole jest wymagane.

Adres e-mail:
 Proszę o podanie prawidłowego adresu email.

Telefon:
 Proszę o podanie prawidłowej liczby.

Pracuję: ☒

Wiadomość:
 To pole jest wymagane.

Rysunek 4.18. Walidacja formularza

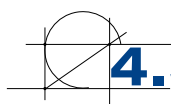
W podanym przykładzie w pierwszej linii kodu po znaczniku `<title>` zostało zapisane dołączenie biblioteki jQuery, następnie dołączenie wtyczki Validate, w kolejnej dołączenie pliku z komunikatami w języku polskim. Następne linie kodu to skrypt uruchamiający wtyczkę dla formularza o zdefiniowanym identyfikatorze `id="formularz"`. Sekcja `<body>` zawiera polecenia utworzenia formularza, który zostanie poddany walidacji. Pola formularza *Nazwisko:* i *Zawód:* są obowiązkowe i muszą zawierać przynajmniej trzy znaki. Pole *Imię:* jest również obowiązkowe i musi zawierać przynajmniej dwa znaki. Pole *Adres e-mail:* też jest obowiązkowe i powinno zostać sprawdzone pod kątem poprawnego adresu e-mail. W przypadku kolejnego pola, *Telefon:*, należy zweryfikować, czy zawiera cyfry i czy nie jest ich mniej niż dziewięć. Ostatnie pole, *Wiadomość:*, może zawierać dowolną informację, ale nie krótszą niż pięć znaków.

Weryfikacja formularza przy użyciu wtyczki Validation polega na wywołaniu funkcji dostępnych w walidatorze. Prawidłowość danych jest sprawdzana automatycznie i potwierdzana odpowiednim komunikatem.

W kodzie wtyczki dla pól, których zawartość jest wymagana, została zdefiniowana klasa `required` (`class="required"`). Jeżeli jest wymagana minimalna liczba wprowadzonych znaków, stosuje się atrybut `minlength` (na przykład `minlength="2"`). Do weryfikacji poprawności adresu e-mail jest wykorzystywana klasa `email` (`class="email"`), a do weryfikacji numeru telefonu klasa `number` (`class="number"`).

Ćwiczenie 4.8

Wykorzystując kod z przykładu 4.18, utwórz na swojej stronie internetowej formularz rejestracyjny z podstawowymi danymi użytkownika. Przy użyciu wtyczki `Validation` przeprowadź walidację formularza. Za pomocą kodu HTML i arkuszy CSS popraw wygląd utworzonego formularza.



4.5. Pytania i zadania

4.5.1. Pytania

1. Jakie zadania można realizować na stronie za pomocą biblioteki jQuery?
2. Podaj metody dołączania biblioteki jQuery do skryptu.
3. Co może być metodą biblioteki jQuery?
4. Jaką rolę odgrywają filtry jQuery?
5. Do jakich celów można w bibliotece jQuery wykorzystać obsługę zdarzeń?
6. Jak działa metoda `animate()`?
7. Do czego służy wtyczka `Validation`?

4.5.2. Zadania

Zadanie 1.

Zbuduj stronę internetową z informacjami na temat komputerów. W lewej części strony utwórz menu zawierające trzy pozycje: *Programy*, *Sprzęt*, *Systemy*. Wybrana myszą pozycja menu powinna zostać podświetlona. Kliknięcie wybranej pozycji menu spowoduje wyświetlenie w określony przez Ciebie sposób opisu zaznaczonego hasła. Po wyborze innej pozycji menu treść hasła zostanie ukryta. W głównej części strony umieść formularz rejestracyjny dla osób odwiedzających stronę. Powinien on zawierać następujące pola: *Nazwisko*, *Imię*, *Adres*, *Wykształcenie*, *Wiek*, *Adres e-mail*. Opracuj walidację formularza, dzięki której sprawdzisz, czy użytkownik wypełnił wymagane pola oraz prawidłowo wpisał dane.



