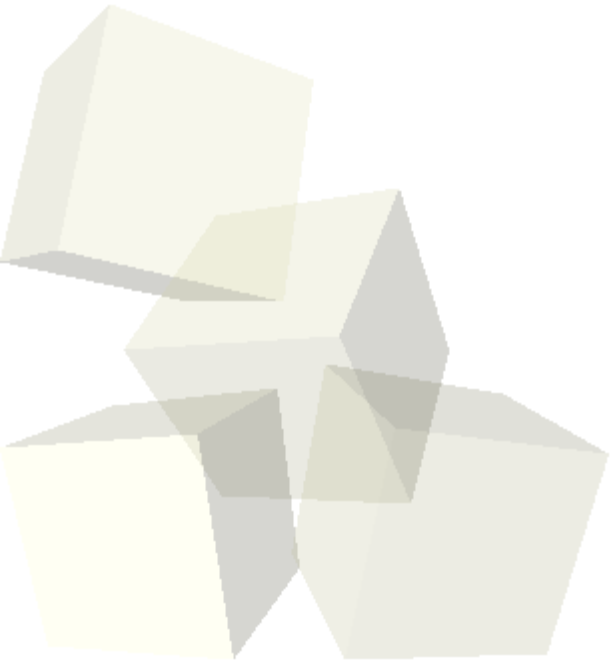




# XML





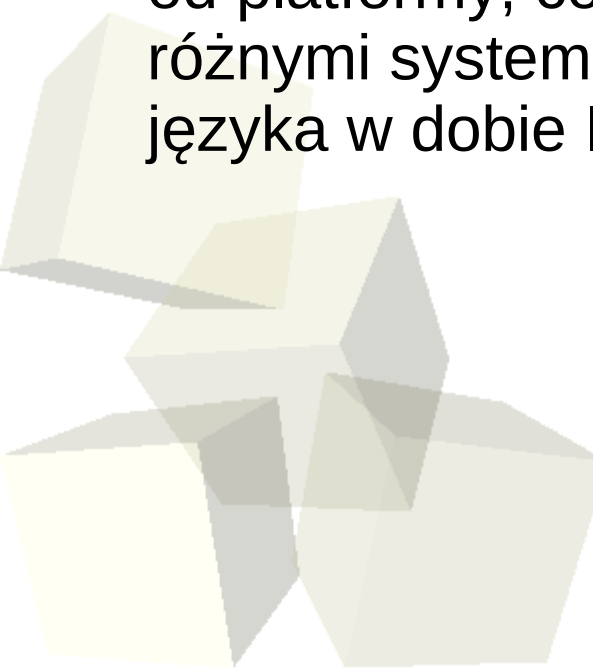
- **XML** – (*ang. Extensible Markup Language*) - Rozszerzalny język znaczników



Aktualna wersja – 1.0 (1.1 czeka na akceptację)

Należy do rodziny SGML.

XML to uniwersalny język formalny przeznaczony do reprezentowania różnych danych w ustrukturalizowany sposób. XML jest niezależny od platformy, co umożliwia łatwą wymianę dokumentów pomiędzy różnymi systemami i znacząco przyczyniło się do popularności tego języka w dobie Internetu.



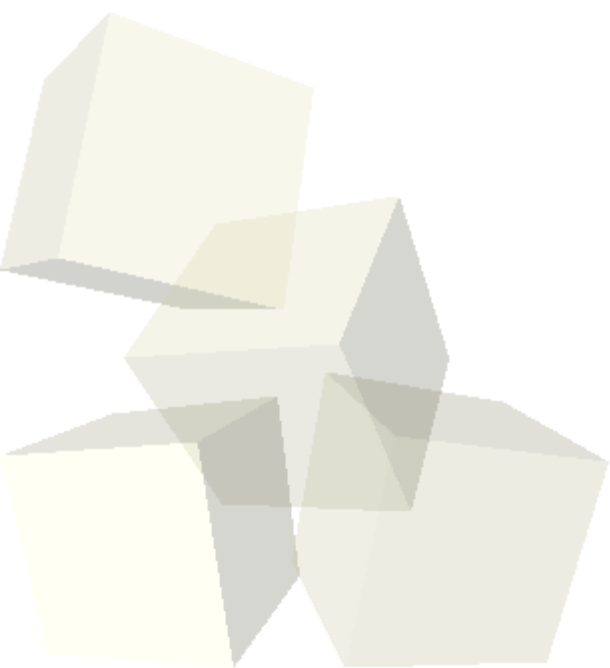


## XML w 10 punktach

(“XML in 10 points”, tłumaczenie Jacek Gleń,  
<http://www.w3.org/XML/1999/XML-in-10-points.pl.html>)

```
<?xml version="1.0"
<quiz>
  <question>
    Who was the forty-second
    president of the U.S.A.?
  </question>
  <answer>
    William Jefferson Clinton
  </answer>
  <!-- Note: We need to add
    more questions later.-->
</quiz>
```

**XML**





# Przykład dokumentu XML

Dokument zawierający opis podziału administracyjnego Polski.

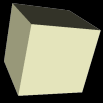
```
<?xml version="1.0" encoding="iso-8859-2"?>
<Polska>
  <województwo położenie="północny-wschód">podlaskie
    <stolica>Białystok</stolica>
    <obszar>20178km2</obszar>
    <ludność>1191000</ludność>
  </województwo>
  <województwo położenie="północ">warmińsko-mazurskie
    <stolica>Olsztyn</stolica>
    <obszar>24173km2</obszar>
    <ludność>1426000</ludność>
  </województwo>
</Polska>
```



## 1. XML służy do budowania struktur danych

Na ustrukturyzowane dane składają się formularze, książki adresowe, parametry konfiguracyjne, operacje finansowe, rysunki techniczne i wiele innych. XML to zestaw reguł (można go traktować również jako wytyczne czy konwencję) do projektowania formatów, na których oparte zostaną dane.

XML nie jest językiem programowania i nie trzeba być programistą, by z niego korzystać. XML sprawia, że generowanie danych i ich odczyt są znacznie łatwiejsze, zapewniając przejrzystą ich strukturę. Konstruując XML twórcy uniknęli typowych błędów zawartych w innych językach: jest on rozszerzalny, niezależny od platformy.



- Przykład: aplikacja przeliczająca waluty. Aktualne kursy walut pobierane z określonego serwisu.

Gdy dane są przechowywane w HTML-u, np. w postaci tabeli, i zmieni się struktura strony (drobne przegrupowanie, dodanie banera) albo też i tabeli, nasze narzędzie będzie musiało być przebudowane.

Zmiany w strukturze XML zwykle wymagają mniejszego nakładu pracy.

- W XML dane są rzeczywiście oddzielone od prezentacji (w przeciwieństwie do HTML-a).  
Format wyświetlania, podobnie jak w przypadku HTML-a, można zdefiniować za pomocą CSS.



```
<?xml version="1.0" encoding="iso-8859-2"?>
<?xml-stylesheet type="text/css" href="arkusz.css"?>

<dokument>
<czzerwony> ten tekst będzie wyświetlany jako czzerwony </czzerwony>
<tlusty> ten tekst będzie wyświetlany jako pogrubiony </tlusty>
</dokument>
```

```
dokument {
    font-family: "MS Sans Serif", sans-serif;
}

czzerwony {
    color: red;
}

tlusty {
    font-weight: bold;
}
```



# Poprawność dokumentu XML

## ■ Poprawność składniowa

- Wszystkie znaczniki powinny posiadać swoje domknięcia.
- Nie może być nakładających się elementów.
- Wartości atrybutów muszą być w apostrofach lub cudzysłowach.
- .....

## ■ Poprawność strukturalna

- Struktura dokumentu zgodna ze strukturą zdefiniowaną w definicji tego dokumentu.
- Obecne wymagane wszystkie atrybuty.

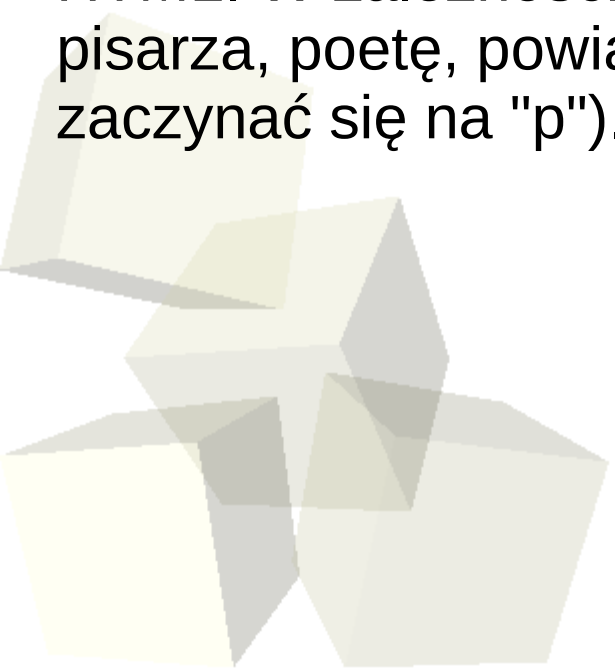
Zgodność z definicją dokumentu określoną za pomocą np. DTD, XML Schema, RELAX NG lub inną.





## 2. XML przypomina HTML

Podobnie jak HTML, XML korzysta z tagów (słowa ujęte w '<' i '>') oraz atrybutów (w postaci nazwa="wartość"). Podczas gdy HTML definiuje dokładne znaczenia każdego z tagów i jego atrybutów, jak również ich wygląd w przeglądarce, XML używa tagów tylko do rozgraniczenia pewnej części z całego dokumentu, a interpretację znaczenia pozostawia aplikacji odczytującej te dane. Inaczej mówiąc, "<p>" w dokumencie XML nie oznacza, że w tym miejscu znajduje się paragraf, jak to ma miejsce w HTML. W zależności od kontekstu taki tag może oznaczać parametr, pisarza, poetę, powiat, p... (i wcale nie jest powiedziane, że musi zaczynać się na "p").





- HTML jest językiem służącym do tworzenia dokumentów
- XML jest natomiast narzędziem do tworzenia języków (metajęzyk) służących do tworzenia dokumentów. Języki te nazywane są aplikacjami XML.
  
- Aplikacje XML – przykłady
  - XHTML – następca HTML-a
  - WAP – wyświetlanie stron w telefonach komórkowych
  - VML i SVG – grafika wektorowa
  - RSS – kanały informacyjne
  - XMLNews – wymiana aktualnych wiadomości
  - MathML – tworzenie dokumentów zawierających formuły matematyczne
  - MusicML – notacja muzyczna
  - VoiceML – notacja głosowa
  - OpenDocument – OASIS Open Document Format for Office Applications, dokumenty biurowe



- W HTML mamy ograniczony zestaw znaczników, a przeglądarka rozpoznaje je na podstawie wbudowanych w nią definicji typów dokumentów, czyli DTD (np. DTD dla HTML 4.01). Jeśli znacznika nie ma w definicji, po prostu go ignoruje (kto nie z nami, ten przeciw nam...). Przykładowo, znaczniki `<tytuł>` oraz `</tytuł>` będą zignorowane przez HTML ale nie przez XML.
- W XML **to TY** tworzysz własny zestaw znaczników, a przeglądarka posłusznie wyświetla je w żądany sposób - dba o to procesor XML. Jeśli życzysz sobie, aby twoja aplikacja miała znaczniki `<tytuł></tytuł>`, `<imię_psa></imię_psa>`, `<kolor_włosów></kolor_włosów>`, przeglądarka musi to uwzględnić.
- Dlaczego mówimy "rozszerzalny język znaczników"? Nie ma już ograniczenia do nic nie mówiących `<p>`, `<h1>`, `<ul>` - możemy wymyślić nazwy znaczników, które coś mówią czy coś wskazują.



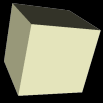
- XML to kilka modułów, które komunikują się ze sobą, podczas gdy SGML jest jednym dużym systemem
- Składnia XML-a została uproszczona w stosunku do SGML-a.
- Konieczność definiowania struktury w SGML; w XML jest to dowolne
- Wiele narzędzi wspomagających tworzenie XML





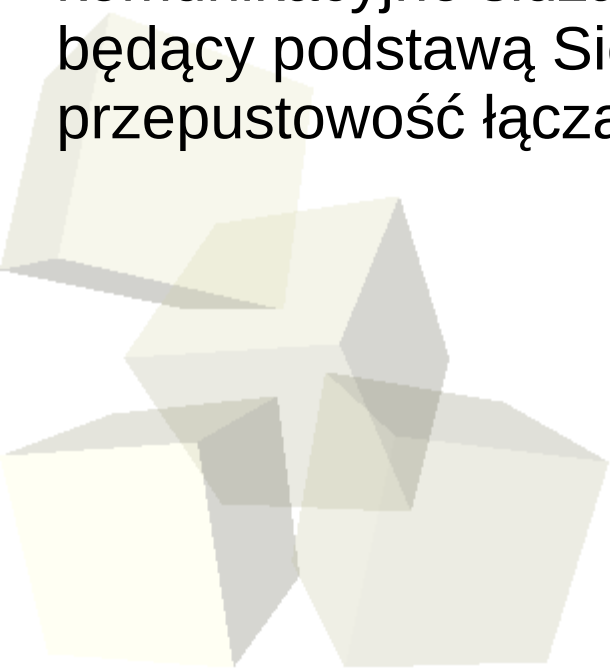
## 3. XML jest tekstem, ale nie należy go czytać

Często się zdarza, że programy korzystające z formularzy, książek adresowych i innych sformatowanych danych przechowują swoje dane na dysku, zapisując je zarówno w postaci binarnej, jak i tekstowej. Dużą zaletą postaci tekstowej jest to, że użytkownik, jeśli to tylko konieczne, może przeglądać dane bez potrzeby korzystania z dodatkowych aplikacji; ostatecznie można odczytać dane korzystając z edytora tekstu. Postać tekstowa pozwala również wykonawcy na łatwiejsze lokalizowanie i usuwanie błędów w aplikacjach. Podobnie jak HTML, pliki XML są tekstowe i choć nie powinny być przeglądane przez osoby postronne, mogą gdy zajdzie taka potrzeba. Jednak w przeciwieństwie do HTML, reguły dotyczące plików XML są ścisłe i nie naruszalne. Źle napisany tag, zgubiony nawias czy atrybut nie ujęty w cudzysłów czyni plik XML bezużytecznym, podczas gdy w HTML taka praktyka jest tolerowana, a często nawet jawnie dozwolona. Oficjalna specyfikacja języka XML zabrania aplikacjom domyślać się co ma znaczyć dany fragment uszkodzonego plik XML; jeśli w pliku jest błąd program powinien wstrzymać wykonywanie i zgłosić błąd.



## 4. XML jest nadmiarowy

Z powodu zastosowania postaci tekstowej i używania tagów do rozgraniczenia danych, pliki XML są znacząco większe niż pliki binarne z takimi samymi danymi. To świadoma decyzja projektantów języka XML. Korzyści płynące z postaci tekstowej są znaczące (patrz punkt 3), a wady takiego rozwiązania mogą być łatwo skompensowane. Miejsce na dysku twardym jest znacznie tańsze niż dawniej, a programy do kompresji takie jak zip czy gzip pakują pliki szybko i skutecznie. Dodatkowo protokoły komunikacyjne służące np. do transmisji modemowej czy HTTP/1.1, będący podstawą Sieci, kompresują pliki w locie, zachowując przepustowość łącza.





## 5. XML to rodzina technologii

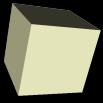
XML 1.0 jest specyfikacją, która definiuje czym są "tagi" i "atrybuty". Oprócz XML 1.0 na "rodzinę XML" składa się stale rosnący zestaw modułów, które oferują usługi przydatne do wypełnienia ważnych i częstych zadań. Xlink Opisuje standardowy sposób dodawania odnośników do plików XML, XPointer i XFragments to sposoby na wskazywanie na daną część dokumentu XML. XPointer przypomina trochę odnośnik URL, jednak zamiast wskazywać na dokument w sieci, wskazuje na dane zawarte w pliku XML. CSS, język opisu stylu strony, można stosować w odniesieniu do XML w taki sam sposób jak do HTML. XSL jest zaawansowanym językiem do określania stylu strony. bazuje na XSLT, języku transformacji, służącym do dodawania, usuwania oraz modyfikowania tagów i atrybutów. DOM jest zestawem standardowych funkcji służących do manipulowania plikami XML (i HTML) z poziomu języka programowania. XML Schemas 1 i 2 pomagają programistom precyzyjnie określać strukturę tworzonych formatów na bazie XML. Wiele innych modułów i narzędzi jest już dostępnych, lub znajdują się jeszcze w fazie produkcji.

## 6. XML to nowość, ale bez przesady

Początek XML to rok 1996, a do roku 1998 był tzw. rekomendacją W3C, co może sprawić wrażenie, że nie jest to najświeższa technologia. Rzeczywiście, technologia nie jest nowa. Zanim powstał XML, był jeszcze SGML stworzony już na początku lat 80-tych, w roku 1986 został standardem ISO, powszechnie używany do tworzenia dokumentacji dużych projektów. Rozwój HTML to rok 1990. Projektanci XML po prostu zostawili najlepsze części specyfikacji SGML i mając na uwadze doświadczenia z językiem HTML, stworzyli rzecz nie gorszą od oryginału, a znacznie bardziej prostszą i przejrzystsza.

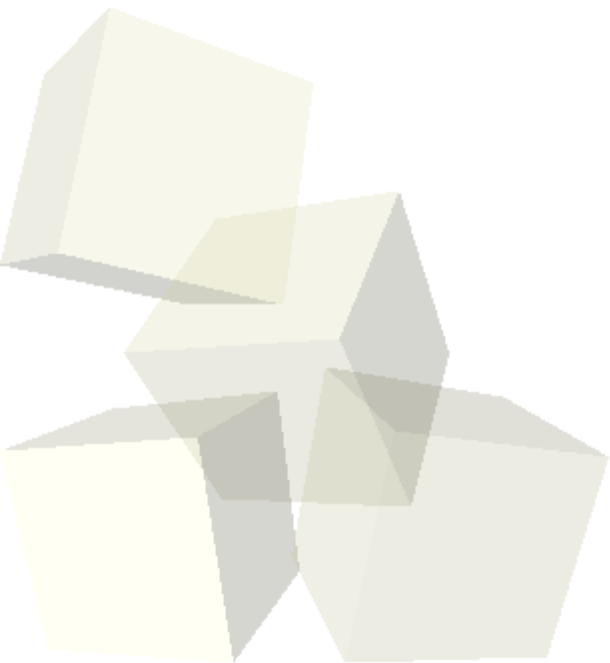
XML w pełni obsługuje Unicode, który z definicji powinien pomieścić wszystkie znaki występujące we wszystkich językach. Dlatego w jednym dokumencie XML można przechowywać dane zapisane w języku polskim jak i cyrylicą.





## 7. XML prowadzi HTML do XHTML

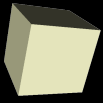
Jest jedna ważna aplikacja XML, to XHTML, następca HTML. XHTML ma wiele wspólnych elementów z HTML. Składnia została lekko zmieniona, by dostosować ją do wymogów XML. Dokument, który "bazuje na XML" dziedziczy składnię i zasady XML w pewien określony sposób (np. XHTML dopuszcza "<p>", ale nie "<r>"); nadaje też znaczenia ustalonym tagom (w XHTML "<p>" oznacza zawsze "paragraf", a nie "pisarza", "powiat" czy cokolwiek innego).



## 8. XML jest modularny

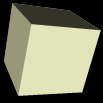
XML pozwala na zdefiniowanie nowego formatu poprzez łączenie lub korzystanie z innego. Ponieważ różne formaty są tworzone zupełnie niezależnie, mogą mieć tagi lub atrybuty o takiej samej nazwie, co powoduje dwuznaczność przy łączeniu takich dokumentów (np. w jednym "<p>" oznacza "paragraf", a w innym "pisarza"). W celu wyeliminowania takiej dwuznaczności w XML wprowadzona mechanizm przestrzeni nazw. Dobrymi przykładami dokumentów korzystających z przestrzeni nazw są XSL i RDF.

XML Schema został zaprojektowany, by odzwierciedlać to wsparcie dla modularności na poziomie definiowania struktury dokumentu XML, ułatwiając połączeniu dwóch schematów w celu stworzenia trzeciego, który obejmuje strukturę połączonych dokumentów.



## 9. XML jest podstawą RDF i Semantic Web - Znaczeniowej Sieci

Resource Description Framework (RDF - szkielet do opisu zasobów) to dokument XML, który wspomaga opis zasobów i metadanych aplikacji, takich jak lista piosenek, kolekcja fotografii i bibliografie. Na przykład RDF może pozwolić użytkownikowi na rozpoznanie osób na fotografii w internecie używając informacji z książki adresowej; następnie klient pocztowy wyśle informacje do tych osób, że ich zdjęcie pokazało się w sieci. Tak jak HTML łączy dokumenty, menu i formę aplikacji, co tworzy oryginalny, znany nam internet, tak RDF integruje szereg aplikacji w jedną Znaczeniową Sieć. Tak jak ludzie potrzebują uzgodnić znaczenie słów jakimi się posługują przy rozmowie, tak komputery potrzebują mechanizmu ustalającego znaczenie terminów używanych we wzajemnej komunikacji. Opisy formalne takich terminów w określonych obszarach (np. sprzedaż lub produkcja) nazwane są ontologiami i są niezbędnym elementem Znaczeniowej Sieci. RDF, ontologie i reprezentacja treści tak, aby komputery mogły lepiej wspomagać ludzi to tematy poruszane w Znaczeniowej Sieci.

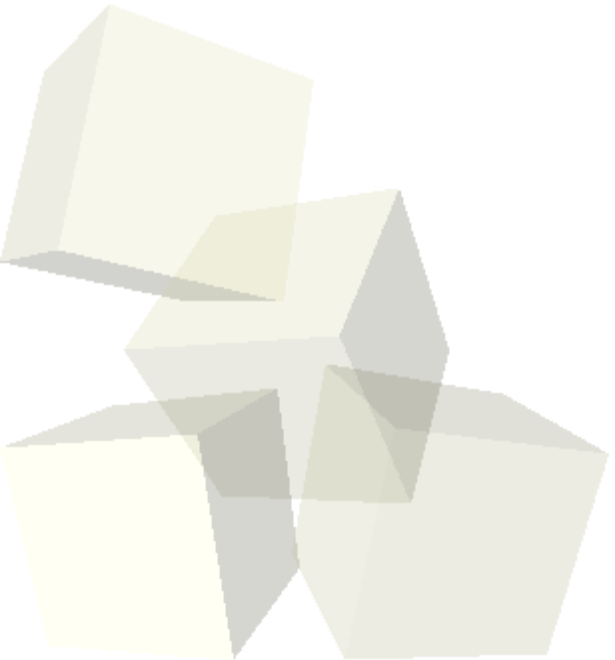


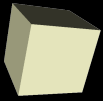
## **10. XML nie jest licencjonowany, jest niezależny od platformy i ma duże zaplecze**

Wybierając XML jako podstawę projektu, uzyskuje się dostęp do olbrzymiej i ciągle rosnącej bazy narzędzi (a być może jedno z nich już robi to czego potrzebujesz) oraz profesjonalistów doświadczonych w tej technologii. Wybór XML to coś jakby wybór SQL do baz danych; wciąż potrzeba zrobić własną bazę danych i aplikację z procedurami operującymi na niej, ale jest całe zaplecze narzędzi i ludzi mogących udzielić pomocy. Ponieważ XML jest wolny od opłat licencyjnych, tworzenie własnej aplikacji nie wymaga dodatkowych kosztów. Olbrzymie zaplecze, które wciąż się powiększa, oznacza, że producent oprogramowania nie jest już związany z jednym dostawcą rozwiązań. XML nie zawsze musi być najlepszym rozwiązaniem, ale zawsze trzeba je rozpatrzyć



# XHTML

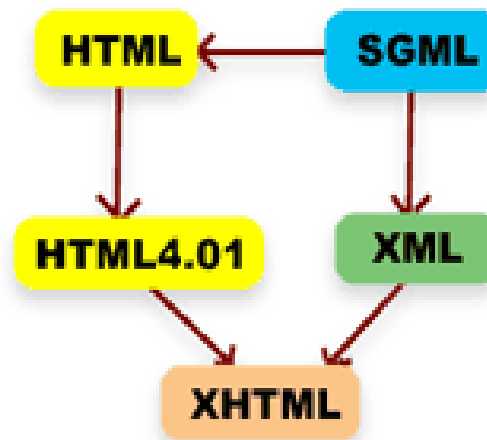




## XHTML (Extensible Hypertext Markup Language) – Rozszerzalny Hipertekstowy Język Oznaczenia

Jest przedstawieniem HTML w postaci XML.

Standard opracowany przez W3C.





- Jeśli strona XHTML zawiera błędy, nie może zostać wyświetlona
- Strony XHTML muszą mieć typ zawartości `application/xhtml+xml` (lub inny XML)

```
<meta http-equiv="content-type" content="application/xhtml+xml; charset=utf-8" />
```

- Dokument powinien rozpoczynać się od deklaracji XML (np. `<?xml version="1.0" encoding="utf-8"?>`); nie jest ona wymagana, gdy dokument ma kodowanie znaków UTF-8 lub UTF-16, albo gdy odpowiednie kodowanie zostało określone w nagłówkach HTTP (jednak nawet wtedy warto dołączać deklarację XML, gdyż pozwala ona ustalić kodowanie np. w przypadku zapisania strony na dysku).

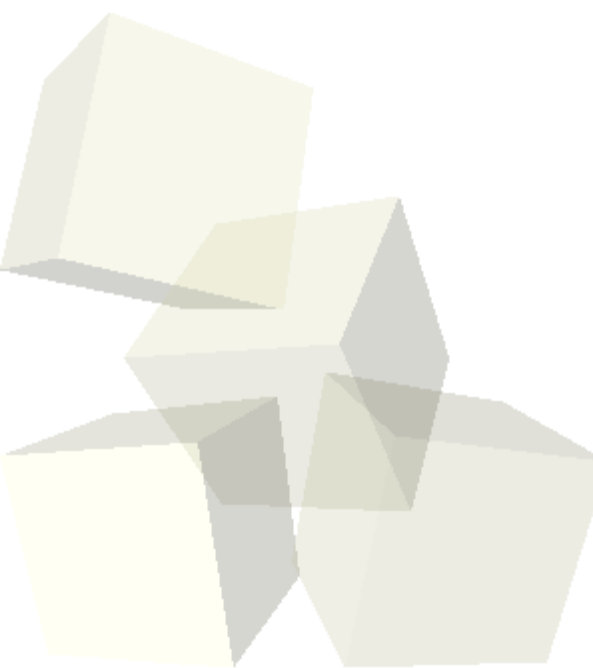
```
<?xml version="1.0" encoding="utf-8"?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```



- Elementem głównym jest <html>
- Element główny (html) musi zawierać atrybut xmlns określający przestrzeń nazw XHTML: <http://www.w3.org/1999/xhtml>

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" >  
      lub  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
```

- Dzięki użyciu XHTML można stosować inne aplikacje XML (jak MathML czy SVG) bezpośrednio w dokumencie, czego nie da się w HTML







**Przykład:** użycie XHTML w połączeniu z Rekomendacją MathML.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
  <head>
    <title>Przykład matematyczny</title>
  </head>
  <body>
    <p>Znaczniki języka MathML:</p>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply> <log/>
        <logbase>
          <cn> 3 </cn>
        </logbase>
        <ci> x </ci>
      </apply>
    </math>
  </body>
</html>
```



- Znacznikowi otwierającemu każdego niepustego elementu powinien odpowiadać znacznik zamykający (np. `<li> ... </li>`)
- Puste elementy muszą także być zamykane (np. zamiast `<br>` musi być `<br/>`, albo ewentualnie `<br></br>`)

```
<br> <hr>                <!-- ŹLE -->  
<br/><br /><hr></hr>      <!-- DOBRZE -->
```

- Elementy muszą być zagnieżdżane w odpowiedni sposób (np. zamiast `<p>Tekst z <em>wyróżnieniem</p></em>` – `<p>Tekst z <em>wyróżnieniem</em></p>`); wprowadzie w HTML także istniał taki wymóg, lecz nie był egzekwowany przez przeglądarki





- Nazwy elementów i atrybutów XHTML muszą być pisane małymi literami. Ta różnica jest konieczna, ponieważ XML rozróżnia wielkość liter, np. `<li>` i `<LI>` to dwa różne znaczniki.

```
<li> tekst </LI>          <!-- ŹLE -->
<li> tekst </li>         <!-- DOBRZE -->
```

- Wszystkie wartości atrybutów, muszą być ujęte w cudzysłów podwójny lub pojedynczy. Dotyczy to również wartości numerycznych.

```
<td rowspan=3>           <!-- ŹLE -->
<td rowspan="3">        <!-- DOBRZE -->
<td rowspan='3'>       <!-- DOBRZE -->
```

- Niedozwolona jest minimalizacja wartości atrybutów logicznych

```
<textarea readonly>     <!-- ŹLE -->
<textarea readonly="readonly"> <!-- DOBRZE -->
```