

Instrukcja 2

1. Elementy odnośników

W tej kategorii znajdują się trzy znaczniki: a, link oraz nav

a) <a> - znacznik definiuje swoją zawartość jako odnośnik do innej części otwartego dokumentu i/lub zupełnie nowego pliku/zewnętrznego źródła (hipertekst). W celu prawidłowego działania musi posiadać poprawnie zdefiniowany parametr href, który staje się miejscem docelowym tworzonego odnośnika. Dodatkowym parametrem może być target – dzięki niemu możemy określić gdzie nasza strona ma się otworzyć.

```
<a href="http://www.example.com">Odnośnik do zewnętrznego źródła</a>
```

```
<a href="#">Odnośnik do samego siebie</a>
```

```
<a href="#" target="_blank">Odnośnik do samego siebie (otwarcie w nowym oknie/zakładce)</a>
```

```
<a href="#wazne">Odnośnik do elementu z wartością atrybutu id = „wazne”</a>
```

```
<p>Standardowy paragraf</p>
```

```
<p>Standardowy paragraf</p>
```

```
<p>Standardowy paragraf</p>
```

```
<p>Standardowy paragraf</p>
```

```
<p>Standardowy paragraf</p>
```

```
<p id="wazne">Ważny paragraf, do którego zrobiliśmy odnośnik</p>
```

```
<p>Standardowy paragraf</p>
```

```
<p>Standardowy paragraf</p>
```

```
<p>Standardowy paragraf</p>
```

```
<p>Standardowy paragraf</p>
```

Aby lepiej sprawdzić działanie odnośnika 'wazne', dobrym rozwiązaniem jest dodać tyle elementów przed nim, by on sam znajdował się poza stroną (tak by w celu zobaczenia go trzeba było przesunąć zawartość strony).

Pozostałe wybrane atrybuty znacznika a:

- download – określa, że docelowa treść z atrybutu href ma zostać zapisana na dysku użytkownika (a nie wyświetlona w przeglądarce); jako wartość określa się nazwę pliku zapisu (zapisywać można pliki różnego typu; w momencie zapisu na dysk klienta rozszerzenie zostanie automatycznie rozpoznane i dodane). Na razie obsługiwane tylko przez przeglądarki Firefox i Chrome/Chromium.

```
<a href="http://www.w3schools.com/images/myw3schoolsimage.jpg"
```

```
download="w3image">Pobierz obrazek!</a>
```

- target – atrybut może przyjąć (poza poznaną we wcześniejszym przykładzie) następujące wartości: „_self” (domyślna akcja; otwiera odnośnik w tej samej aktywnej ramce), „_parent” (otwiera dokument w ramce-rodzicu), „_top” (otwiera odnośnik na całej dostępnej przestrzeni okna) oraz „<iframe>” (otwarcie w ramce o podanej nazwie).

- type – atrybut typu informacyjnego; określa się w nim typ treści zapisanej w odnośniku (np. dla stron WWW będzie to text/html, dla obrazów jpg image/jpeg, dla obrazów png image/png itd.; więcej typów można znaleźć pod adresem <http://www.iana.org/assignments/media-types>)

```
<a href="http://www.example.com" type="text/html">Przykładowa strona</a>
```

b) <nav> - znacznik ten grupuje całe zbiory odnośników (elementów <a>) w jeden kontener

(opakowanie/zestaw). Jego przydatność widoczna jest szczególnie w przypadku stron mobilnych – przeglądarki te mogą schować cały blok <nav> tworząc większy obszar dla samej treści strony, podczas gdy użytkownik przegląda zawartość serwisu (pojawia się przy przewijaniu strony).

```
<nav>
<a href="http://www.example.com">Examples</a> -
<a href="http://www.w3c.org">Strona organizacji WWW</a> -
<a href="http://w3school.com">Strona dla początkujących projektantów WWW</a>
</nav>
```

c) <link> - specjalny tag, podawany przeważnie w nagłówku strony (<head></head>), dzięki któremu możemy do naszej strony zaimportować pewne treści z zewnętrznego źródła (np. z innej strony bądź dokumentu). Przeważnie wykorzystuje się go do importu plików ze stylami CSS (inne typy plików mogą nie być wspierane przez poszczególne przeglądarki). Możliwe atrybuty:

- rel – relacja (powiązanie) pomiędzy aktualnie przetwarzanym dokumentem a dokumentem zewnętrznym (linkowanym do aktualnego dokumentu); możliwe wartości:

- 1) stylesheet (dokument CSS),
- 2) icon - ikona reprezentująca dokument
- 3) search - narzędzie przeszukujące dokument
- 4) prev/next - wskazuje, że aktualny dokument jest częścią pewnej serii i wskazuje poprzednią/następną część
- 5) prefetch - wskazuje gdzie docelowe źródło powinno być przechowywane
- 6) license - odnośnik do informacji prawnych o dokumencie
- 7) help wskazuje plik pomocy dla dokumentu
- 8) author - odnośnik do autora dokumentu
- 9) alternate - alternatywna wersja strony, np. wersja do druku, tłumaczenie na inny język lub kopia zapasowa

- type – określa typ treści zewnętrznego źródła (typy można znaleźć pod adresem <http://www.iana.org/assignments/media-types>)

- href – odnośnik do podłączanego dokumentu (pełny adres w przypadku dokumentów znajdujących się na zewnętrznych serwerach)

- media – określa zachowanie linkowanego dokumentu względem urządzenia, na którym jest wyświetlana docelowa strona; szczegółowe informacje można znaleźć pod adresem http://www.w3schools.com/tags/att_link_media.asp

```
<link rel="stylesheet" type="text/css" href="plik.css"/>
<link rel="icon" type="image/x-icon" href="ikona.ico"/>
<link rel="stylesheet" type="text/css" href="akuszdruku.css" media="print"/>
```

2. Elementy blokowe

Tym mianem określa się obiekty-znaczniki w ciele strony WWW, które stanowią niejako autonomiczną część naszego serwisu. Dzięki nim nasza strona może nabrać cech „dynamizmu” - jeżeli wykorzystamy odpowiednie skrypty JS oraz CSS możemy np. ukrywać bądź pokazywać wskazane bloki strony, przemieszczać je czy nawet usuwać. Pierwszym tego typu elementem (tagiem) był div, popularnie nazywany warstwą. Wprowadzenie go miało na celu podanie projektantom wygodnego sposobu podziału strony na dowolne elementy. Jednocześnie miało ich odczytać stosowania tzw. układu tabelarycznego (wykorzystywali element Table, który NIE JEST przystosowany do podziału treści na stronie; przykładowo zmiana rozdzielczości powoduje przeważnie zniszczenie całego układu) oraz ramkowego (element frame NIGDY NIE NALEŻAŁ DO STANDARDU HTML!).

W HTML5 do elementów blokowych dołączyły kolejne tagi (wymienione na wykładzie). Wiele

kontrowersji wzbudziły pogłoski o wyrzuceniu znacznika div, które oczywiście okazały się nieprawdziwe. Element div nadal jest (i będzie) dostępny w HTML5, jest on jednak elementem nie mającym określonej definicji. Jest po prostu abstrakcyjnym zbiorem czegoś bliżej nieokreślonego. Stąd wskazanie ze strony W3C by nie używać go, jeżeli można zamiast niego wykorzystać inny znacznik, bardziej pasujący do naszego zastosowania (ze wskazaniem na section oraz article). Poniżej wymienione zostaną wszystkie elementy blokowe dostępne w HTML5:

- <article>
- <section>
-

- <div> - element abstrakcyjny bez specjalnego przeznaczenia. Może mieć własny, określony rozmiar oraz położenie w dokumencie (bezwzględne, względne lub stałe). Dzięki elastyczności położenia oraz możliwości dynamicznej zmiany wielkości (np. poprzez JavaScript) dokumenty WWW powinny prezentować się tak samo w dowolnej przeglądarce, niezależnie od ustawionej rozdzielczości ekranu. Następny element po div zawsze będzie przeniesiony do nowej linii (jest on tzw. elementem blokowym).

```
<div>
<p>To jest jedna warstwa</p>
<div>
<p>Warstwy mogą się zagnieżdżać; mogą też posiadać więcej niż jeden element</p>
<p> tak jak ta warstwa</p>
</div>
<div>
<p>Ta warstwa pojawi się pod wcześniejszą; aby to zmienić (np. aby przylegała do poprzedniej)
należy użyć odpowiednich stylów CSS.</p></div>
</div>
```

Jeżeli chcemy natomiast wydzielić część strony by zawierała pewną treść „odseparowaną” od pozostałych elementów powinniśmy używać tagu <article>:

Przykład:

```
<article>
<h1>Tytuł wpisu</h1>
<p>Nowy tekst części tworzonej strony; może to być dowolna treść (np. wpis na blogu, wiadomość
na forum czy też po prostu jakakolwiek informacja, która nie powinna znajdować się jako taka w
sekcji body</p>
<p>Oczywiście możemy w article wrzucać dowolną ilość nowych znaczników z różną treścią;
zaletą takiego rozwiązania jest większa kontrola nad tak skonstruowaną treścią – w przyszłości
będzie można zmieniać rozmiar całego takiego wpisu, kolor czcionki czy nawet tła</p>
Można nawet dodawać treść bez dodatkowych znaczników – i tak się wyświetli.
</article>
```

Tag <section> natomiast powinien służyć do grupowania większej ilości elementów typu article, chociaż tak jak w poprzednim wypadku umieszczona treść pomiędzy nim może być dowolna.

Przykład:

```
<section>
<p>Nowy tekst części tworzonej strony; może to być dowolna treść. Dzięki umieszczeniu jej w
dodatkowym znaczniku możemy decydować o jej wielkości, przyleganiu do innego elementu oraz
możemy, jako całość, modyfikować (np. zmieniać kolor, tło, położenie wysokości względem
pozostałych elementów itp.)</p>
Można nawet dodawać treść bez dodatkowych znaczników – i tak się wyświetli.
</section>
```

Znacznik `` służy do wydzielenia w tekście (`<p>`, `<article>`) jego fragmentu, dla którego chcemy zmienić właściwości (podświetlić go lub odznaczyć większym rozmiarem czcionki). Poza tym znacznik ten może pełnić podobną rolę do `<div>` z tą różnicą, że jego zakończenie nie powoduje przeniesienia następnego elementu dokumentu do nowej linii.

Przykład:

```
<p>To jest jakiś nowy paragraf; <span>Ten fragment jest ujęty w znacznik span</span>, a ten już nie</p>
```

Oczywiście powyższy przykład na chwilę obecną nic nie zmienia (sam w sobie znacznik `span` jest „niewidoczny”). Przy omawianiu CSS ukaże się jego „potencjał”.

Wyświetlenie niektórych znaków, w tym białych spacji oraz przełamania tekstu (poza poznanym znacznikiem `
`) jest niemal niewykonalne. Z pomocą przychodzi specjalny element `<pre>`.

Przykład:

```
<pre>To jest tekst predefiniowany. Rozpoznaje wiele spacji białych, znaki specjalne &%^?/, a nawet znaczniki, wyłączając z tego jednak znaczniki .
```

Tak samo rozpoznawane są znaki nowych linii

dzięki czemu można pisać swobodnie -

```
HTML nie zmieni naszej koncepcji.</pre>
```

Możemy także użyć tagu `<cite>` by uzyskać efekt cytowania kogoś:

Przykład:

```
<cite>Tutaj cytowany fragment</cite>
```

Specjalnym tagiem `<address>` można otoczyć adres kontaktowy.

Przykład:

```
<address>
```

```
http://www.example.com<br/>
```

```
Przykładowa i spółka <br/>
```

```
ul. Nieznana 14 m. 4 <br/>
```

```
00-000 Niemiasto<br/>
```

```
Polska
```

```
</address>
```

3. Elementy tabelaryczne

Tabele w HTML służą do wyświetlania danych statystycznych, informacyjnych lub innych (np. bazodanowych). NIE NALEŻY stosować ich do tworzenia układu strony HTML.

a) `<table>` - pomiędzy tym znacznikiem umieszcza się cały układ tabeli – nagłówki, wiersze oraz kolumny. Możliwe atrybuty:

- `border` – określa czy komórki tabeli powinny mieć obramowanie czy nie (wartość „1” jeżeli tak, „” jeżeli mają nie posiadać obramowania)

b) `<tr>` - znacznik określa rozpoczęcie nowego wiersza tabeli

c) `<th>` - określa kolumnę nagłówka (jeżeli chcemy np. 3 kolumny w każdym wierszu to należy wstawić 3 znaczniki tego typu w zawartość znacznika `<tr>`).

- `scope` – atrybut określa czy wskazany nagłówek dotyczy się kolumny (`col`), wiersza (`row`), grupy kolumn (`colgroup`) czy grupy wierszy (`rowgroup`)

d) `<td>` - wstawia komórkę danych do wiersza (jeżeli wstawiliśmy 3 kolumny nagłówka to w

każdym kolejnym wierszu powinniśmy wstawić 3 znaczniki td)

Zarówno <th> jak i <td> mogą posiadać następujące atrybuty:

- colspan – (wartość liczbowa) scala określoną (podana w wartości atrybutu) ilość kolumn w jedną (dla wartości 2 tego atrybutu użytego w pierwszym znaczniku td wybranego wiersza zostaną połączone odpowiednio dwie pierwsze komórki w tym wierszu)
- headers – (wartość liczbowa) określa jeden lub więcej nagłówków komórek, do których wskazana (edytowana) komórka się odnosi
- rowspan – (wartość liczbowa) określa ile komórek w kolejnych wierszach ma zostać złączonych z aktualnie edytowaną

e) <thead> - określa nagłówek tabeli; jest elementem grupującym wszystkie kolumny nagłówka tabeli

f) <tbody> - określa ciało tabeli; jest elementem grupującym w sobie komórki (całe wiersze ze wszystkimi kolumnami), które zawierają w sobie określone dane

g) <tfoot> - określa stopę tabeli (zakończenie); jest elementem grupującym kolumny (rzadko wiersze) będące zakończeniem tabeli (np. podsumowania)

h) <col> - umożliwia grupowe określenie właściwości danej kolumny bądź grupy kolumn; efekt stosowany jest dla wszystkich wierszy; wszystkie tego typu elementy muszą znaleźć się w zbiorczym tagu <colgroup>

i) <colgroup> - znacznik pozwala na zdefiniowanie właściwości formatowania grupy/grup kolumn. Jeżeli chcemy ustawić osobne właściwości dla każdej z kolumn należy wewnątrz opisywanego znacznika wstawić odpowiednią ilość znaczników <col>

Znaczniki <col> oraz <colgroup> posiadają jeden unikatowy parametr:

- span – określa ilość zgrupowanych ze sobą kolumn, dla których ustawione zostaną właściwości (bądź jaki zasięg ma posiadać znacznik colgroup)

j) <caption> - w znaczniku umieszcza się wyświetlany podpis tabeli

Przykład tabeli:

1) prosta:

```
<table border="1">
  <caption>Przykładowa tabela</caption>
  <tr>
    <th>ID</th>
    <td>Imię</td>
    <td>Nazwisko</td>
    <td>Pesel</td>
  </tr>
  <tr>
    <td>1</td>
    <td>Justyna</td>
    <td>Milla</td>
    <td>888888888888</td>
  </tr>
```

```

<tr>
  <td>2</td>
  <td>Martyna</td>
  <td>Jurat</td>
  <td>888888888888</td>
</tr>
<tr>
  <td>3</td>
  <td>Paweł</td>
  <td>Mall</td>
  <td>888888888888</td>
</tr>
<tr>
  <td>3</td>
  <td colspan="3">Ogółem osób</td>
</tr>
</table>

```

2) rozbudowana (wraz z właściwościami kolumn)

```

<table border="1">
  <caption>Przykładowa tabela</caption>
  <colgroup>
    <col style="background-color:yellow"/>
    <col span="2" style="background-color:green"/>
    <col style="background-color:#5e5e5e"/>
  </colgroup>
  <tfoot>
    <tr>
      <td>3</td>
      <td colspan="3" style="background-color:white">Ogółem osób</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>1</td>
      <td>Justyna</td>
      <td>Milla</td>
      <td>888888888888</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Martyna</td>
      <td>Jurat</td>
      <td>888888888888</td>
    </tr>
    <tr>
      <td>3</td>
      <td>Paweł</td>
      <td>Mall</td>
      <td>888888888888</td>
    </tr>
  </tbody>
</table>

```

```

</tbody>
<thead>
  <tr>
    <th>ID</th>
    <td>Imię</td>
    <td>Nazwisko</td>
    <td>Pesel</td>
  </tr>
</thead>
</table>

```

4. Elementy formularzy WWW

Formularze w języku HTML pełnią bardzo ważną rolę – umożliwiają wprowadzanie danych przez użytkownika, dzięki którym możliwe jest np. większe spersonalizowanie strony. Dane mogą być dowolne – informacje tekstowe, predefiniowane (wybór z gotowej listy produktów czy właściwości) lub też pliki, które użytkownik zechce przesłać. Formularzem jest chociażby wyszukiwarka internetowa (wpisujemy frazy do znalezienia); to pola formularzy pozwalają zmieniać kolor na stronie czy język (np. strony wielojęzyczne). Wprowadzone treści przesyłane są do odpowiednich skryptów języków wykonywanych po stronie serwera (PHP, AS, JSP, Python) jednak istnieje także możliwość wysłania danych na wskazaną pocztę elektroniczną.

- a) `<form>` - znacznik otwiera nowy formularz. To w jego ciele muszą znaleźć się kolejne pola formularza (w przeciwnym wypadku przeglądarka nie będzie „wiedziała” co ewentualnie zrobić z wprowadzonymi danymi). Parametry:
- `method` – określa w jaki sposób mają zostać przesłane dane z tworzonych formularzy. Może przyjąć dwie wartości: „get” (domyślna wartość; przesyła dane jako zmienne poprzez adres URL widoczny w pasku adresu strony; posiada limit 3000 znaków, dane są widoczne dla użytkownika więc nie nadaje się do przesyłania ważkich danych) oraz „post” (dane przesyłane wewnątrz HTTP request – nie są widziane dla użytkownika; nie posiadają limitu rozmiaru, a poszczególne wartości nie mogą zostać bezpośrednio podmienione bądź zapisane przez użytkownika portalu)
 - `action` – określa gdzie mają zostać przesłane dane – przeważnie jest to odpowiedni skrypt języka wykonywanego po stronie serwera; formularz można także przesłać sam do siebie – poprzez język skryptowy (taki jak JavaScript) również można obsłużyć przesłane dane (np. wyświetlić nazwę wprowadzonego użytkownika). Wreszcie dane można przesłać poprzez dowolny, inny niż WWW protokół obsługiwany przez przeglądarkę – np. na wspomnianą pocztę (wartość pola wygląda wtedy następująco: „<mailto:poczta@naserwerze.pl>”). Aby jednak przesłać dane w ten sposób trzeba posiadać zainstalowany odpowiedni program (w tym wypadku program pocztowy). Listę obsługiwanych protokołów można znaleźć w sieci Internet.
 - `id` – identyfikator formularza (nazwa formularza)
 - `enctype` – określa jak dane z formularza powinny zostać zakodowane w momencie przesyłania; możliwe wartości: „application/x-www-form-urlencoded” (domyślne; każdy znak jest wstępnie kodowany - białe spacje zamieniane są na znaki „+”, a znaki specjalne na wartość heksadecymalną ASCII), „text/plain” (spacje zamieniane są na „+”, lecz znaki specjalne nie są zamieniane) lub „multipart/form-data” (żaden ze znaków nie jest zamieniany; takie kodowanie jest wymagane gdy przez formularz przesyłane są pliki).
 - `accept-charset` – umożliwia wstawianie kodowania znaków w formularzu; przykładowe wartości: „ISO-8859-2”, „ISO-8859-1”, „Windows-1250”, „UTF-8”, „UTF-16” itd.
 - `autocomplete` – możliwe wartości to „on” lub „off”; określa czy pola we wskazanym formularzu mają być automatycznie uzupełniane.
 - `novalidate` – jedyna możliwa wartość „novalidate”; jeżeli ustawiona to dane z formularza nie będą sprawdzane i zamieniane na czytelne (znaki specjalne będą konwertowane na wartości

szesnastkowe, spacje na znaki „+”)

- target – określa gdzie mają zostać wyświetlone wyniki (domyślnie `_self`; atrybut identyczny jak w znaczniku `<a>`).

b) `<input>` - wstawia nowe pole w formularzu; wygląd oraz widoczność określane jest bezpośrednio przez typ pola. Parametry:

- type – określa typ wstawianego pola; HTML5 umożliwia wstawianie następujących rodzajów pól:

1) button – wstawia nowy przycisk (głównie do użycia wraz ze skryptami JavaScript)

2) checkbox – wstawia pole do zaznaczania

3) color – wstawia pole wyboru koloru (chwilowo obsługiwane przez Chrome oraz Operę)

4) date – wstawia pole umieszczania daty (nieobsługiwane przez wszystkie przeglądarki)

5) datetime – wstawia pole do przechwytywania daty i czasu (nieobsługiwane przez wszystkie przeglądarki)

6) datetime-local – jak powyżej, lecz dla czasu lokalnego

7) email – pole do wstawiania adresu email (jeżeli wprowadzony adres jest niepoprawny użytkownik zostanie o tym poinformowany)

8) file – tworzy pole umożliwiające wybranie pliku do przesłania przez formularz

9) hidden – pole ukryte (niewidoczne dla użytkownika); stosuje się je przeważnie do przesyłania danych, które ważne są z punktu widzenia programisty, jednak niepożądane jest by ktokolwiek modyfikował ich wartość (przykładowo wstawiane są odpowiednie informacje na podstawie typu użytkownika, godziny bądź dnia)

10) image – wstawia przycisk wysłania danych formularza jako obraz

11) month – wstawia pole pozwalające na podanie miesiąca oraz roku (nie obsługiwane przez wszystkie przeglądarki)

12) number – wstawia pole umożliwiające podawanie tylko liczbowych danych (pozostałe nie zostaną wpisane)

13) password – wstawia pole, w którym wpisywane znaki są automatycznie ukrywane za pomocą znaku maskującego (gwiazdek, kółek itp.)

14) radio – wstawia pole do zaznaczenia; w przeciwieństwie do pól checkbox, które pozwalają na zaznaczenie kilku wartości (np. 5 pól checkbox w danym formularzu), typ radio umożliwia zaznaczenie tylko jednego obiektu tego typu (stosowany przy jednoznacznych odpowiedziach, np. płeć)

15) range – wstawia obiekt podawania wartości liczbowej za pomocą suwaka

16) reset – wstawia przycisk czyszczący ustawienia wszystkich pól danego formularza

17) search – wstawia pole umożliwiające wpisywanie frazy do wyszukania np. po zawartości strony; ogólnie pole to nie różni się od pól do wprowadzania normalnego tekstu poza dodatkową opcją auto uzupełniania pola (na podstawie wcześniejszych wartości) oraz przycisku kasującego wprowadzoną frazę.

18) submit – wstawia przycisk wysyłania danych z formularza pod adres wskazany w parametrze action znacznika `<form>`

19) tel – wstawia pole parsujące numery telefoniczne (nieobsługiwane przez przeglądarki)

20) text – wstawia jednoliniowe pole do wprowadzania tekstu (domyślnie ma szerokość odpowiadającą 20 znakom tekstu)

21) time – wstawia obiekt umożliwiający wprowadzanie czasu

22) url – wstawia pole umożliwiające wprowadzanie odnośników URL

23) week – wstawia pole umożliwiające ustawienie tygodnia oraz roku

Ponadto pola typu range oraz number mogą zawierać dodatkowe atrybuty (atrybuty te dodaje się do znacznika `input`):

I) step – określa co którą wartość ma zmieniać się wartość pola (domyślnie zwiększenie/zmniejszenie odbywa się o wartość 1; można ustawić dowolną)

II) min – określa minimalną dozwoloną wartość przekazywaną przez pole

III) max – określa maksymalną dozwoloną wartość przekazywaną przez pole

Dla pola typu file możliwe jest ustawienie następującego atrybutu:

I) accept – określa się jakiego typu pliki możemy przesłać przez formularz (np. „image/*”, „image/png”, „video/*”, „text/*” itd.)

II) multiple – przyjmuje wartość „multiple”; umożliwia przesłanie przez jedno pole typu file wielu plików (umożliwia wybór kilku plików poprzez kombinację klawiszy myszy z ctrl, shift lub poprzez grupowe zaznaczenie plików)

Dla pól radio oraz checkbox można dodatkowo ustawić jeden parametr:

I) checked – przyjmuje tylko wartość „checked”; jeżeli ustawiony to wskazane pole będzie automatycznie zaznaczone

Dla pól typu submit oraz image możliwe są następujące atrybuty:

I) formenctype – atrybut identyczny do atrybutu enctype <form>

II) formaction - atrybut identyczny do atrybutu action <form>

III) formmethod - atrybut identyczny do atrybutu method <form>

IV) formtarget - atrybut identyczny do atrybutu target <form>

V) formnovalidate - atrybut identyczny do atrybutu novalidate <form>

Pole image może być użyte z następującymi atrybutami:

I) height – wysokość pola (w pikselach)

II) width – szerokość pola (w pikselach)

III) alt – nazwa alternatywna dla obrazu

IV) src – źródło obrazu dla pola input

- value – określa wartość pola jaka zostanie przekazana przez formularz; jeżeli formularz został utworzony wraz z podaną wartością value będzie ona domyślną wartością wskazanego pola

- size – określa widoczną szerokość pola (w ilości znaków)

- maxlength – maksymalna ilość znaków jaka może zostać zapisana w polu

- required – przyjmuje tylko wartość „required”; określa, że dane pole musi zostać wypełnione zanim formularz zostanie wysłany

- readonly – przyjmuje tylko wartość „readonly”; wskazane pole będzie tylko do odczytu

- id – identyfikator pola (jednoznaczna nazwa)

- disabled – przyjmuje tylko wartość „disabled”; wskazane pole będzie wyłączone/zablokowane

- autocomplete – wartość auto uzupełnienia danych można włączyć dla poszczególnych pól (atrybut działa identycznie jak opisywany w form)

- placeholder – wartość przypisana temu atrybutowi pojawia się jako podpowiedź co należy wpisać w dane pole input (podpowiedź znika po wprowadzeniu co najmniej jednego znaku)

- pattern – umożliwia sprawdzanie, czy wprowadzone dane pasują do wzorca

- form – atrybut umożliwia przypisanie znacznika do jednego bądź wielu formularzy; kolejne nazwy formularzy (id) podaje się oddzielone znakiem białej spacji

- autofocus – atrybut przyjmuje wartość „autofocus”; powinien być ustawiony tylko przy jednym polu w danym dokumencie; wybrane w ten sposób pole będzie automatycznie „aktywne” (czyli np. od razu będzie miało w sobie aktywny kursor tekstowy)

- list – określa listę danych, jaka ma zostać dodana do „podpowiedzi” danego pola; należy pamiętać by ustawiając atrybut list nie ustawiać atrybutu input; parametr ten wymaga dodatkowego elementu <datalist>, z którego będzie pobierał wartości.

c) <option> znacznik definiuje nową wartość opcji w listach wyboru. Parametry:

- value – zawiera przekazywaną wartość danej opcji przez formularz

- selected – może posiadać jedynie wartość „selected”; opcja, dla której zostanie ustawiony ten parametr, będzie domyślnie wybrana na liście po załadowaniu formularza

- label – definiuje krótszą nazwę dla danej opcji

- disabled – może posiadać jedynie wartość „disabled”; określa, że wskazana opcja powinna być wyłączona (niemożliwa do wybrania)

d) <datalist> - znacznik przechowujący wartości, które mogą zostać bezpośrednio podstawione w

pole, do którego zostanie podłączony przygotowywany element. Używa atrybutów globalnych; do działania wymagane jest jedynie ustawienie atrybutu id (określającego jego nazwę; po niej będzie łączony ze wskazanym polem listowym); znacznik ten powinien zawierać co najmniej jeden element `<option>` (może zawierać ich dowolną ilość)

e) `<optgroup>` - tag służy do grupowania kilku pól `<option>` pod jedną, wspólną nazwą (lepsza czytelność). Jeden tag `<optgroup>` może zawierać dowolną ilość tagów `<option>`. Parametry:

- label – wyświetlana nazwa tworzonej grupy opcji
- disabled – wskazana grupa powinna być zablokowana (wyłączona)

f) `<select>` - znacznik tworzy w formularzu nowy element typu drop-down list (listę rozwijalną od dołu). W znaczniku powinien zostać zawarty co najmniej jeden element `<option>` (może zawierać ich dowolną ilość) lub co najmniej jeden element `<optgroup>` (może zawierać ich dowolną ilość).

Atrybuty:

- autofocus – możliwa wartość „autofocus”; jeżeli zostanie ustawiona to po załadowaniu strony wskazana lista zostanie automatycznie wybrana jako aktywna
- disabled – wskazana lista powinna być zablokowana
- form – atrybut identyczny jak w elemencie `<input>`
- multiple – możliwa wartość „multiple”; pozwala na jednorazowy wybór wielu elementów na liście
- id – identyfikator listy (jej jednoznaczna nazwa)
- required – atrybut identyczny jak w elemencie `<input>`
- size – wartość liczbowa; określa ile elementów ma być widocznych dla użytkownika (domyślnie 1)

g) `<label>` - element definiuje nazwę (etykietę) pola formularza; może on zawierać dany element w sobie (np. `<input>` lub `<select>`) bądź zostać przypisany poprzez odpowiedni atrybut. Dostępne parametry:

- for – jego wartość powinna zawierać wartość atrybutu id pola formularza, do którego etykieta ma zostać dowiązana.
- form – atrybut identyczny do opisanego w `<input>`

h) `<textarea>` - wstawia pole edycji tekstu wieloliniowego. Liczba znaków jest teoretycznie nieograniczona; przeglądarka wybiera dla niego czcionkę o stałej szerokości znaków (np. Courier). Atrybuty:

- rows – określa liczbę widocznych linii tekstu (wysokość pola – np. 3 linie * wysokość czcionki)
- cols – określa liczbę widocznych kolumn (znaków) tekstu (szerokość pola – np. 50 znaków * szerokość czcionki)
- disabled – atrybut działa jak w `<input>`
- autofocus - atrybut działa jak w `<input>`
- maxlength – maksymalna ilość znaków jakie można wprowadzić w elemencie
- form - atrybut działa jak w `<input>`
- placeholder - atrybut działa jak w `<input>`
- readonly - atrybut działa jak w `<input>`
- required - atrybut działa jak w `<input>`
- id - atrybut działa jak w `<input>`
- wrap – określa jak tekst ma być łamany do kolejnych linii po przesłaniu na serwer. Domyślnie nie jest łamany po przesłaniu (wartość „soft”). Będzie łamany (zostanie zachowany układ linii) przy wartości „hard”; jednak ustawienie „hard” jest dozwolone tylko w przypadku gdy został ustawiony parametr cols (liczba kolumn).

i) `<button>` - znacznik tworzy w formularzu nowy przycisk. W przeciwieństwie do przycisku tworzonego za pomocą pola `<input>` przyciski tworzone za pomocą opisywanego tagu mogą

zawierać dowolną treść – tekst, obraz bądź dowolny inny element HTML (pomiędzy `<button></button>` można wstawić np. znacznik `` ze źródłem do obrazu). Ponadto nie musi on mieć jedynie opcji wysłania/wyczyszczenia danych przez formularz – za pomocą języka JavaScript można nadać mu dowolną funkcjonalność. Parametry:

- autofocus - atrybut działa jak w `<input>`
- disabled - atrybut działa jak w `<input>`
- form - atrybut działa jak w `<input>`
- formaction - atrybut działa jak w `<input>`
- formenctype - atrybut działa jak w `<input>`
- formmethod - atrybut działa jak w `<input>`
- formnovalidate - atrybut działa jak w `<input>`
- formtarget - atrybut działa jak w `<input>`
- id - atrybut działa jak w `<input>`
- value - atrybut działa jak w `<input>`
- type – może zawierać jedną z trzech wartości: „button” - tworzy klikalny przycisk ogólnego przeznaczenia; „submit” - tworzy klikalny przycisk, po kliknięciu którego wysłany zostanie cały formularz; „reset” - tworzy klikalny przycisk czyszczący z wartości wszystkie pola formularza

j) `<fieldset>` - dzięki temu elementowi można pogrupować kilka pól formularza; grupa będzie odznaczona poprzez obrysowanie grupy linią. Parametry:

- disabled - atrybut działa jak w `<input>`
- form - atrybut działa jak w `<input>`
- id - atrybut działa jak w `<input>`

j) `<legend>` - dodaje nazwę grupy elementu `<fieldset>`

k) `<keygen>` - dodaje do formularza pole z możliwością wykorzystania algorytmu szyfrującego dane z innych pól. Należy pamiętać, że pole to tylko wskazuje na ewentualne żądanie zakodowanie danych – jednak samo kodowanie odbywa się po stronie serwera. Parametry:

- autofocus - atrybut działa jak w `<input>`
- disabled - atrybut działa jak w `<input>`
- form - atrybut działa jak w `<input>`
- id - atrybut działa jak w `<input>`
- challenge – wartość „challenge”; określa, że wartość pola `<keygen>` powinno zostać przetestowane przy wysłaniu
- keytype – określa algorytm kodowania danych; możliwe wartości: „rsa”, „dsa”, „ec”

l) `<output>` - znacznik wyświetla wynik kalkulacji i/lub przetwarzania danych przez formularz (musi działać wraz z językiem skryptowym, np. JavaScript). Atrybuty:

- for - atrybut działa jak w `<label>`
- id - atrybut działa jak w `<input>`
- form - atrybut działa jak w `<input>`

Przykład użycia:

```
<form action="mailto:adres@poczty.pl" enctype="plain/text" method="post">
<label for="imie">Imię: </label>
<input type="input" id="imie" name="imie"/><br/>
<label for="nazwisko">Nazwisko: </label>
<input type="input" id="nazwisko" name="nazwisko"/><br/>
<label for="pesel">Pesel: </label>
<input type="input" id="pesel" name="pesel"/><br/>
<label for="auto">Ulubione auto: </label>
```

```

<select id="auto">
  <optgroup label="europejscie">
    <option value="fiat">Fiat</option>
    <option value="volvo">Volvo</option>
    <option value="citroen">Citroen</option>
    <option value="ferrari">Ferrari</option>
    <option value="rover">Rover</option>
  </optgroup>
  <optgroup label="azjatyckie">
    <option value="honda">Honda</option>
    <option value="toyota">Toyota</option>
  </optgroup>
  <optgroup label="amerykańskie">
    <option value="Ford">Ford</option>
  </optgroup>
</select><br/>
<label for="opis">Napisz coś o sobie: </label>
<textarea id="opis">
</textarea><br/>
<fieldset>
  <legend>Wykształcenie</legend>
  <input type="radio" name="education" value="1"/>podstawowe</br>
  <input type="radio" name="education" value="2"/>średnie</br>
  <input type="radio" name="education" value="3"/>techniczne</br>
  <input type="radio" name="education" value="4"/>wyższe</br>
</fieldset><br/>
<input type="submit" value="Wyślij"/>
</form>

```

5. Elementy list

HTML pozwala na tworzenie elementów listowych. Elementy te pozwalają wypunktować np. tematy wiadomości dostępne na stronie, listę części dostępnych w sklepie, usługi realizowane przez firmę itd.

a) - definiuje pojedynczy element listy; używany w każdym typie listy (uporządkowanej, nieuporządkowanej czy menu)
 - value – atrybut określa wartość danego elementu na liście; jeżeli tworzymy listę uporządkowaną to od tej wartości nastąpi numeracja (zwiększana co jeden) kolejnych elementów na liście.

b) - tag rozpoczynający listę nieuporządkowaną; może zawiera dowolną liczbę elementów listy

```

<ul>
<li>Pierwszy element</li>
<li>Drugi</li>
<li>Trzeci</li>
<li>...</li>
<li>KONIEC</li>
</ul>

```

c) - tag rozpoczynający listę uporządkowaną; może zawierać dowolną liczbę elementów

- reversed – jeżeli ustawiony to elementy porządkowane są w odwrotnej kolejności
- start – określa od której wartości ma być porządkowana lista (startowa wartość)
- type – określa jakiego typu mają być kolejne „liczby” porządkowe: „1” (liczby arabskie, domyślnie), „a” (kolejne litery alfabetu, małe), „A” (kolejne litery alfabetu, duże), „i” (liczby rzymskie, małe), „I” (liczby rzymskie, duże)

```
<ol>
<li>Pierwszy element</li>
<li>Drugi</li>
<li>Trzeci</li>
<li>...</li>
<li>KONIEC</li>
</ol>
```

```
<ol start="10" reversed="reversed">
<li>Pierwszy element</li>
<li>Drugi</li>
<li>Trzeci</li>
<li>...</li>
<li>KONIEC</li>
</ol>
```

```
<ol type="i">
<li>Pierwszy element</li>
<li>Drugi</li>
<li>Trzeci</li>
<li>...</li>
<li>KONIEC</li>
</ol>
```

d) <dl> - element tworzy listę opisową

e) <dt> - tworzy w liście opisowej nazwę-warunek

f) <dd> - tworzy w liście opisowej opis pola zdefiniowanego w tagu <dt>

Przykład użycia:

```
<dl>
<dt>Roślina</dt>
<dd>Zielona, z czerwonym kwiatem</dd>
<dt>Drzewo</dt>
<dd>Ogromne, dające sporo cienia</dd>
<dt>Trawa</dt>
<dd>Pożółkła, spalona słońcem</dd>
</dl>
```

10. Elementy graficzne na stronie.

Ludzie bardziej niż tekst cenią sobie dobrą oprawę graficzną. Dlatego strony internetowe, by zwrócić „na siebie” uwagę użytkownika, muszą posiadać elementy graficzne. HTML pozwala na wstawianie obrazów na stronie internetowej; pozwala także na rysowanie grafiki (tylko HTML5).

a) - tworzy element graficzny na stronie z podanego źródła w parametrze src. Źródłem może być dowolny obraz jpg, png, gif, tiff, svg oraz inny, obsługiwany przez przeglądarkę WWW.

Posiada następujące parametry:

- src – jako wartość przyjmuje ścieżkę do wstawianego obrazu; obraz może znajdować się lokalnie w katalogu strony (np. „obrazek.png”), w podkatalogu (np. „./pliki/obrazek.png”) lub na serwerze zewnętrznym (np. „<http://www.example.com/obrazek.png>”). Jeżeli obraz nie będzie mógł zostać z jakichkolwiek powodów załadowany przez przeglądarkę zostanie wyświetlony błąd obrazu – złamana ikona, puste miejsce z ikoną X itp.
- alt – przechowuje tekst z podpisem obrazu; podana nazwa wyświetli się jeżeli obraz nie będzie mógł zostać załadowany (bądź gdy najedziemy na niego kursorem)
- width/height – parametry umożliwiają ustawienie szerokości/wysokości obrazu w pikselach; jeżeli obraz źródłowy będzie większy to zostanie on przeskalowany do podanych ram
- ismap – może przyjąć tylko wartość „ismap”; określa po stronie serwera, że obraz jest mapą. Używane tylko wraz z językami przetwarzanymi po stronie serwera! (np. PHP, ASP, JSP itp.)
- usemap – określa po stronie klienta, że obraz jest mapą. W wartości podajemy nazwę mapy, która ma zostać wykorzystana (poprzedzając nazwę znakiem #)

b) <map> - tag definiujący mapę obrazu po stronie klienta. W jego ciele mogą znajdować się znaczniki <area> definiujące odpowiednie fragmenty mapy obrazu (hiperłącza) wraz z akcją jaka ma zostać podjęta po ich aktywacji.

- name - znacznik określa nazwę mapy

c) <area> - znacznik definiujący odnośniki (linki) wewnątrz obrazu-mapy.

- alt – nazwa definiowanej części obrazu (obligatoryjna)
 - href – definiuje hiperłącze do docelowego dokumentu
 - shape – definiuje figurę (kształt) jaką ma być tworzona strefa; istnieją cztery możliwe wartości: „default” (przeważnie prostokąt), „rect” (prostokąt), „circle” (okrąg) oraz „poly” (dowolny wielokąt)
 - coords – punkty koordynujące rozmiar strefy; ilość punktów oraz ich kolejność zależą od podanej figury w polu shape (dla kwadratu 4 wartości: x1,y1 stanowiące górny lewy wierzchołek oraz x2, y2 stanowiące prawy dolny wierzchołek); dla okręgu 3 wartości: x, y stanowiące punkt centralny oraz r będące promieniem koła; wielokąty w zależności od oczekiwanego efektu).
- Ponadto area posiada atrybuty: rel, download, target, media , type które były już wcześniej omawiane.

```



```

```

<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="http://www.w3schools.com/tags/sun.htm"/>
  <area shape="circle" coords="90,58,3" alt="Mercury" href="http://www.w3schools.com/tags/mercur.htm"/>
  <area shape="circle" coords="124,58,8" alt="Venus" href="http://www.w3schools.com/tags/venus.htm"/>
</map>
```

HTML5 wprowadza ponadto nowy element strony – płótno (canvas). Aby go jednak poprawnie

użyć trzeba wykorzystać skrypty po stronie klienta. Ponieważ na tą chwilę nie posiadamy o nich wiedzy element ten zostanie omówiony w trakcie zapoznawania się z nimi (programowanie aplikacji internetowych – przyszły semestr).

6. Ramki

HTML5 wspiera tylko jeden typ ramek -iframe (tzw. ramka pływająca). Swego czasu bardzo popularne było tworzenie stron w oparciu o ramki (frame); W HTML5 ramki zostały całkowicie wyeliminowane z użytku.

<iframe> - znacznik pozwalający na wstawienie pływającej ramki do obecnie wyświetlanego dokumentu; ramka ta może zawierać w swojej treści inny dokument html, np. zupełnie inną stronę WWW i/lub następny dokument html.

- height/width – wysokość/szerokość ramki; jeżeli nie ustawione zostaną przyjęte wartości domyślne
- id – identyfikator/nazwa ramki
- src – źródło ładowanego dokumentu; może to być dokument znajdujący się w zbiorze lokalnym (dysk twardy) jak i adres zewnętrzny
- srcdoc – jako wartość wstawia się kod HTML, który ma zostać wyświetlony w ramce
- seamless – ustawienie tego parametru powoduje, że pływająca ramka wygląda jak fragment całej tworzonej strony (nie ma obramowania)
- sandbox – umożliwia zarządzanie ograniczeniami działania załadowanego dokumentu w ramce. Jeżeli wartość pola będzie pusta wtedy wszystkie ograniczenia zostaną włączone. Możliwe wartości: „allow-script” (wykonywanie skryptów z zewnętrznego źródła), „allow_forms” (wysyłanie danych z form zewnętrznego dokumentu), „allow-top-navigation” (umożliwia dokumentowi z ramki ładowanie zawartości z głównego dokumentu) lub „allow-same-origin” (treść ładowanego w ramce dokumentu może być traktowana jako zawartość dokumentu głównego).

```
<iframe src="http://www.w3schools.com"></iframe>
```

Instrukcja nie przedstawia wszystkich znaczników HTML. Niektóre z nich są bowiem niemal niewykorzystywane, inne straciły na znaczeniu wraz z pojawieniem się stylów CSS.

Zadania do samodzielnego wykonania.

1. Proszę stworzyć szkielet prostej strony. Strona powinna zawierać:

- sekcję tytułową
- sekcję z menu
- obszar właściwy strony
- stopkę

Proszę pamiętać by szkielet został dobrze zaprojektowany, tzn. aby każda część strony była w odpowiednim elemencie blokowym.

Na podstawie obecnie zdobytej wiedzy w belce tytułowej powinien znaleźć się duży napis z nazwą strony, np. „Moja nazwa strony WWW” (ale adekwatna dla treści strony)

W belce menu powinny znaleźć się następujące odnośniki: Strona główna, Co oferujemy, Sklep, Wypożyczalnia, Galeria, O nas, Kontakt.

W obszarze właściwym strony ciężko jest teraz cokolwiek wpisywać (ma to być szkielet). Dlatego dobrym rozwiązaniem jest uzupełnić tekst obszaru właściwego losowymi wartościami (w internecie można znaleźć programy-generatory losowej treści).

W stopce proszę umieścić informacje o autorze, adres e-mail (nie musi być prawdziwy) oraz informacje o prawach autorskich (znak specjalny ©) oraz rok.

2. Proszę stworzyć kolejne szkielety stron dla kolejnych podstron. Dobrym rozwiązaniem byłoby całą „stronę” zapisać w jednym katalogu; poprzednio tworzony plik powinien zyskać nazwę index.htm lub index.html. Kolejne podstrony (oferta, sklep, wypożyczalnia, galeria, o nas, oraz kontakt) można nazwać dowolnie; pamiętać należy jedynie o rozszerzeniach plików – muszą to być .htm lub .html.

3. Na stronie galerii proszę dodać 10 losowo wybranych z internetu zdjęć. Zdjęcia na tej stronie powinny mieć ściśle określoną wielkość – po kliknięciu na którekolwiek z nich powinno otworzyć się w nowej zakładce/oknie przeglądarki w pełnym wymiarze.

4. Na stronie o nas proszę wstawić sekcję adresową z wypisanym adresem i telefonem (dane mogą być fikcyjne). Proszę także umieścić zdjęcie, najlepiej siedziby jakiejś firmy zajmującej się sprzedażą/wypożyczeniem samochodów, z którego można będzie zrobić mapę obrazową. Proszę za pomocą znacznika area utworzyć figurę (dowolną) obrysującą jeden z wybranych domów (bądź jego fragmentu), po kliknięciu której przeniesieni zostaniemy do nowej strony, na której będzie powiększone zdjęcie klikniętego domu (można powiększyć ten z obrazka za pomocą dowolnego programu graficznego).

5. Na ostatniej stronie kontakt proszę utworzyć formularz kontaktowy – powinien zawierać sekcję rodzaj informacji (lista rozwijalna z co najmniej 4 tematami do wyboru), tytuł właściwy, treść, podpis, pliki do przesłania, adres e-mail.

6. Na stronach sklep oraz wypożyczalnia należy utworzyć tabelę HTML.

Dla sklepu tabela ma mieć nagłówki:

| Id | nazwa | przebieg | wyposażenie | zdjęcie | szczegółowy opis | porównaj |
|----|-------|----------|-------------|---------|------------------|----------|
|----|-------|----------|-------------|---------|------------------|----------|

Dla wypożyczalnia:

| Id | nazwa | wyposażenie | koszt | taryfa | ilość | dostępność | szczegóły | porównaj |
|----|-------|-------------|-------|--------|-------|------------|-----------|----------|
|----|-------|-------------|-------|--------|-------|------------|-----------|----------|

Do każdej z tabel należy dodać ok. 10 wpisów. Kolumny „porównaj” mają posiadać element typu checkbox (wraz z odpowiednimi parametrami); Proszę spróbować dodać ten element do odpowiedniego formularza (formularz powinien zostać dodany na końcu tabeli i zawierać przycisk typu submit). W celu wypróbowania działania formularza można spróbować wysłać dane na adres e-mail.

7. Na stronie oferta należy dodać w postaci listy oferowane usługi. Lista powinna być co najmniej raz zagnieżdżona: posiadać główny element listy (np. skup pojazdów) i kilka podpunktów (sprawnych, do 10 lat, z przebiegiem nie większym niż 300.000, itp.) Główne elementy listy mogą być jako nieuporządkowane, natomiast podpunkty powinny być listą numerowaną.

Zadanie dodatkowe:

Proszę zaprojektować dodatkową stronę formularza rejestracyjnego na stronie serwisu. Proszę zastanowić się nad polami takiego formularza (jaki powinien zawierać) i utworzyć je zgodnie ze standardem nowych elementów (pól typu input).