

Moduł 1

Podstawy HTML

Spis treści

| | |
|----------------------------------|----|
| Podstawy HTML | 1 |
| Informacje o module..... | 2 |
| Przygotowanie teoretyczne..... | 3 |
| Przykładowy problem | 3 |
| Podstawy teoretyczne..... | 3 |
| Uwagi dla studenta | 18 |
| Dodatkowe źródła informacji..... | 18 |

Informacje o module

Opis modułu

W tym module znajdziesz informacje dotyczące języka HTML. Poznasz jego możliwości i ograniczenia. Zobaczysz, w jaki sposób używać tabel i formularzy do pobierania informacji od użytkownika. Nauczysz się, jak tworzyć proste strony i serwisy internetowe.

Cel modułu

Celem modułu jest przedstawienie możliwości wykorzystania języka HTML, formatowania dokumentu przy użyciu tabel oraz pobierania informacji od użytkownika.

Uzyskane kompetencje

Po zrealizowaniu modułu będziesz:

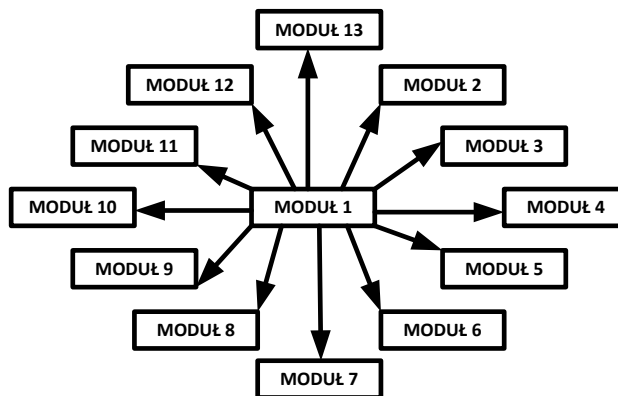
- potrafił tworzyć nowe strony internetowe w języku HTML
- wiedział jak tworzyć i kiedy warto stosować tabele
- potrafił tworzyć formularze i przysyłać je do odbiorcy

Wymagania wstępne

Ten moduł nie ma żadnego wymagania wstępnego. Możesz od razu rozpocząć pracę z tym modułem.

Mapa zależności modułu

Zgodnie z mapą zależności przedstawioną na Rys. 1, przed przystąpieniem do realizacji tego modułu nie jest wymagane zapoznanie się z materiałem zawartym w innych modułach.



Rys. 1 Mapa zależności modułu

Przygotowanie teoretyczne

Przykładowy problem

Znajomość języka HTML w projektowaniu stron internetowych w dobie narzędzi pracujących w trybie WYSIWYG (ang. *What You See Is What You Get*, co znaczy dosłownie „to co widzisz jest tym, co otrzymasz”) wydaje się niepotrzebna. Nie jest to prawdą. Projektant aplikacji internetowej musi w swojej pracy sięgać również na poziom znaczników HTML.

Pełna kontrola nad kodem tworzonym po stronie użytkownika wymaga jawnego odwołania się do znaczników HTML. Znaczniki HTML nie wymagają przetwarzania na serwerze. Znajomość znaczników pozwala nam również łatwiej zrozumieć problemy wyświetlania informacji.

Bez znajomości języka HTML bardzo trudno jest nam tworzyć dobre strony. Tabele w HTML są bardzo przydatne do przedstawiania danych tabelarycznych, choć niektórzy do dzisiaj budują w oparciu o nie wygląd aplikacji. Nie jest to błąd, ale powoduje ograniczenie dostępności strony dla urządzeń mobilnych, czy dla osób z ograniczonymi zdolnościami percepcji.

Formularze w języku HTML udostępniają podstawową funkcjonalność tworzenia elementów interakcji z użytkownikiem.

Podstawy teoretyczne

World Wide Web (w skrócie WWW lub Web) jest multimedialnym, hipertekstowym systemem informacyjnym opartym na publicznie dostępnych, otwartych standardach. Podstawowym zadaniem WWW jest publikowanie informacji.

WWW jest siecią milionów komputerów działających w oparciu o model klient-serwer. Serwer przechowuje informacje i obsługuje zapytania klienta. Klient tworzy zapytanie, wysyła, odbiera je i przetwarza, a następnie wyświetla informacje dla końcowego użytkownika. Klientem sieci Web jest najczęściej przeglądarka internetowa.

Cechy WWW:

- hipertekstowy system informacyjny
- graficzna reprezentacja informacji
- łatwa nawigacja
- system wieloplatformowy
- system rozproszony
- sieć dynamiczna
- sieć interaktywna
- uniwersalny sposób nazewnictwa zasobów informacyjnych (URI)
- ustalony protokół dostępu do zasobów (HTTP)
- stosowanie specjalnego typu języka w celu nawigacji po zasobach (HTML)

Strona internetowa

Strona internetowa to dokument tekstowy napisany w jednej z odmian języka znaczników, np. HTML, interpretowanego przez przeglądarkę użytkownika. Strony można podzielić ze względu na sposób jej tworzenia na statyczne i dynamiczne.

Strona statyczna musi być przygotowana wcześniej przez twórcę. Jej ostateczny wygląd oraz cała treść musi być znany na etapie projektowania.

Zalety:

- łatwość implementacji

Wady:

- brak elastyczności

Strona dynamiczna jest generowana przez serwer na podstawie działania użytkownika serwisu i danych zapisanych najczęściej w relacyjnej bazie danych.

Zalety:

- interaktywność
- łatwość aktualizacji

Wady:

- trudność w implementacji, testowaniu i wdrażaniu
- większe wymagania sprzętowe serwera i łączy internetowych

Serwis internetowy

Zbiór powiązanych ze sobą i umieszczonych w jednej lokalizacji stron internetowych nazywa się *witryną* lub *serwisem internetowym*. *Portal* jest zbiorem wielu informacji i zasobów z kilku lub jednej dziedziny. Portale zawierają najczęściej: wiadomości, informacje tematyczne, forum, Portale można podzielić na:

- **poziome** – zawierają informacje z wielu dziedzin, dodatkowo mogą udostępniać np. wyszukiwarki, pocztę elektroniczną
- **pionowe** – zawierają informacje z jednej dziedziny (nazywane wortalami)

Protokół HTTP

HTTP (ang. *HyperText Transfer Protocol*) – protokół warstwy aplikacji służący do transferu dokumentów HTML, obrazów, multimediiów itp. Jest protokołem bezstanowym – interakcja klienta (najczęściej przeglądarki internetowej) z serwerem WWW odbywa się w modelu bezpołączeniowym:

- Klient nawiązuje połączenie TCP/IP z serwerem, najczęściej na porcie 80.
- Klient wysyła żądanie do serwera wraz z informacjami na temat systemu operacyjnego, wersji języka, obsługiwanych protokołów informacyjnych. W sieci przesyłane są polecenia tekstowe zorganizowane w wiersze np.:

```
GET /KatalogNaSerwerze/plik.html HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language: pl
[dodatkowa pusta linia]
```

- Serwer wysyła do klienta kod odpowiedzi (informujący o sukcesie lub problemach z wysłaniem żądanego zasobu), informacje dotyczące rodzaju zawartości zasobu, kodowania, itp. oraz żądany zasób:

```
HTTP/1.1 200 OK.
Server: ASP.NET Development Server/8.0.0.0
Date: Sat, 29 Jul 2006 22:49:42 GMT
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 487
Connection: Close
```

- Serwer po przesłaniu odpowiedzi automatycznie zamyka połączenie z klientem.

Najważniejsze metody klienta

Jeśli użytkownik musi podać jakieś dane, to jedyną możliwością jest pobranie ich ze standardowego formularza HTML. Dane podane mogą być przesłane do serwera za pomocą ośmiu metod, z których najczęściej wykorzystywane to:

- **GET** – pobieranie zasobu wskazanego przez URL. Może zawierać parametry dołączane do URL (po znaku zapytania).
- **POST** – wysyła dane użytkownika do określonego zasobu:
 - dane zawarte są w ciele komunikatu
 - nagłówek Content-Type i Content-Length
 - w odpowiedzi klient otrzymuje dokument wygenerowany dynamicznie.

Przekazywanie parametrów

Parametry można przysyłać przy pomocy metody POST lub GET.

W metodzie GET Parametry są przekazywane w adresie URL po znaku zapytania, oddzielone od siebie znakiem **&**, w formacie parametr=wartość, np.

```
GET /index.aspx?cat=2&prodid=23
```

Znaki zastrzeżone (=, &, &, +) są kodowane do postaci %xx, gdzie xx jest kodem ASCII znaku zapisanym szesnastkowo, np. znak spacji zamieniany jest na %20.

W metodzie POST informacje są pobierane z formularza znajdującego się na stronie internetowej.

Język HTML

HTML (ang. HyperText Markup Language) jest językiem znaczników zaprojektowanym do tworzenia stron internetowych. Służy on do opisywania struktury dokumentu (nagłówki, akapity, listy), a nie jego formatowania. Umieszczanie informacji o formatowaniu w dokumencie jest złą praktyką. Interpretacją znaczników i graficznym przedstawieniem zawartości zajmują się przeglądarki internetowe. Gramatyka języka jest oparta o standard SGML.

SGML (ang. *Standard Generalized Markup Language*) to ustandaryzowany nadrzędny język znaczników służący do ujednoczenia struktury i formatu różnego typu informacji. Umożliwia zapisywanie informacji w formie dokumentu tekstowego, dzięki czemu możliwe jest ich łatwe przenoszenie, wyświetlanie i drukowanie.

Podstawowe właściwości:

- Znacznik jest reprezentowany przez tekst zawarty w nawiasach trójkątnych (< i >).
- Tekst pomiędzy nawiasami to nazwa znacznika.
- Istnieją trzy rodzaje znaczników:
 - otwierający postaci <znacznik>
 - zamykający postaci </znacznik>
 - samodzielny postaci <znacznik />
- Znacznik otwierający może posiadać szereg atrybutów definiowanych jako nazwa="wartość".
- Znaki białe, np. spacje, tabulatory lub znaki nowej linii są ignorowane.

Dokument HTML

Dokument HTML jest plikiem tekstowym, w którym znajdują się polecenia (znaczniki) HTML. Z tego wynika, że dokument taki można utworzyć za pomocą dowolnego edytora tekstowego, ręcznie dodając znaczniki. Metoda taka jest skuteczna, ale zbyt uciążliwa i podatna na błędy, szczególnie przy projektowaniu dużych serwisów internetowych. Dlatego na rynku pojawiło się wiele

specjalizowanych edytorów, które wydatnie ułatwiają konstruowanie dokumentu, wspomagając wprowadzanie poleceń. Narzędzia można podzielić na dwie kategorie: narzędzia graficzne WYSIWYG oraz tekstowe. Narzędzia graficzne umożliwiają tworzenie serwisów osobom, które nie znają języka HTML, jednak znajomość tego języka ułatwia pracę twórcom stron.

Dokument rozpoczyna się definicją typu dokumentu. W języku HTML w wersji 4.01 element ten jest wymagany, podobnie jak w języku XHTML. Cały dokument zawarty jest w znaczniku `<html>` i składa się z dwóch głównych części: nagłówka (`<head>`) oraz ciała – treści dokumentu `<body>`.

Wersja minimalna dokumentu:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <!-- Naglowek dokumentu -->
    <title> Tytuł strony </title>
  </head>
  <body>
    <!-- Treść dokumentu -->
  </body>
</html>
```

Definicja typu dokumentu

Definicja typu dokumentu określa wersję języka, która jest wykorzystywana w dokumencie. Poprawny dokument HTML w wersji 4.01 zawiera w pierwszym wierszu definicję typu dokumentu. Podobnie jest w XHTML. Deklaracja typu zawiera informacje o dokumencie definicji typu (ang. *Document Type Definition*, DTD) wykorzystywanym w dokumencie. DTD zawiera definicje elementów, które można użyć w danym dokumencie.

W języku HTML w wersji 4.01 określono trzy dokumenty definicji typu, które można użyć w projektowanej stronie internetowej:

- **Ścisły** (ang. *strict*) – zawiera wszystkie elementy i atrybuty, które są zdefiniowane w standardzie bez elementów nie zalecanych oraz informacji dotyczących dokumentów zawierających ramki. Deklaracja typu dokumentu:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- **Przejściowy** (ang. *transitional*) – zawiera wszystkie elementy typu bezpośredniego oraz elementy i atrybuty niezalecane (większość dotyczy wizualnej formatowania). Deklaracja typu dokumentu:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

- **Zbiór ramek** (ang. *frameset*) – zawiera wszystkie elementy typu przejściowego oraz elementy zbioru ramek. Deklaracja typu dokumentu:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

W dokumencie XHTML przed elementem korzenia (najczęściej znacznikiem `<html>`) musi znajdować się deklaracja typu dokumentu, tzw. *DOCTYPE*. Publiczny identyfikator dołączony do tej deklaracji musi odnosić się do jednego z trzech DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Nagłówek strony

Nagłówek strony zawiera informacje, które nie są wyświetlane na stronie, ale mają wpływ na jej wygląd. Przykłady tu prezentowane zapisane zostały zgodnie z językiem HTML. Różnica pomiędzy zapisem HTML a XHTML to przede wszystkim konieczność stosowania w XHTML bardziej restrykcyjnych reguł formatowania dokumentu, np. zamknięcie znacznika czy pisanie nazw znaczników małymi literami, lub otaczanie wartości atrybutów cudzysłowem, podczas gdy w HTML można użyć apostrofów, w XHTML wymagana jest również odpowiednia kolejność zamykania znaczników.

Przykładowe znaczniki w sekcji <head>:

- <title> – zawiera tytuł dokumentu wyświetlany w pasku tytułowym okna przeglądarki. Zgodnie ze specyfikacją języka, każdy dokument musi posiadać ten znacznik w sekcji <head>.
- <meta> – umożliwia autorom strony na podawanie informacji o stronie. Informacje mogą być zawarte w następujących atrybutach:
 - lang – określa język zawartości
 - content – ustala wartość atrybutu
 - name – ustala nazwę i rodzaj informacji
 - scheme – określa schemat interpretacji informacji
 - http-equiv – ustala nazwę i rodzaj informacji
- <link> – określa relację pomiędzy połączonymi dokumentami. Znacznik wykorzystywany do określania lokalizacji zewnętrznego dokumentu CSS. Atrybuty znacznika:
 - charset – definiuje kodowanie znaków we wskazanym zasobie
 - href – adres zasobu
 - type – określa typ MIME wskazanego zasobu
 - href – adres zasobu
 - rel – definiuje związek pomiędzy aktualnym, a wskazywanym zasobem.Rodzaje związków:
 - stylesheet – zewnętrzny arkusz styli
 - alternate – alternatywna wersja dokumentu, często stosowana z atrybutem Lang
 - glossary – słownik
 - copyright – prawa autorskie dokumentu
 - help – plik z dodatkową pomocą, linkami itp.
 - rev – definiuje związek pomiędzy wskazywanym, a aktualnym zasobem

Formatowanie treści dokumentu

W znaczniku <body> umieszczona jest treść dokumentu. W tym miejscu możliwe jest korzystanie z pełnej palety możliwości języka HTML, w szczególności:

- Definiowanie struktury tekstu:
 - nagłówki, akapity itp.
 - listy wypunktowane i numerowane
 - wyróżnianie cytatów, itp.
- Formatowanie tekstu:
 - wielkość i kolor tekstu
 - podkreślanie, pogrubianie, pochylanie, wyrównanie tekstu do lewej, prawej, środka, lub do obydwu marginesów, itp.

- Tabele
- Osadzanie grafiki i innych obiektów
- Odnośniki
- Formularze

Definiowanie struktury tekstu

Definiowanie struktury dokumentu polega na określaniu podziału tekstu na rozdziały, akapity i inne jednostki logiczne dokumentu.

Nagłówki

Znacznik `<h1>` do `<h6>` określa nagłówek – tytuł rozdziału. `<h1>` definiuje nagłówek najwyższego poziomu (dotyczy głównego rozdziału i jest wyświetlany największymi literami), a `<h6>` najniższego poziomu (jest wyświetlany najmniejszą czcionką). Dobrą praktyką jest umieszczanie co najwyżej jednego nagłówka pierwszego poziomu `<h1>`.

`<h1>`To jest nagłówek 1`</h1>`

To jest nagłówek 1

`<h2>`To jest nagłówek 2`</h2>`

To jest nagłówek 2

`<h3>`To jest nagłówek 3`</h3>`

To jest nagłówek 3

`<h4>`To jest nagłówek 4`</h4>`

To jest nagłówek 4

`<h5>`To jest nagłówek 5`</h5>`

To jest nagłówek 5

`<h6>`To jest nagłówek 6`</h6>`

To jest nagłówek: 6

Akapity i wiersze

Akapity są podstawową jednostką logiczną podziału dokumentu. HTML nie rozpoznaje białych znaków (spacje, znaki końca linii), traktując je, niezależnie od ich ilości, jak pojedynczą przerwę – odstęp. Zawartość akapitu należy umieścić w specjalnym znaczniku `<p>`. Znacznik `<p>` musi zostać zamknięty (`</p>`).

Przeglądarka sama decyduje, w którym miejscu ma zostać przełamana linia. Użytkownik może wymusić przełamanie za pomocą znacznika `
`.

Listy wypunktowane i numerowane

W wielu dokumentach istnieje potrzeba zaprezentowania fragmentu tekstu za pomocą listy wypunktowanej lub numerowanej. Możliwe jest również tworzenie wielopoziomowych list.

Znacznik `` umożliwia utworzenie szkieletu wykazu wypunktowanego. Poszczególne punkty wykazu należy umieszczać między znacznikami `` i ``. Standardowo pozycje listy oznaczone są kołem. Do utworzenia listy numerowanej zamiast znacznika `` wykorzystuje się znacznik ``. Standardowo pozycje listy są numerowane za pomocą liczb arabskich.

Zmiana wyglądu oraz sposobu znakowania listy jest możliwa przy pomocy kaskadowych arkuszy stylów, które poznasz w kolejnym module.

Inne wyróżnienia tekstu

W języku HTML możliwe jest logiczne wyróżnianie pewnych informacji, np. cytatów. Wyróżnienie nie tylko dodaje informacje o strukturze, ale również jest związane z formatowaniem danego fragmentu przez przeglądarkę. Przeglądarki albo zmieniają wygląd zawartości tekstu, albo ignorują dany znacznik nie zmieniając formatowania.

Lista znaczników:

- `` – wyróżnienie
- `` – mocniejsze wyróżnienie

- `<cite>` – zawiera cytowanie lub odnośnik do innego źródła
- `<dfn>` – zawiera definicję
- `<code>` – fragment kodu programu komputerowego
- `<samp>` – zawiera przykładowy wynik działania programu, skryptu, itp.
- `<kbd>` – określa tekst, który ma wprowadzić użytkownik
- `<var>` – zawiera zmienną lub argument programu
- `<acronym>` – zawiera akronim, np. WWW, HTTP, URI, itp.

Osadzanie grafiki i innych obiektów

Grafika oraz inne multimedialne elementy są dzisiaj jedną z najbardziej charakterystycznych wyróżników stron internetowych.

Osadzanie grafiki na stronie

Podstawowa konstrukcja ma następującą postać:

```

```

Atrybut `alt` służy do wyświetlania tekstu, gdy nie jest możliwe wyświetlenie grafiki.

Atrybuty wymagane:

- `src` – lokalizacja pliku (URL)
- `alt` – krótki opis wyświetlany, gdy nie można wyświetlić grafiki

Pozostałe atrybuty opcjonalne są w większości przestarzałe i zamiast nich lepiej jest używać stylu CSS.

Osadzanie multimediiów na stronie

Do osadzania grafiki służy znacznik `<object>`. Umożliwia on określenie danych i parametrów obiektu osadzonego w dokumencie HTML wraz z kodem, który może być użyty do wyświetlenia lub manipulacji danymi.

Atrybuty znacznika:

- `align` – wyrównanie obiektu (`top`, `bottom`, `left`, `right`)
- `archive` – lista ładowanych wstępnie zasobów
- `border` – rozmiar ramki obiektu
- `classid` – klasa obiektu
- `codebase` – określa adres bazowy dla `classid`, `data` i `archive`
- `codetype` – określa typ obiektu
- `data` – wskazuje adres danych obiektu
- `declare` – deklaracja obiektu
- `height` – określa wysokość obiektu
- `hspace` – pionowy odstęp od obiektu
- `name` – unikatowa nazwa elementu
- `standby` – informacja pokazywana w trakcie ładowania obiektu
- `type` – określa typ danych
- `usemap` – podaje mapę elementu
- `vspace` – odstęp poziomy od obiektu
- `width` – określa szerokość obiektu

Grupowanie liniowe i blokowe

Znacznik `` tworzy uniwersalny zakres liniowy i pozwala nadać mu wspólne atrybuty za pomocą stylu.

Znacznik `<div>` obejmuje blokiem grupę elementów w dokumencie i pozwala im nadać wspólne atrybuty. Element ten jest kontenerem umożliwiającym dodanie struktury do dokumentu HTML. Najczęściej służy on do logicznego podziału zawartości strony, np. nawigacji, nagłówka czy stopki strony.

Odnośniki

Odnośniki (odsyłacze, hipertącza, łącza hipertekstowe, odnośniki) są najbardziej charakterystycznym elementem sieci WWW. Dzięki nim można mówić o HTML jako języku hipertekstowym – odnośniki w jednym kontekście do innego kontekstu. Bez odsyłaczy nie byłoby możliwe tworzenie serwisów internetowych, bez nawigacji stałby się on tylko zbiorem niepowiązanych dokumentów lub musiałby zostać przedstawiony na jednej stronie, co jest najczęściej możliwe, chociaż w przypadku dużych portali mało prawdopodobne.

Odnośnik to wskazanie do innego miejsca. Kliknięcie na nim przenosi użytkownika do określonego miejsca. Może to być miejsce na tej samej stronie, w tym samym serwisie, ale równie dobrze może to być strona na drugim końcu świata.

Do utworzenia odnośnika w języku HTML potrzebne są lokalizacja zasobu oraz tekst opisujący zasób. Format najprostszego odnośnika:

```
<a href="lokalizacja_zasobu">Tekst opisujący zasób</a>
```

Parametr `href` (ang. *Hypertext Reference*) – odnośnik hipertekstowy. Służy do określenia URL zasobu, wskazywanego przez dane połączenie.

Lokalizacja to najczęściej adres URL do zasobu lub nazwa pliku. Lokalizacja może zawierać ścieżkę względną lub bezwzględną.

Ścieżka względna jest podawana względem aktualnego pliku, np.:

```
<a href="../plik_w_katalogu_nadrzednym">Tekst opisujący zasób</a>  
<a href="/kat/plik_w_katalogu_podrzednym">Tekst opisujący zasób</a>
```

Ścieżka bezwzględna jest podawana jako pełny adres internetowy strony.

```
<a href="http://www.example.com/plik">Tekst opisujący zasób</a>
```

Połączenie do określonych miejsc w dokumencie

Aby odwołać się do określonego miejsca w dokumencie należy określić zakładkę (kotwicę, etykietę), a następnie wskazać ją w odnośniku.

Aby utworzyć zakładkę należy użyć konstrukcji:

```
<a name="nazwa_zakladki"> </a>
```

Możliwe jest utworzenie dowolnej ilości etykiet na stronie. Jest to szczególnie wygodne w przypadku stron o dużej objętości.

Odnośnik do etykiety ma postać:

```
<a href="strona.html#nazwa_zakladki">Odnośnik do zakładki</a>
```

lub jeśli odnośnik jest na tej samej stronie co zakładka, to można odnieść się w skróconej formie:

```
<a href="#nazwa_zakladki">Odnośnik do zakładki</a>
```

Odnośniki do adresu e-mail

Odnośnik do adresu e-mail jest taki sam, jak standardowy odnośnik. Różnica polega na tym, że po kliknięciu go nie zostanie otworzona nowa strona, lecz aplikacja do wysyłania listów elektronicznych. Linki do adresów e-mail są szczególnie pomocne dla kontaktu z autorem strony czy biurem obsługi klienta na stronie firmowej.

Odnośnik do adresu e-mail ma postać:

```
<a href="mailto:adres@serwer">Wyślij do nas list</a>
```

Tabele

Tabele w języku HTML są podobne do tabel spotykanych w innych dziedzinach życia, umożliwiając przedstawienie danych w uporządkowanej postaci. Dodatkowo umożliwiają układanie danych – tekstu, obrazków, odnośników, formularzy, pół formularza, innych tabel, itp. w wiersze i kolumny komórek. Należy jednak dodać, że współcześnie panuje tendencja odchodzenia od wykorzystywania tabel w charakterze szkieletu strony.

Tabela może zawierać nagłówek zawierający krótki opis zawartości tabeli. Tabel składa się z wierszy, które zawierają komórki. Istnieje możliwość scalania wielu wierszy lub kolumn.

Znacznik <table>

Tabele rozpoczyna znacznik <table>, a kończy </table>. Znacznik <table> posiada kilka atrybutów. Atrybut width określa szerokość tabeli. Może być ona podana m.in. w pikselach lub jako procent całej szerokości dostępnej dla okna przeglądarki. Przykłady:

```
<table width="150"> <table width="50%">
```

Innym atrybutem jest border, który określa grubość ramki. Parametr ten określa w pikselach grubość ramki. Wartość 0 oznacza całkowite wyłączenie ramki.

Dodatkowo możliwe jest określenie odległości pomiędzy komórkami – atrybut cellspacing. Odległość od komórki do jej zawartości określa atrybut cellpadding. Obie wartości określane są w pikselach.

```
<table border="2" cellspacing="3" cellpadding="3">
```

Wiersze tabeli

Podstawową jednostką tabeli są wiersze. Każda tabela musi zawierać co najmniej jeden wiersz. Wiersz tabeli reprezentowany jest przez znacznik <tr>.

```
<table border="2" cellspacing="3" cellpadding="3">
  <tr> </tr>
</table>
```

Wiersz tabeli tworzy ramy dla komórek z danymi. Możliwe jest dodanie wielu wierszy do tabeli:

```
<table border="2" cellspacing="3" cellpadding="3">
  <tr> </tr>
  <tr> </tr>
</table>
```

Komórka tabeli

Komórka tabeli dodaje pionową kolumnę do tabeli. Są one reprezentowane przez znacznik <td>. Wspólnie ze znacznikami <table> oraz <tr>, znacznik <td> stanowi najważniejsze elementy budowy tabeli:

```
<table width="80%" border="2" cellspacing="3" cellpadding="3">
<tr>
  <td></td>
  <td></td>
```

```
</tr>  
</table>
```

Powyższy kod HTML tworzy tabelę przedstawioną poniżej.

Nagłówki tabeli

Nagłówek tabeli określa nagłówek kolumny lub wiersza. Tekst w komórce, która jest nagłówkiem jest pogubiony. Nagłówek tabeli reprezentowany jest przez znacznik <th>.

W przypadku nagłówka wiersza:

```
<table width="80%" border="2" cellspacing="3" cellpadding="3">  
  <tr>  
    <th>Lp.</th>  
    <th>Imię</th>  
  </tr>  
  <tr>  
    <td>1</td>  
    <td>Piotr</td>  
  </tr>  
</table>
```

W przypadku nagłówka kolumny:

```
<table width="80%" border="2" cellspacing="3" cellpadding="3">  
  <tr>  
    <th>Lp.</th>  
    <td>1</td>  
  </tr>  
  <tr>  
    <th>Imię</th>  
    <td>Piotr</td>  
  </tr>  
</table>
```

Tytuł tabeli

Często niezbędne jest dołączenie do tabeli jej tytułu. Tytuł powinien zawierać krótki opis wyjaśniający jej treść. Tytuł reprezentowany jest przez znacznik <caption>, który należy umieścić zaraz za znacznikiem <table>, a przed znacznikami <tr>.

```
<table width="80%" border="2" cellspacing="3" cellpadding="3">  
  <caption>Tytuł</caption>  
  <tr>  
    <th>Lp.</th><td>1</td>  
  </tr>  
  <tr>  
    <th>Imię</th><td>Piotr</td>  
  </tr>  
</table>
```

Scalanie wierszy

Czasem istnieje konieczność scalenia kilku wierszy w jednej komórce. W tym celu należy użyć atrybutu rowspan, podając ile wierszy ma zostać połączonych.

Przykład:

```
<table border="1" cellpadding="2" cellspacing="2">  
  <tr>  
    <th rowspan="2" >Nagłówek (scalone 2 wiersze)</th>  
    <td>dane</td>  
    <td rowspan="3" > dane (scalone 3 wiersze)</td>
```

```
    <td> dane</td>
</tr>
<tr>
    <td>dane</td>
    <td>dane</td>
</tr>
<tr>
    <th>Nagłówek</th>
    <td>dane</td>
    <td>dane</td>
</tr>
</table>
```

Scalanie kolumn

Podobnie jak wiersze, można również scalić kolumny. Służy do tego atrybut `colspan`.

```
<table border="2" cellpadding="2" cellspacing="2">
  <tr>
    <th>Nagłówek</th>
    <th>Nagłówek</th>
    <th colspan="2">Nagłówek (scalne 2 kolumny)</th>
  </tr>
  <tr>
    <td>dane</td>
    <td colspan="2"> dane (scalone 2 kolumny)</td>
    <td>dane</td>
  </tr>
  <tr>
    <td colspan="2">dane (scalone 2 kolumny)</td>
    <td>dane</td>
    <td>dane</td>
  </tr>
</table>
```

Możliwe jest również jednoczesne scalanie wierszy i kolumn:

```
<table border="2" cellpadding="2" cellspacing="2">
  <tr>
    <th>Nagłówek</th>
    <th>Nagłówek</th>
    <th>Nagłówek</th>
    <th>Nagłówek</th>
    <th>Nagłówek</th>
    <th>Nagłówek</th>
  </tr>
  <tr>
    <td>dane</td>
    <td colspan="2">dane (scalone 2 kolumny)</td>
    <td rowspan="2" colspan="2">dane (scalone 2 wiersze, 2 kolumny)</td>
    <td>dane</td>
  </tr>
  <tr>
    <td>dane</td>
    <td>dane</td>
    <td>dane</td>
    <td>dane</td>
  </tr>
</table>
```

| Nagłówek | Nagłówek | Nagłówek | Nagłówek | Nagłówek | Nagłówek |
|----------|-----------------------------|----------|---|----------|----------|
| dane | dane (scalone 2 kolumny) | | dane (scalone 2 wiersze, 2 kolumny) | | dane |
| dane | dane | dane | | | dane |

Rys. 2 Przykład tabeli

Formularze

Formularz umożliwia pobranie od użytkownika pewnych informacji. Zawiera przycisk umożliwiający wysłanie danych do określonego pliku/aplikacji na serwerze. Przykłady wykorzystania formularza to:

- podanie danych osobowych i adresu email w celu zapisania się na forum
- rezerwacja miejsca wakacji w biurze podróży
- podawanie danych do przelewu w bankowości elektronicznej
- zakup określonych towarów i usług w sklepie internetowym

Formularze są relatywnie proste do tworzenia, przynajmniej z pozycji kodu HTML. Największy problem stanowi przetworzenie wpisanych danych po stronie serwera tak, aby odpowiadały one zamierzeniom projektanta i użytkownika.

Formularz zawarty jest w znaczniku <form>. Ma on dwa atrybuty:

- **method** – określa metodę, jaką zostaną wysłane dane do serwera. Są to metody protokołu http, takie jak GET i POST.
- **action** – określa ścieżkę do miejsca, w którym skrypt zostanie przetworzony.

Parametr **action** może wskazywać nie tylko ścieżkę na serwerze, ale również w prostych przypadkach umożliwia przesyłanie danych z formularza przy użyciu poczty elektronicznej:

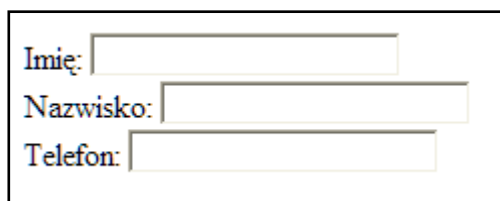
```
<form method="post" action="mailto:adres@serwer?subject=tytuł">
```

Pola tekstowe

Pola tekstowe są wykorzystywane w wielu przypadkach do pobrania od użytkownika określonych, krótkich fragmentów tekstu, takich jak imię, nazwisko i numeru telefonu. Pole tekstowe jest tworzone przy wykorzystaniu znacznika `input` z parametrem `type="text"`.

Przykład:

```
<form method="post" action="Default.aspx">  
  Imię: <input type="text" /><br />  
  Nazwisko: <input type="text" /><br />  
  Telefon: <input type="text" />  
</form>
```



Wygląd formularza z trzema polami tekstowymi. Pierwsze pole jest poprzedzone słowem 'Imię:', drugie 'Nazwisko:', a trzecie 'Telefon:'. Każde pole jest prostokątnym polem tekstowym z pionową linią kursora.

Rys. 3 Przykład formularz

Należy zauważyć, że nie ma możliwości rozróżnienia poszczególnych pól formularza. Do tego celu służą parametry `name` i `id` z odpowiednimi wartościami. Wykorzystanie obu parametrów zapewni

kompatybilność z wcześniejszymi aplikacjami oraz umożliwi identyfikowanie określonych przestrzeni formularza dla formatowania oraz oprogramowania.

```
<form method="post" action="Default.aspx">  
  Imię: <input id="imie" name="imie" type="text" /> <br />  
  Nazwisko: <input id="nazwisko" name="nazwisko" type="text" /> <br />  
  Telefon: <input id="telefon" name="telefon" type="text" />  
</form>
```

Kolejnym elementem związanym z polem tekstowym jest określanie rozmiaru pola za pomocą atrybutu `size` oraz określanie maksymalnej ilości znaków, jaką pole akceptuje za pomocą atrybutu `maxlength`.

Pole tekstowe umożliwia również ukrywanie wprowadzanych ważnych informacji, takich jak hasła, zamieniając atrybut `type="text"` na `type="password"`:

```
Hasło <input id="haslo" name="haslo" type="password" />
```

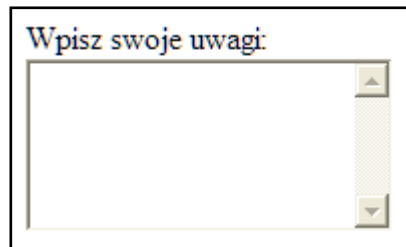


Rys. 4 Pole ukryte

Obszar tekstowy

Obszar tekstowy to miejsce, w którym użytkownik strony może wpisać swoje dodatkowe uwagi. Obszar ten umożliwia również wielowierszowe wprowadzanie danych. Jest on tworzony przy pomocy znacznika `<textarea>` wraz z parametrami `rows` i `cols`, określającymi odpowiednio ilość wierszy i kolumn przeznaczonych do wprowadzania tekstu.

```
<form method="post" action="Default.aspx">Wpisz swoje uwagi:<br />  
<textarea id="komentarz" cols="20" rows="20" > </textarea>  
</form>
```



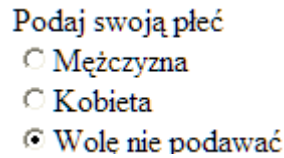
Rys. 5 Przykład obszaru tekstowego

Pole opcji i pole wyboru

Pole opcji (przełącznika) jest polem, w którym użytkownik wybiera jedną z dostępnych możliwości. W polu stosowany jest znacznik `<input>` oraz parametr `type="radio"`.

Do grupowania wartości poszczególnych opcji wykorzystywany jest parametr `name`, który musi być wspólny dla wszystkich elementów listy. Parametr `checked="checked"` umożliwia domyślne zaznaczenie danej opcji.

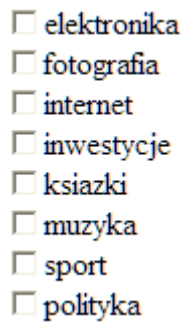
```
<form method="post" action="Default.aspx">  
  Podaj swoją płeć<br />  
  <input id="meczyczna" name="plec" type="radio" /> Mężczyzna<br />  
  <input id="kobieta" name="plec" type="radio" /> Kobieta<br />  
  <input id="brak" name="plec" type="radio" checked="checked" />  
  Wolę nie podawać<br />  
</form>
```



Rys. 6 Przykład pola opcji

Pole wyboru umożliwia dokonanie wyboru kilku opcji jednocześnie, np. zaznaczenie kilku zainteresowań, czy kilku sposobów odpoczynku. W polu stosowany jest znacznik `<input>` oraz parametr `type="checkbox"`.

```
<form method="post" action="Default.aspx">
  <input type="checkbox" name="zain" id="elektronika"/>elektronika<br />
  <input type="checkbox" name="zain" id="fotografia"/>fotografia<br />
  <input type="checkbox" name="zain" id="internet"/>internet<br />
  <input type="checkbox" name="zain" id="inwestycje"/>inwestycje<br />
  <input type="checkbox" name="zain" id="ksiazki"/>ksiazki<br />
  <input type="checkbox" name="zain" id="muzyka"/>muzyka<br />
  <input type="checkbox" name="zain" id="sport"/>sport<br />
  <input type="checkbox" name="zain" id="polityka"/>polityka
</form>
```



Rys. 7 Przykład pola wyboru

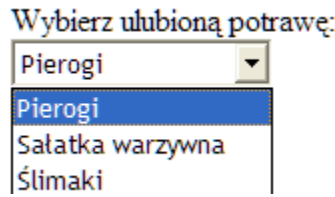
Lista rozwijana oraz wyboru

Lista rozwijana służy do tworzenia rozwijanych menu, zawierających kilka opcji. Użytkownik może wybrać jedną z nich. Składnia jest następująca:

```
<select name="nazwa_listy" size="liczba_wyświetlanych_pozycji" >
  <option> pierwsza pozycja</option>
  <option> druga pozycja</option>
  <option> trzecia pozycja</option>
  ...
</select>
```

Przeglądarka tworzy listę rozwijaną zawierającą poszczególne elementy zawarte w znaczniku `<option>`, parametr `value` umożliwia rozpoznanie wybranego elementu:

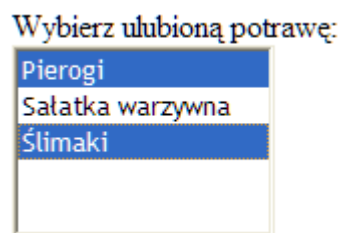
```
<form method="post" action="Default.aspx">
  Wybierz ulubioną potrawę:<br />
  <select id="potrawa">
    <option value="pierogi">Pierogi</option>
    <option value="slatka">Sałatka warzywna</option>
    <option value="slimaki">Ślimaki</option>
  </select>
</form>
```

Rys. 8 Przykład listy rozwijanej

Lista wyboru służy do tworzenia menu, zawierających kilka opcji. Możliwe jest wybranie jednej lub kilku z nich. Listę ta tworzy się podobnie do listy rozwijanej dodając atrybut `size="wyświetlana wielkość listy"`. wybór wielokrotny możliwy jest po dodaniu atrybutu `multiple="multiple"`.

```
<form method="post" action="Default.aspx">
  <select id="potrawa" size="5" multiple="multiple">
    <option value="pierogi">Pierogi</option>
    <option value="słatka">Sałatka warzywna</option>
    <option value="slimaki">Ślimaki</option>
  </select>
</form>
```



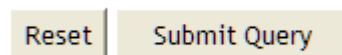
Rys. 9 Przykład listy wyboru

Wysyłanie informacji

Po wykonaniu formularza należy jeszcze zapewnić wysłanie jego zawartości do autora. Sposób wysłania jest określony w atrybutach `method` i `action` znacznika `<form>`. Do wysłania służy znacznik `input` z atrybutem `type="submit"`. Po kliknięciu tego przycisku przeglądarka wyśle dane (wypełnione pola tekstowe, wybrane opcje itp.) do miejsca zdefiniowanego w znaczniku `form`.

Często istnieje potrzeba zresetowania całego formularza, służy do tego znacznik `input` z atrybutem `type="reset"`.

```
<form method="post" action="Default.aspx">
  <input type="reset" />
  <input type="submit" />
</form>
```



Rys. 10 Przyciski formularza - Reset i Submit

Parametr `value` w obu przypadkach umożliwia dopasowanie wyświetlanego komunikatu na przyciskach:

```
<form method="post" action="Default.aspx">
  <input type="reset" value="Skasuj dane"/>
  <input type="submit" value="Wyślij"/>
</form>
```

Podsumowanie

W tym rozdziale przedstawione zostały podstawy języka HTML. Pokazaliśmy najważniejsze znaczniki, które są wykorzystywane do tworzenia stron internetowych.

Uwagi dla studenta

Jesteś przygotowany do realizacji laboratorium jeśli:

- znasz podstawowe znaczniki HTML
- wiesz jak tworzyć tabelę i jak łączyć wiersze i kolumny
- wiesz jak tworzyć i przysyłać formularze internetowe napisane w języku HTML

Pamiętaj o zapoznaniu się z uwagami i poradami zawartymi w tym module. Upewnij się, że rozumiesz omawiane w nich zagadnienia. Jeśli masz trudności ze zrozumieniem tematu zawartego w uwagach, przeczytaj ponownie informacje z tego rozdziału i zajrzyj do notatek z wykładów.

Dodatkowe źródła informacji

1. Bryan Pfaffenberger, Steven M. Schafer, Chuck White, Bill Karow, *HTML, XHTML i CSS. Biblia*, Helion, 2005

Książka zawiera opis najnowszych standardów związanych z tworzeniem stron WWW – obowiązującej obecnie specyfikacji HTML, CSS oraz XHTML. Przedstawia nowoczesne metodologie tworzenia i testowania witryn WWW oraz ich aktualizowania i modyfikowania.

2. Wempen Faithe, *HTML i XHTML Krok po kroku*, RM, 2007

Z książki nauczysz się nadawać strukturę stronom WWW, stosując tabele, ramki lub makiety oparte na podziałach. Utworzysz listy wypunktowane i numerowane, aby poprawić przekaz informacji. Będziesz wiedział, jak dodać hiperłącza i paski menu, aby ułatwić nawigację. Dowiesz się, w jaki sposób umieścić obrazki, klipy dźwiękowe i wideo na stronach WWW. Wprowadzisz kolory, rozmiary czcionek i różne formatowanie, stosując znaczniki i kaskadowe arkusze stylów. Zbudujesz formularze z przyciskami opcji, polami wyboru i rozwijanymi menu.

3. Jennifer Niederst Robbins, *HTML i XHTML. Leksykon kieszonkowy*, Helion, 2006

W książce znajdziesz zestawienie znaczników HTML – wraz z atrybutami i ich dopuszczalnymi wartościami – które jest niezwykle przydatne każdemu projektantowi witryn WWW. Autor opisuje najnowsze specyfikacje HTML i XHTML, wskazuje podobieństwa i różnice pomiędzy nimi, a także zawiera praktyczne informacje, dzięki którym Twoja witryna WWW zostanie zaprojektowana zgodnie z rekomendacjami W3C.