

# MODUŁ 1: Języki znaczników i (X)HTML

## Cel modułu

Moduł służy zapoznaniu z elektronicznymi dokumentami znacznikowym na przykładzie języka HTML. Przyjrzymy się cechom szczególnym języka HTML, zakresowi wspieranych przezeń rozwiązań, oraz wyjaśnimy na jego przykładzie pojęcia metajęzyka, poprawności składniowej, strukturalnej oraz walidacji dokumentu.

## Efekty kształcenia

Niniejsza lekcja służy osiągnięciu następujących efektów kształcenia.

### Wiedza

Po ukończonym kursie student / studentka:

- Charakteryzuje język znaczników i wskazuje cechy różniące go od języków programowania,
- Definiuje pojęcie metajęzyka w kontekście technologii WWW,
- Rozróżnia pojęcia występujące w językach znaczników i objaśnia sposób ich odzwierciedlenia w składni SGML i XML,
- Rozróżnia kryteria poprawności składniowej i strukturalnej dokumentu znacznikowego.

### Umiejętności

Po ukończonym kursie student / studentka:

- Dokonuje walidacji dokumentów znacznikowych przy użyciu odpowiednich narzędzi,
- Tworzy strony WWW w języku HTML,
- Tworzy poprawne składniowo dokumenty XML,
- Samodzielnie uzupełnia swoje umiejętności za pomocą dostępnych w WWW materiałów w języku polskim i angielskim.

## Uzasadnienie

Języki znaczników tworzą ramę koncepcyjną i składniową dla języka HTML służącego budowie stron WWW, a także dla licznych innych języków; zarówno tych służących realizacji warstwy prezentacyjnej, jak też opisujących sposób przetwarzania danych. Ponadto, jak przekonamy się w kolejnych modułach, języki znaczników stanowią, dzięki ich niezależności od platformy języka programowania i systemu operacyjnego, istotny element rozwiązań integrujących oprogramowanie.

## Wprowadzenie

Pojęcie języków znaczników jest znacznie starsze niż komputery i wykorzystywane przezeń różnorodne formaty dokumentów elektronicznych. Przykładem nie-elektronicznego języka znaczników mogą być dla nas np. stosowane w erze maszynopisu różnorodne umowne oznaczenia wskazujące, jak przekazany manuskrypt powinien być wydrukowany przez drukarnię. Mogły one informować np. o tym, jakie części tekstu mają zostać wyróżnione inną czcionką, które akapity powinny zostać objęte ramką, które zdanie powinno zostać umieszczone w notce na marginesie itp.

To jest jakiś fragment maszynopisu. Ilustruje, jak mogą wyglądać znaczniki tworzone ręcznie na dokumencie drukowanym. Są one wówczas naniesione za pomocą innych środków. A co najważniejsze – ich rozpoznanie jest dokonywane przez człowieka.

Inaczej ma się sprawa z dokumentami elektronicznymi. Tutaj zarówno treść i jej opis stanowią po prostu sekwencję bajtów. Trzeba umożliwić komputerowi ich niezawodne rozróżnianie!

Ciąg dalszy tekstu...



Na tym przykładzie możemy sobie wyjaśnić pojęcie znacznika (które zresztą trafnie tłumaczy samo brzmienie tego słowa). Otóż znacznik nadaje określone znaczenie wyodrębnionemu przezeń fragmentowi tekstu. Co z tego wynika?

Po pierwsze, znacznik musi precyzyjnie wyodrębniać, fragment tekstu, do którego ma zastosowanie. Po drugie, intencje komunikowane przez znacznik, czy inaczej – znaczenie z nim związane – muszą być uzgodnione i znane stronie, która będzie odbiorcą, „konsumentem” takiego dokumentu.

Tego typu, uzgadniane ad-hoc języki miały jedną istotną cechę: były interpretowane przez człowieka, który mógł w sytuacjach wątpliwych posłużyć się intuicją, czy wręcz poprosić autora o wyjaśnienie. Ponadto same znaczniki (nanoszone np. ołówkiem czy flamastrem) stanowiły jakby odrębną warstwę ponad tekstem, a więc były od niego łatwo odróżnialne.

Nadejście ery komputerów spowodowało skonstruowanie elektronicznych języków znaczników, które mogłyby być przetwarzane samodzielnie przez oprogramowanie. Tym samym powstał problem zaprezentowania za pomocą łącznej sekwencji bajtów zarówno samego tekstu dokumentu jak i samych nałożonych nań znaczników oraz niezawodnego odróżniania jednych od drugich.

```
546f206a657374206a616b697320667261676d656e74206d61737a796e6f7
06973752e20496c75737472756a652c206a616b206d6f6761207779676c61
646163207a6e61637a6e696b692074776f727a6f6e65207265637a6e69652
06e6120646f6b756d656e636965206472756b6f77616e796d2e205361206f
6e652077f377637a6173206e616e696573696f6e65207a6120706f6d6f636
120696e6e7963682073726f646bf3772e204120636f206e616a77617a6e69
656a737a6520962069636820726f7a706f7a6e616e6965206a65737420646
f6b6f6e7977616e652070727a657a20637a6c6f7769656b612edada496e61
637a656a206d612073696520737072617761207a20646f6b756d656e74616
d6920656c656b74726f6e69637a6e796d692e20547574616a207a6172f377
6e6f2074726573632069206a656a206f706973207374616e6f77696120706
f2070726f7374752073656b77656e636a652062616a74f3772e2054727a65
626120756d6f7a6c69776963206b6f6d70757465726f776920696368206e6
9657a61776f646e6520726f7a72f37a6e69616e696521dada436961672064
616c737a792074656b7374752e2e2e
```

?



Wymagało to przyjęcia ścisłych reguł składniowych i nominowania wybranych znaków jako znaki specjalne, służące konstrukcji znaczników.

Kontrola bieżąca	( )	komentarz odpowiedzi
Zaznacz poprawne zdania na temat elektronicznych języków znaczników:		
Nie sposób za ich pomocą zapisać instrukcji tradycyjnego języka programowania		Czemu nie? Wszystko zależy od tego, jakie znaczniki zaprojektujemy dla naszego języka
Dokument znacznikowy można przetwarzać w zwykłym edytorze tekstu	x	Tak. Dokument znacznikowy wciąż pozostaje dokumentem tekstowym
Zastosowanie znaczników w dokumencie tekstowym sprawia, że staje się dokumentem binarnym		Nie. Dokument znacznikowy wciąż pozostaje dokumentem tekstowym

## HTML jako język znaczników

HTML powstał, na początku lat 1990-tych, początkowo z myślą o reprezentowaniu zwyczajnych statycznych dokumentów połączonych między sobą hipertąciami. Stąd też problem, z jakim musieli zmierzyć się jego autorzy był podobny jak wyżej wspomniane zagadnienie budowy języka znaczników wyznaczającego role fragmentów tekstu na potrzeby ich druku (skład tekstu). Wprawdzie tutaj głównym sposobem wykorzystania dokumentu znacznikowego był nie druk a prezentacja na ekranie w oknie przeglądarki, jednak problem pozostawał zbliżony.

Znaczniki nałożone na dokument HTML tworzą hierarchię, w której można wyodrębnić część nagłówkową (*head*), służącą zgromadzeniu meta-danych umożliwiających poprawną prezentację i obsługę dokumentu oraz część ciała dokumentu (*body*) zawierającą treść dokumentu przewidzianą do pokazania w oknie przeglądarki.

Podział dokumentu na część nagłówkową przypomina nam wzorzec występujący w protokołach komunikacyjnych. Podobnie jak tam – nagłówek zawiera dane, które umożliwiają właściwą interpretację i przetworzenie ciała. Napotykamy więc tutaj drugie już w tym module zastosowanie słowa meta – tym razem w pojęciu metadanych dokumentu.

Pierwszym zaś i w tym module najistotniejszym – będzie jego użycie w pojęciu meta-języka, które objaśnimy w kolejnej sekcji.

Nie zamierzamy tu powielać żadnego z licznych dostępnych samouczków języka HTML, zamiast tego odsyłając do choćby portalu [www.w3schools.com](http://www.w3schools.com), zawierającego zarówno różnorodne wprowadzenia w temat, jak też słownik poszczególnych elementów języka HTML (zawiera strony przykładów pozwalające na przetestowanie tych konstrukcji w praktyce).

Ograniczymy się natomiast do wskazania kilku ważnych etapów rozwoju tego języka:

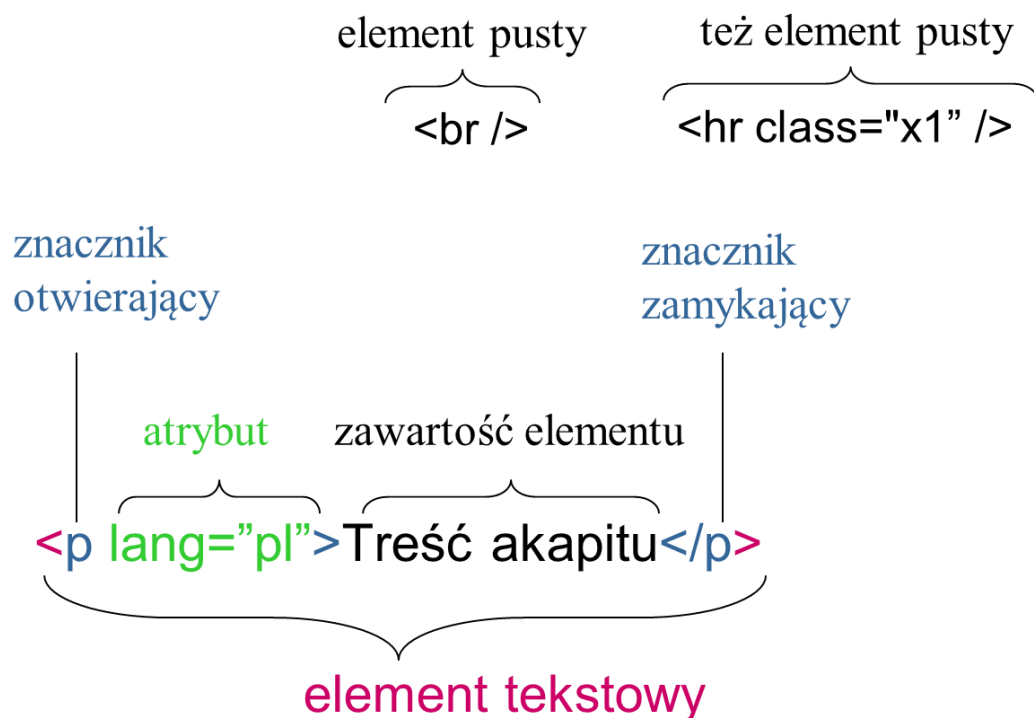
- Powstanie w 1992 roku HTML jako języka znaczników opartego na ramie składniowej języka SGML (wywodzącego swą historię jeszcze z lat 1960-tych języka umożliwiającego definiowanie specjalizowanych, dedykowanych określonym dziedzinom zastosowań, języków znaczników). Język koncentrował się na wyodrębnieniu struktury dokumentu (nagłówki, akapity) oraz rodzajów treści (wyróżnienia, przykłady kodu)
- Stopniowy rozrost języka towarzyszący powstawaniu nowych przeglądarek i jego „zaśmiecenie” znacznikami służącymi wyłącznie określeniu sposobu formatowania (prezentacji) dokumentu (fonty, barwy, efekty tekstu itp.)

- Wprowadzenie rozdzielenia zagadnień treści i formatowania dzięki standardowi CSS (będzie przedmiotem osobnego wykładu)
- Stworzenie XHTML - wariantu języka opartego na XML – charakteryzował się restrykcyjnymi wymaganiami poprawności składniowej, pozwalającymi uprościć przetwarzanie dokumentów przez oprogramowanie (łatwiejszy, mniej absorbujący obliczeniowo, rozbiór dokumentu)
- Wprowadzenie HTML 5 jako języka wzbogaconego o nowe elementy semantyczne, skuteczniej wspierającego konstrukcję aplikacji WWW (elementy o charakterze interakcyjnym) oraz przyjmującego pragmatyczne podejście do kwestii poprawności składniowej oraz rozstrzygania sposobu obsługi błędów składniowych.

Sprecyzujmy też terminologię: znaczniki służą wyodrębnianiu elementów, które mogą posiadać zawartość:

- tekstową;
- pustą;
- mieszaną (tekst a pośród niego inne elementy);
- elementową (element zawiera wyłącznie zagnieżdżone w nim podelementy).

Znaczniki mogą posiadać atrybuty.



Kontrola bieżąca	( )	komentarz odpowiedzi
Zapoznaj się z budową elementu HTML służącego reprezentacji hiperłącza. Jak (w jego najprostszej użytecznej postaci) sklasyfikujemy go według wyżej przytoczonej terminologii?		
Element atrybutowy		Nie. Nie stosujemy takiej terminologii. Wprawdzie posiada on atrybut <code>href</code> , ale rodzaje elementów wyróżniamy według tego, co jest między znacznikiem otwierającym a zamykającym.
Element pusty		Nie. Zawiera on zwykle tekst wyświetlany jako etykieta znacznika.
Element tekstowy	x	Tak. Zawiera on zwykle tekst wyświetlany jako etykieta znacznika.
Element o zawartości mieszanej		Nie. Przynajmniej w najprostszej jego postaci – nie zagnieżdżamy w nim innych (pod)elementów.
Element o zawartości elementowej		Nie. Przynajmniej w najprostszej jego postaci – nie zagnieżdżamy w nim innych (pod)elementów.

Jako podsumowanie tej sekcji przytoczmy prawie minimalny poprawny dokument HTML, który może posłużyć jako szkielet dla naszej strony WWW. Element `meta` służący wskazaniu sposobu kodowania znaków można byłoby pominąć, o ile zadbalibyśmy o dostarczenie tej informacji do programu-konsumenta strony w inny sposób.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8" />
```

```
    <title>Tytuł dokumentu</title>
```

```
  </head>
```

```
  <body>
```

```
    Hello!
```

```
  </body>
```

```
</html>
```

## Język a metajęzyk - poprawność składniowa a strukturalna

Zagadnieniem, na które chcemy zwrócić uwagę w dzisiejszej lekcji pozostaje natomiast pojęcie poprawności składniowej i jego odróżnienie od poprawności strukturalnej.

Jak mogliśmy się zorientować już przy omówieniu wprowadzającego przykładu dotyczącego składu tekstu – potrzeby dotyczące rozlicznych znaczników mogły być bardzo rozbudowane i zróżnicowane zależnie od indywidualnych potrzeb. Z tego względu już język SGML zaprojektowano nie jako wyczerpującą specyfikację wszystkich możliwych znaczników, ale jako otwarty na definiowanie nowych znaczników. Tym samym, zamiast dostarczyć gotowego, zamkniętego katalogu znaczników, SGML zapewnił:

- reguły składniowe, mówiące jak powinny być budowane znaczniki;
- język definicji „typów dokumentów”, pozwalający definiować nowe znaczniki i określać dlań: nazwy, typ dopuszczalnej zawartości, nazwane atrybuty wraz z ich dziedzinami.

Zagadnienia definiowania typów dokumentów (albo, innymi słowy, *schematów dokumentów*) będą przedstawione w osobnym wykładzie.

Zwróćmy uwagę, że można zatem powiedzieć, że SGML dostarczał jego użytkownikom narzędzi pozwalających zdefiniować **nowy własny język znaczników** oparty na wspólnych regułach składniowych SGML.

Według analogicznych założeń zaprojektowano, ściśle zresztą inspirowany SGML-em język XML dedykowany dla definiowania różnorodnych języków znaczników na potrzeby Internetu.

Z powyższych względów SGML i XML są niekiedy nazywane **meta-językami**.

Ten wywód pozwala nam łatwo wyjaśnić różnicę pomiędzy kryterium poprawności składniowej a poprawnością strukturalną.

Otóż poprawność składniowa (ang. *well-formedness*) jest kryterium słabszym (mniej wymagającym) i przewiduje, że dany dokument znacznikowy jest zgodny z regułami składniowymi wspólnej ramy składniowej danego standardu (czyli SGML czy XML). W praktyce oznacza to, że znaczniki są w dokumencie opatrzone odpowiednimi znakami i są poprawnie pozagnieżdżane. Pozwala to oprogramowaniu (nawet nieposiadającemu wiedzy o semantyce poszczególnych znaczników) na skutecznie zidentyfikowanie hierarchii elementów dokumentu i ich odwzorowanie w pamięci komputera. Przykładem niech będzie chociażby programistyczny edytor tekstu, który zdolny jest taki dokument znacznikowy zaprezentować w postaci zwijalnych/rozwijalnych szczebli hierarchii.

Wyższy poziom poprawności stanowi poprawność strukturalna (*validity*). Wymaga ona jako warunku koniecznego poprawności składniowej. Dodatkowo przewiduje ona, że:

- oprogramowaniu znane są kryteria dotyczące znaczników dopuszczalnych oraz wymagalnych w dokumencie danego języka (tj. zostały one wyspecyfikowane w odpowiednim dokumencie schematu, który może (tak jak np. w przypadku HTML czy XHTML) mieć status standardu (w dokumentach XML nie ma obowiązku definiowania dlań schematów),
- dokument zawiera znaczniki w pełni zgodne z tą definicją.

W przytoczonym wyżej przykładzie kodu, pierwszy wiersz stanowi właśnie deklarację wskazującą nazwę języka zastosowanego w danym dokumencie znacznikowym (jest nim tutaj HTML5). Dostarczenie tej informacji jest niezbędne, by można było mówić o poprawności strukturalnej. Więcej szczegółów na temat definiowania typów (albo schematów) dokumentu znacznikowego poznamy w modułach poświęconych językowi XML.

Przykładowo, dla dokumentu HTML może to oznaczać, że nie tylko znaczniki tworzą poprawną hierarchię, ale m.in. spełnione są następujące dodatkowe kryteria:

- korzeniem hierarchii jest element o nazwie `html`,
- element `html` zawiera podelementy `head` oraz `body`;
- element `head` zawiera podelement `title`;
- element `title` posiada zawartość tekstową.

Kontrola bieżąca	( )	komentarz odpowiedzi
XML oraz SGML nazywamy meta-językami, gdyż:		
Dostarczają rozwiązań dla zdefiniowania dowolnych nowych języków dziedzinowych	x	Tak. Definiują ogólną składnię umożliwiającą stosowanie znaczników a także dostarczają języka definicji schematu, pozwalającego zdefiniować reguły poprawności strukturalnej.
Każdy dokument musi mieć obowiązkowo wyspecyfikowany schemat, stanowiący jego metadane.		Nie. Taki wymóg dotyczył SGML. Dla XML zaś dostarczanie definicji schematu jest nieobowiązkowe.
Stosują znaczniki		Nie. Pojęcie meta-języka nie jest związane ze znacznikami.
Służą wyłącznie zapisowi metadanych		Nie. W dokumentach znacznikowych, zależnie od ich zastosowań, możemy gromadzić zarówno dane jak i metadane.

<<SCREENCAST S1 – wsparcie edycji HTML w edytorze i interpretacja drzewa dokumentu w przeglądarce>>

Kontrola bieżąca	( )	komentarz odpowiedzi
Załóżmy, że (posiadający definicję schematu) dokument XML służący przekazywaniu danych o zamówieniach zawiera element zamawiający, w którym na skutek błędu umieszczono atrybutu o nazwie <code>nazwisk</code> zamiast <code>nazwisko</code> . O poprawności tego dokumentu możemy powiedzieć, że:		
Pomimo tej usterki spełnia kryteria poprawności składniowej i strukturalnej		Nie. Wprawdzie uda się odtworzyć drzewo dokumentu, to obecność elementu o nazwie <code>nazwisk</code> jest niezgodna z opisaną przez schemat strukturą.
Dokument jest niepoprawny strukturalnie, a więc również niepoprawny składniowo		Nie. Brak poprawności strukturalnej nie implikuje niepoprawności składniowej.
Dokument jest niepoprawny strukturalnie, ale, o ile nie wkrały się inne błędy, może być poprawny składniowo	x	Tak. O ile nie złamano reguł poprawności składniowej, to mamy tu wyłącznie problem poprawności strukturalnej.

## Zadanie

**Sporządź poprawny składniowo i strukturalnie dokument HTML5.** Dokument powinien prezentować:

- Użycie znaczników nagłówków, paragrafów, tabeli, list wypunktowanych, numerowanych;
- Zawierać obraz (niech będzie on pochodził ze źródła dostępnego w Internecie i będzie wskazany poprzez adres bezwzględny URL).

**Ponadto, niech ww. dokument zawiera wstępną propozycję tematu projektu zaliczeniowego.** Powinna nim być prosta aplikacja WWW (a więc oprogramowanie dostępne jako zbiór dynamicznych stron WWW) o następujących cechach:

- Realizacja spójnego zestawu funkcjonalności związanej z działaniami jakiejś firmy, portalu WWW czy instytucji;
- Wykorzystanie trwałych danych znajdujących się w bazie danych po stronie serwera;
- Przetwarzanie danych przekazywanych przez użytkowników w ramach ich współdziałania ze stroną;
- Współdziałanie (wymiana danych) z inną witryną WWW (np. udostępnianie cenników detalistom, sprawdzanie kursów walut w zewnętrznej usłudze itp.).

*Dla zilustrowania powyższych wymagań, przyjmijmy popularny przykład – witryny sklepu internetowego. Gdyby stanowił on temat naszego projektu zaliczeniowego, to w zakresie funkcjonalności należałoby zadbać o:*

- *zaimplementowanie funkcjonalności dostępnej poprzez przeglądarkę – np. związanej z dokonywaniem zakupów: witryna nie może się ograniczać do samego prezentowania treści;*
- *oprogramowanie gromadzenia danych po stronie serwera i ich wykorzystania – tak, by zawartość odpowiednich struktur w bazie danych wpływała na treści prezentowane na stronach;*
- *przyjmowanie przez witrynę wprowadzanych przez użytkownika danych (tekst, liczby, daty, czy choćby tylko wybory prezentowanych elementów) oraz ich zapis w bazie danych (np. zapis złożonego zamówienia);*
- *współdziałanie z innymi aplikacjami dostępnymi poprzez WWW (np. udostępnianie interfejsu do pobierania aktualnej wersji cennika, albo też – konsumowanie podobnej funkcjonalności oferowanej przez inną witrynę).*

Powyższy szkic ma charakter wstępny. Rozumiemy, że część powyższych pojęć stanie się jaśniejsza w dalszej części kursu. Stąd też tematyka projektu zaliczeniowego będzie mogła zostać jeszcze zmodyfikowana czy zmieniona.

**Sporządź kopię ww. strony i przekształć ją do postaci zgodnej z XHTML 1.0 Strict.** Niech obie wersje strony poprawnie walidują się w walidatorze W3C (<http://validator.w3.org/>).