XML
eXtensible Markup Language

XML

Zpočátku programy pro přípravu tisku, specializované na tu kterou osvitovou jednotku (vzájemně nekompatibilní), tj. v textu řídící sekvence pro osvitovou jednotku.

Potom se začaly používat společné příkazy, které se překládaly pro konkrétní typ osvitové jednotky (nejrozšířenější troff a TeX).

Programy jako TeX se však hodí pouze pro zpracování dokumentů, které se mají ve výsledku tisknout. Hlavně kvůli tomu, že nabízejí příkazy, které umožňují měnit druh použitého písma, způsob zarovnání a nepřeberné množství dalších parametrů.

XML

S rozmachem Internetu a dalších médií (např. CD-ROM) vznikla potřeba jedny a tytéž informace prezentovat mnoha způsoby – kvalitním tiskem na papíře, jako hypertextovou příručkou na CD-ROMu apod. Pro tyto účely je však potřeba znát logickou strukturu dokumentu. Musíme vědět, že tohle je nadpis a tohle zase popis obrázku. Konkrétní velikost písma a způsob formátování záleží až na tom, zda chceme produkovat tištěnou knihu nebo multimediální CD-ROM.

XML

Potřebujeme tedy jazyk, který umožní označit *význam* jednotlivých částí textu, a ne jejich *vzhled*. Takovýmto „samopopisným jazykem“ je i XML. Nejde však zdaleka o první jazyk svého druhu. Jazykům, které umožňují vyznačovat části textu, se říká *značkovací jazyky* (markup language).

XML

- Asi prvním známým značkovacím jazykem byl GML (Generalized Markup Language, autoři Ch. Goldfarb, E. Mosher, R. Lorie ?)
- Princip GML se osvědčil a v 80. letech začala na základě GML vyvíjet standardizační organizace ANSI jazyk, který umožňoval definici vlastních značkovacích jazyků – uživatel si dle potřeb mohl vytvořit vlastní sadu značek vhodnou pro daný druh dokumentů. Tou dobou se sdružení GCA (Graphics Communications Association) snažilo vytvořit standardní formátovací jazyk GenCode použitelný na širokém spektru zařízení. Mnohé cíle obou projektů byly podobné, a proto se obě aktivity spojily.
- Výsledkem byl jazyk SGML (Standard Generalized Markup Language), který je definován v ISO normě 8879 z roku 1986.

XML

Jazyk SGML je skutečně hodně obecný – samozřejmě umožňoval definici vlastních značkovacích jazyků (sad značek a jejich vzájemných vztahů) pomocí tzv. definic typu dokumentu (DTD). Navíc měl spoustu volitelných parametrů – počínaje maximální délkou názvů značek a konče určením znaků použitelných jako oddělovače značek od textu. Komplexnost standardu SGML poněkud zbrzdila jeho praktické využívání. Velkou podporou pro SGML bylo americké ministerstvo obrany, které od svých dodavatelů vyžadovalo dokumentaci k výrobkům právě ve formátu SGML. Důvod byl zřejmý – bylo třeba, aby byla dokumentace použitelná v poměrně dlouhém období. Nešlo tedy použít nějaký proprietární formát textového procesoru, který se každých pár let mění.

XML

Asi nejznámější aplikací SGML je jazyk HTML (Hypertext Markup Language), který se používá pro tvorbu webových stránek. To jaké značky můžeme na stránkách používat určuje příslušné DTD, které je pro každou verzi HTML trošku jiné.

XML

V polovině 90. let došlo k paradoxní situaci. Jazyk HTML si získal velkou oblibu díky své jednoduchosti, která byla v ostrém kontrastu s komplexností SGML. Ukázalo se však, že pevně daná skupina značek, které HTML používá, už nestačí. Pro účely vyhledávání a vůbec efektivnější výměny dat by bylo lepší mít možnost používání vlastních značek, které by přesně vymezily význam textu. Požadavek by tedy mohl bez problémů splnit jazyk SGML.

XML

Standard SGML je dost komplexní a jeho úplná implementace velice náročná. Přitom se během deseti let používání SGML ukázalo, že se v praxi používá stejně jen část jeho možností. Tato nejdůležitější podmnožina SGML proto byla vybrána jako nový jazyk, který dovede Web do třetího tisíciletí. Správně již tušíte, že nový jazyk dostal jméno XML (eXtensible Markup Language). Jedná se o podmnožinu SGML, která si zachovává možnost definování vlastních DTD, a tedy značek, pro jednotlivé skupiny dokumentů.

XML

Na rozdíl od SGML je mnoho parametrů předem určeno a nelze je měnit – délka názvů značek, použité oddělovače a speciální znaky atd. XML už rovnou počítá s podporou všech možných jazyků, takže není tak úzce svázáno s angličtinou jako většina předchozích počítačových technologií. Navíc je syntaxe zápisu dokumentů v XML oproti SGML poměrně přísná, což umožní mnohem snazší a levnější vývoj aplikací, které umožňují s XML pracovat.

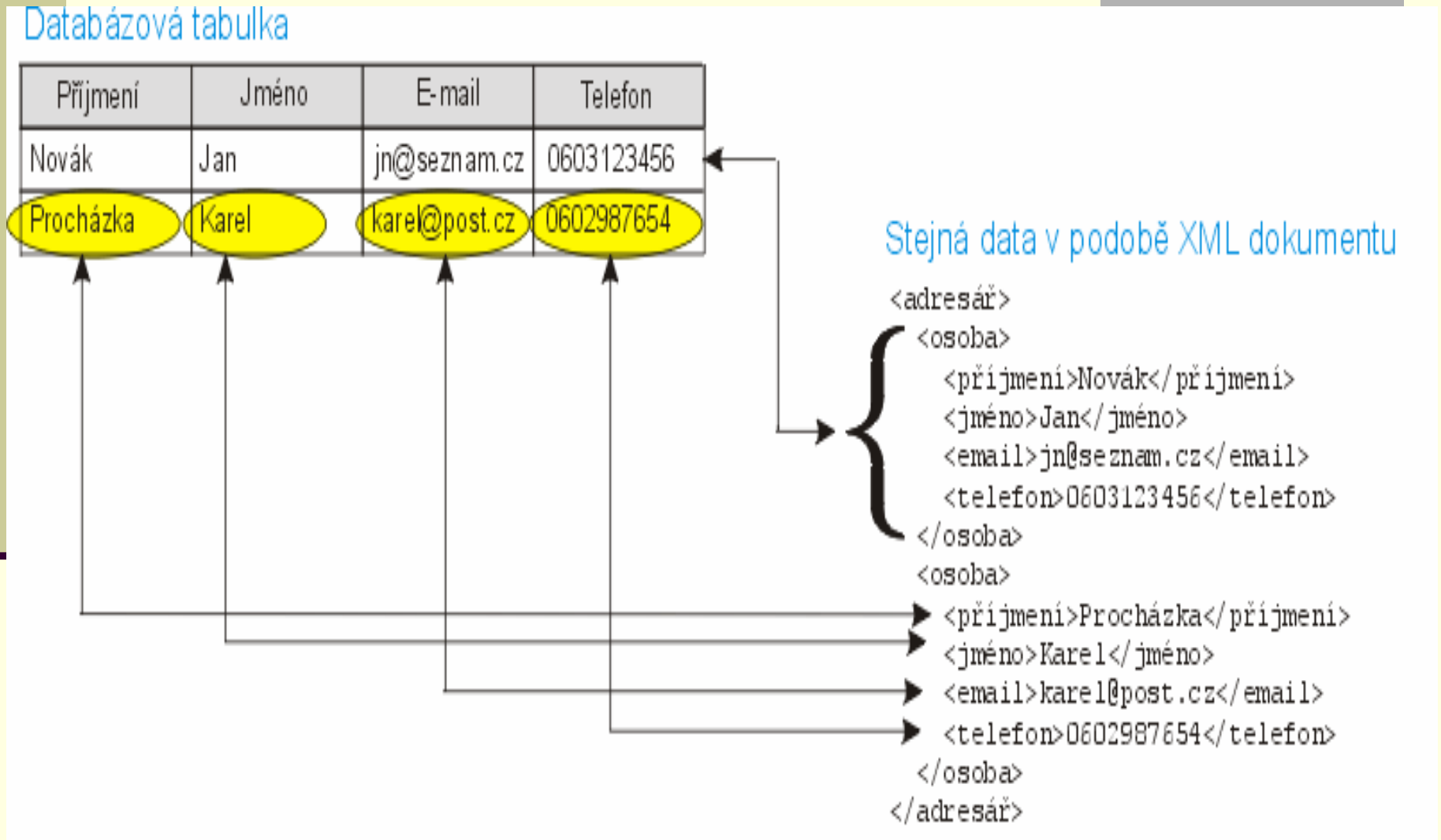
XML

XML pochází z oblasti, která se zaměřuje na uchovávání a zpracování textových dokumentů. Pro tyto účely se XML výborně hodí. Mnoho velkých i malých firem vyrábějících software, hardware nebo třeba letadla používá pro tvorbu dokumentace systémy založené na XML nebo SGML.

XML

Elektronické publikování dokumentů však není jedinou doménou XML. Značky umožňují v dokumentu zachytit důležité informace o struktuře a významu. Není proto problém do XML dokumentu uložit například obsah tabulky z relační databáze.
O dokumentech bychom měli spíše uvažovat jako o nosičích informací – není už tak důležité, jak moc strukturovaná jsou data v nich. Některé aplikace pracují s dokumentem, který je filosofickou esejí, jiné za dokument považují řadu čísel s burzovními indexy.

XML


Databázová tabulka
Příjmení
Jméno
E-mail
Telefon
Novák
Jan
jn@seznam.cz
0603123456
Procházka
Karel
karel@post.cz
0602987654
Stejná data v podobě XML dokumentu
<adresář>
<osoba>
<příjmení>Novák</příjmení>
<jméno>Jan</jméno>
<email>jn@seznam.cz</email>
<telefon>0603123456</telefon>
</osoba>
<osoba>
<příjmení>Procházka</příjmení>
<jméno>Karel</jméno>
<email>karel@post.cz</email>
<telefon>0602987654</telefon>
</osoba>
</adresář>

XML

- **Co přináší XML nového**
  - **Standardní formát pro výměnu a sdílení informací**
  - **Mezinárodní podpora**
  - **Vysoký informační obsah**
  - **Snadná konverze do dalších formátů**
  - **Automatická kontrola struktury dokumentu**

XML

Co přináší XML nového?

- **Standardní formát pro výměnu a sdílení informací (j***e potřeba používat nějaký jednoduchý otevřený formát, který není úzce svázán s nějakou platformou nebo proprietární technologií).*
  - *Otevřený formát je to proto, že jeho specifikace je každému zdarma k dispozici na serveru konsorcia W3C, které se stará i mnoho dalších technologií souvisejících s Webem. Každý tak může bez problémů do svých aplikací implementovat podporu XML. To je velký rozdíl oproti firemním formátům, k nimž není k dispozici žádná dokumentace a navíc se jedná v porovnání s XML o velice složité, často binární, formáty.*

XML

Mezinárodní podpora

- XML používá znakovou sadu ISO 10646. Pod tím si asi nepředstavíte nic konkrétního. ISO 10646 je 32bitová znaková sada, která dokáže pojmout všechny dnes používané znaky všech jazyků.
- V XML proto můžeme vytvářet dokumenty, které obsahují texty v mnoha jazycích najednou – můžeme míchat např. češtinu, angličtinu, ruštinu, arabštinu a korejštinu zcela dle libosti. Pokud by dokumenty obsahovaly pouze český text, bylo by ukládání přímo v ISO 10646 zbytečné plýtvání místem. XML dokument proto může být v libovolném kódování (např. windows-1250, iso-8859-2). Kódování je však v každém dokumentu přesně určeno, takže odpadají problémy s konverzí z jednoho kódování do druhého. Je hned jasné, v jakém kódování dokument je.

XML

Vysoký informační obsah

- Pomocí XML značek vyznačujeme v dokumentu význam jednotlivých částí textu. Říkáme "toto je název výrobku, tohle zase telefonní číslo a tohle je číslo našeho účtu". Dokumenty obsahují mnohem více informací, než kdyby se používalo prezentační značkovaní – tohle je tučným písmem Arial o velikosti 12 bodů zarovnané vlevo.

XML

Snadná konverze do dalších formátů

- V mnoha případech potřebujeme XML dokument zobrazit na nějakém běžném médiu – na obrazovce, na papíře. V tomto případě už samozřejmě chceme přesně ovlivnit, jak se obsah jednotlivých značek zobrazí. XML samo o sobě žádné takové prostředky nenabízí. Existuje však několik *stylových jazyků*, které umožňují definovat, jak se mají jednotlivé elementy zobrazit. Souboru pravidel nebo příkazů, které definují jak se dokument převede do jiného formátu se říká styl.

XML

Stylových jazyků existuje dnes několik.

- Mezi nejznámější patří asi **kaskádové styly (CSS).** Ty lze použít pouze pro jednoduché formátování, které dobře poslouží pro zobrazení dokumentu na obrazovce v XML editoru nebo v prohlížeči.
- Pro náročnější aplikace slouží jazyk **XSL (eXtensible Stylesheet Language).** Ten umožňuje před samotným zformátováním dokument různě upravovat a transformovat (části dokumentu třeba vypustit nebo naopak automaticky vygenerovat obsah dokumentu). Dvě části: **XSLT** (transformace), **XSLFO** (formátovací objekty)
- Společně s XML lze použít i velice výkonný, i když pro některé aplikace příliš složitý, jazyk **DSSSL (Document Style Semantics and Specification Language),** který byl původně vyvinut pro potřeby jazyka SGML.

XML

Automatická kontrola struktury dokumentu

- XML nám umožňuje definovat si vlastní sadu značek, které chceme v dokumentu používat. Tuto možnost samozřejmě využít nemusíme – můžeme používat jaké značky chceme. Pokud si však předem pomocí DTD definujeme, jaké značky může dokument obsahovat, máme další život mnohem lehčí. Zcela automaticky můžeme kontrolovat, zda dokument obsahuje pouze povolené značky.
- Programu, který kontroluje správnost XML dokumentů, se říká *parser*. To má velký význam i při vývoji aplikací. Pro čtení dat můžeme použít parser, který za nás detekuje většinu chyb v datech – obrovsky nám to ušetří práci.

XML

- K čemu můžeme XML použít již dnes
  - B2B – business-to-business aplikace
  - „Inteligentní“ webové stránky
  - Elektronické publikování
  - Univerzální datový formát

XML

B2B – business-to-business aplikace

- Výměnu informací mezi obchodními partnery v elektronickém formátu. Jako vhodný formát pro přenos dat se jeví právě jazyk XML, který je velice jednoduchý a podporovaný na mnoha počítačových platformách. Pomocí XML si firmy mohou vyměňovat objednávky, faktury a mnoho dalších údajů.
- To bylo možné již dříve díky EDI (Electronic Data Interchange). Datové formáty používané v EDI byly však dost složité a jejich implementace byla poměrně nákladná. Navíc jednotlivé EDI systémy nebyly mezi sebou kompatibilní, a tak bylo často potřeba informační systém speciálně upravit pro každého dalšího obchodního partnera, se kterým jsme chtěli komunikovat elektronicky.

XML

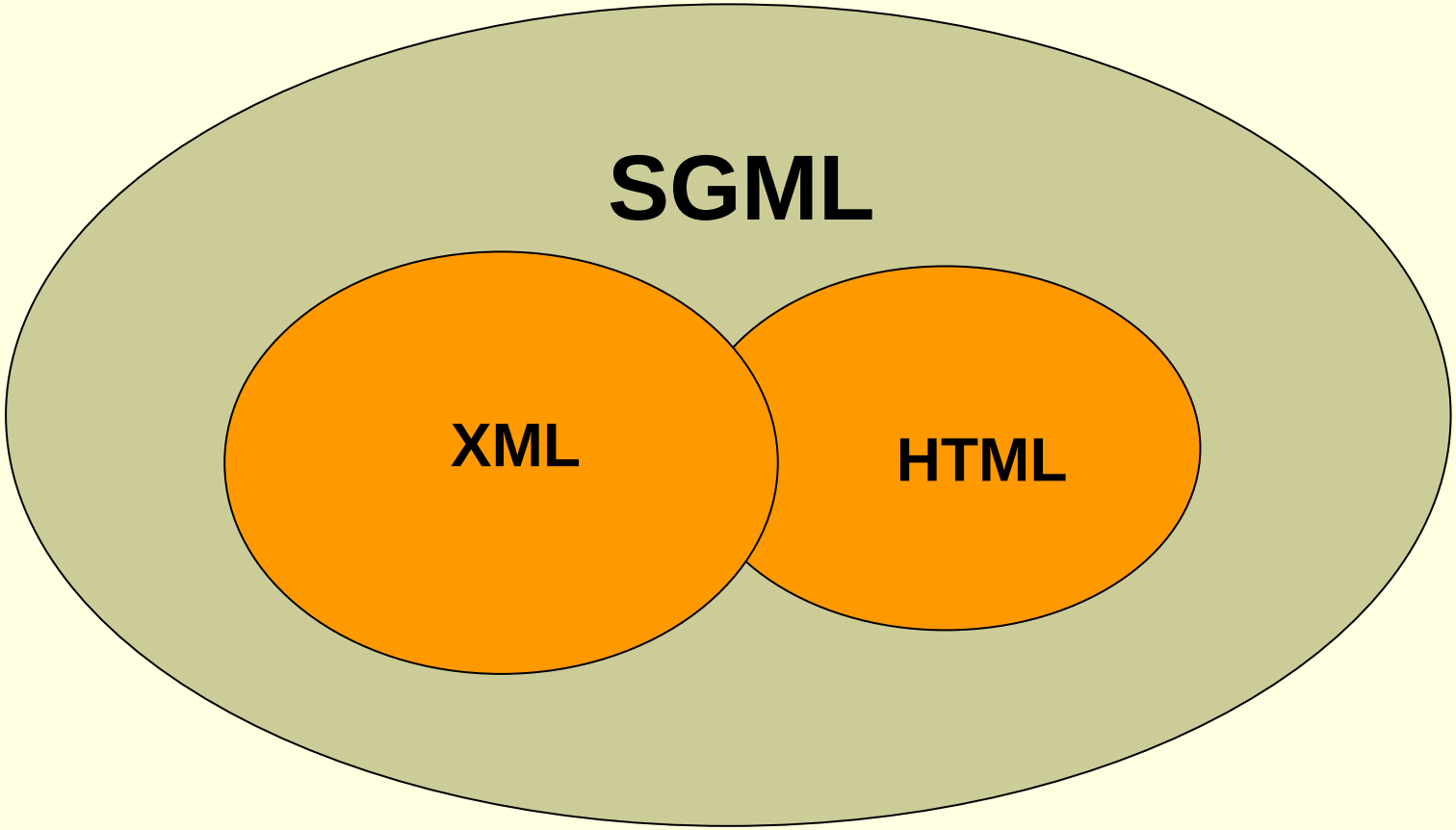
Inteligentní“ webové stránky

- Nasazení na Webu. Možnost definice vlastních značek, které přesně vyznačí význam jednotlivých částí stránky bude mít pozitivní efekt na přesnost a rychlost vyhledávání informací.
- V zásadě lze XML pro tvorbu stránek využít dvěma způsoby. První, více revoluční přístup, počítá s tím, že stránky budou používat zcela vlastní sady značek. Pro mnoho aplikací je však mnohem jednodušší používat již zažité HTML značky a pouze je vhodně doplnit o pár dalších, kterými se označí části stránky důležité pro vyhledávání. XHTML jazyk

XML

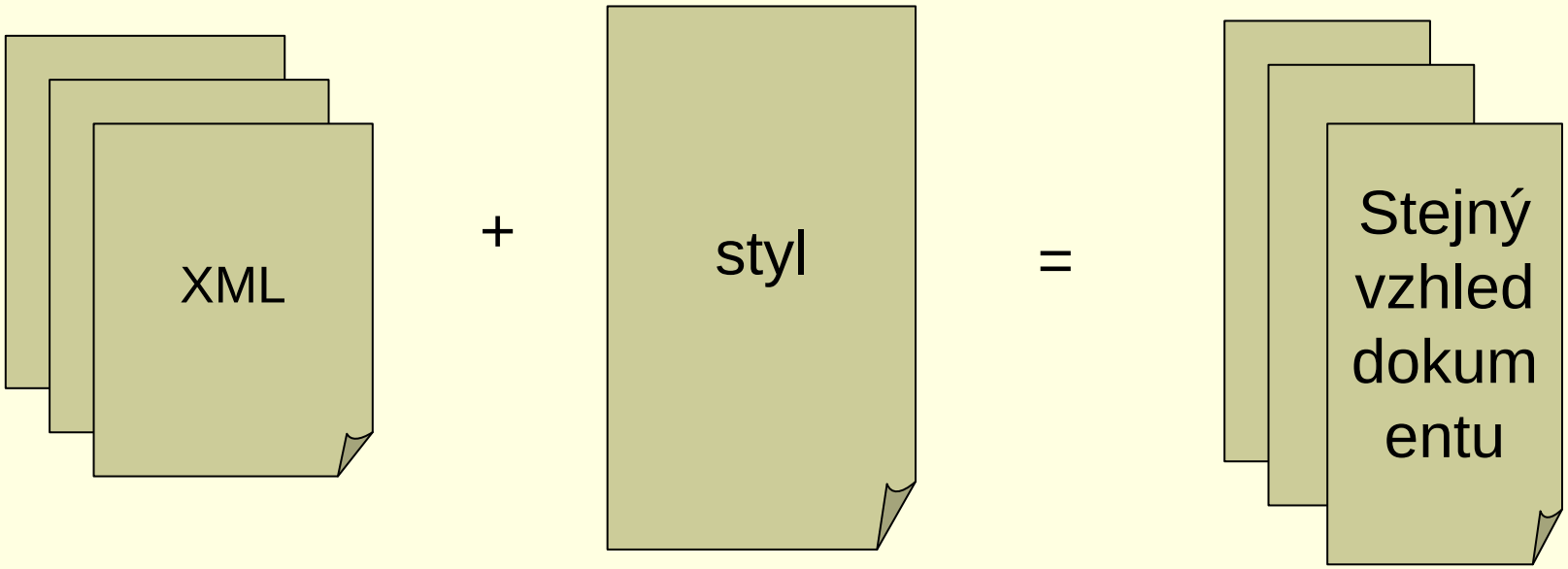
- **Elektronické publikování**
- Papír však dnes není jediné cílové médium. Často potřebujeme jeden dokument v několika různých formátech – jako tištěnou knihu, sadu provázaných webových stránek nebo hypertextovou příručku na CD-ROMu. Stojíme před novým problémem – už nestačí pohodlně vytvořit text a ten rozmnožit v libovolném počtu výtisků. My navíc potřebujme jeden text publikovat v několika naprosto odlišných formátech.

XML



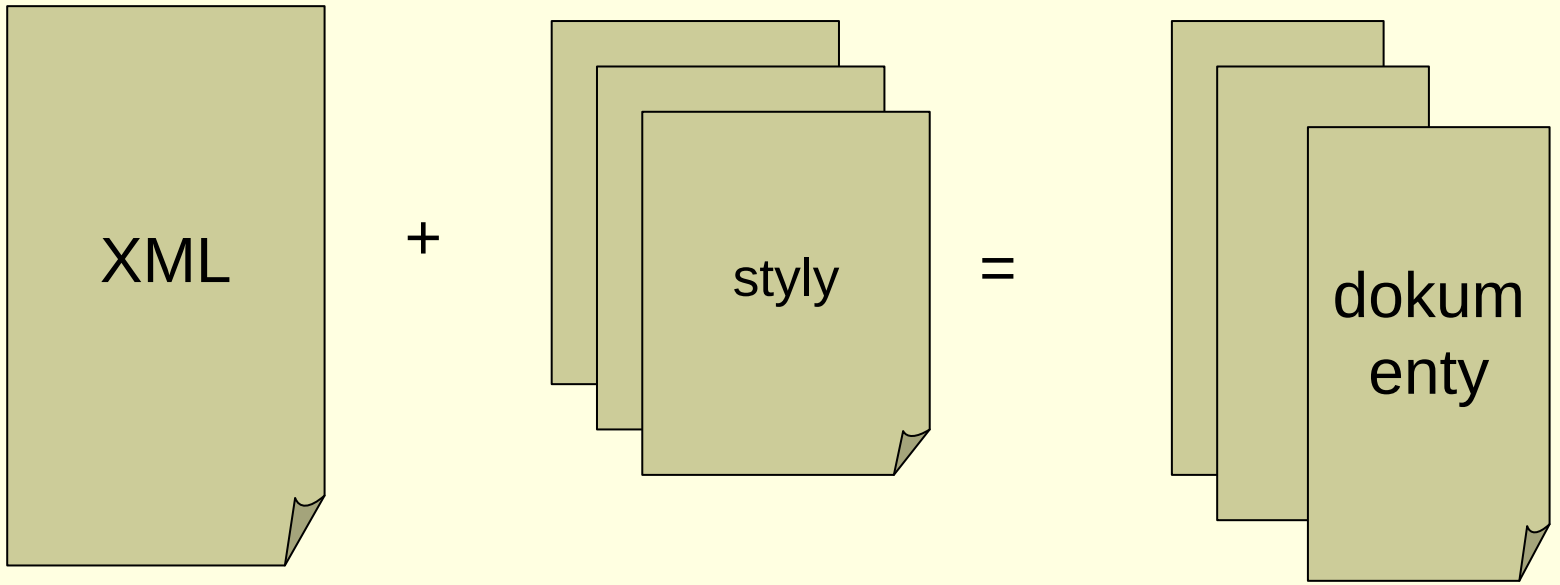
XML

Různé xml (odlišná data, ale stejné značky) a jeden styl (definice vzhledu) vytvoří dokumenty podobného typu



XML

Různé styly vytvoří ze stejného xml různé dokumenty



XML

- Element – základ struktury XML dokumentu
- Obsah elementu:
  - Počáteční značka (tag)
  - Obsah elementu
  - Koncová značka = lomítko + počáteční značka

Značky je zapsaná v lomených závorkách

Př. <ukazka> obsah ukazky </ukazka>

Prázdný element <ukazka> </ukazka> = <ukazka/>

XML

- Názvy značek
  - Musí začínat písmenem nebo podtržítkem
  - Dále písmena, číslice, podtržítka, pomlčky a tečky
  - Jiné znaky nejsou povoleny
  - Xml je case senzitive
    - <ukazka>nějaký text </Ukazka> je chybně
- Elementy se nesmí křížit
- U každého elementu (počáteční značky) se může vyskytnout atribut, je oddělen mezerou od názvu značky, pak rovnítko, je uzavřen v apostrofech nebo uvozovkách

<ukazka cislo="1"> nějaký text </ukazka>

XML

- Zakázané znaky – místo nich vestavěné znakové entity
- Znak < nahrazuje entita &lt
- Znak > nahrazuje entita &gt
- Znak & nahrazuje entita &amp
- Znak “ nahrazuje entita &quot
- Znak ´ nahrazuje entita &apos

XML

- Poznámka – komentář
- <!-- text poznámky -->

XML

Příklad

- <student>
- <jmeno> Karel Novak </jmeno>
- <adresa>
- <ulice> Kamenicky 1</ulice>
- <mesto> Brno </mesto>
- </adresa>
- <kontakt>
- <telefon> 444 555 666 </telefon>
- <e-mail> novakkarel@seznam.cz </e-mail>
- </kontakt>
- </student>

XML

Příklad s chybami

- <student>
- <jmeno> Karel Nová**á**k </jmeno>
- <adresa>
- <ulice> Kameni**č**ky 1</ulice>
- <mesto> Brno </mesto>
- </adresa>
- <kontakt>
- <telefon> 444 555 666 </**T**elefon>
- <e-mail> novakkarel@seznam.cz </e-m**e**il>
- </kontakt>
- </student>

XML

- Proč chyby?
- Nov**á**k, Kameni**č**ky – je použitá diakritika, ale není sděleno, jaké kódování se bude používat, implicitní kódování nezná češtinu
- /**T**elefon – v koncové značce je použité velké písmeno T, v začátečním je malé t, jde o dvě různé značky (XML je case senzitive)
- /e-m**e**il – rozdíl od začáteční značky e-mail

XML

- U některých elementů na obrazovce (odkazy v následujícím snímku) je znak – (znak mínus)
  - Jde o ty elementy, které obsahují ještě jiné elementy, pomocí mínus je mohu skrýt nebo zobrazit pomocí plus

XML

- Ukázky
  - Ukázka první a třetí jsou ukázky správně syntakticky napsaného dokumentu (well formed document), ve třetí ukázce je přidán řádek s verzí XML a použitým kódováním
  - UKAZKAXML.xml
  - UKAZKAXMLchyby.xml
  - UKAZKAXMLkodovani.xml

XML

- Pokud dokument odpovídá syntaktickým pravidlům, říkáme že se jedná o správně strukturovaný dokument, well-formed document.

XML

- Ale je ještě sada pravidel pro tvorbu určitého typu dokumentu, soubor podmínek. Jde o Definici Typu Dokumentu (Document Type Definition) neboli DTD. Jde o důležitou vlastnost, zděděnou od SGML.Umožní nám definovat vlastní sadu značek.
- Pokud dokument odpovídá všem pravidlům v určitém DTD, říkáme že je validní.

XML

- DTD
  - Element
    - Jméno
    - Obsah
  - Atribut

XML

- Element
- Obecný tvar
- <! ELEMENT název_elementu určení obsahu >
  - Každá deklarace začíná <!
  - Následuje název druhu prvku který definujeme, zapsaný vždy velkými písmeny ELEMENT
  - Následuje název elementu
  - Poslední je určení obsahu elementu a znak >
  - Př. <! ELEMENT student>

XML

- Obsah elementu
- Je určen modelovou skupinou
  - Skládá se ze závorek a minimálně jednoho jmenného symbolu
- Druhy obsahu elementu
  - Data (text) tvar (#PCDATA), # určuje že se nejedná o název elementu
  - Př. <! ELEMENT student (#PCDATA) >

XML

- Druhy obsahu elementu
  - Prvek, který obsahuje další elementy
    - Element aaa obsahuje právě jeden element bbb
    - <! ELEMENT aaa (bbb) >
    - Element aaa obsahuje vždy jeden element bbb a ccc v daném pořadí
    - <! ELEMENT aaa (bbb, ccc) >
    - Element aaa obsahuje buď jeden bbb nebo jeden ccc
    - <! ELEMENT aaa (bbb|ccc)>

XML

- Druhy obsahu elementu
  - Předchozí deklarace byly vždy pouze pro jeden element bbb nebo ccc
  - Více elementů bbb (může se objevit jednou i vícekrát, tj. minimálně jednou)
  - <! ELEMENT aaa (bbb+) >
  - Element bbb buď jednou nebo vůbec ne
  - <! ELEMENT aaa (bbb?) >
  - Element bbb se může vyskytnou v libovolném počtu
  - <! ELEMENT aaa (bbb*) >

XML

Druhy obsahu elementu – identifikátory kvantity

,	Přesně určené pořadí
|	Nahrazuje slovo nebo
+	Výskyt alespoň jednou (jednou a vícekrát)
?	Výskyt jednou nebo vůbec ne (0 nebo jedenkrát)
*	Libovolný počet opakování (0 a vícekrát)

XML

- Druhy obsahu elementu
  - Identifikátory kvantity se mohou použít na celou modelovou skupinu
    - Kombinace elementů bbb, ccc se musí objevit alespoň jednou
    - <! ELEMENT aaa (bbb, ccc)+ >
    - Musí být aaa, dále musí být bbb nebo ccc
    - <! ELEMENT (aaa, (bbb|ccc)) >

XML

Mimo deklarace obsahu může DTD obsahovat i deklaraci atributů

Tvar:

<! ATTLIST aaa ccc CDATA>

V elementu **aaa** může jeho atribut **ccc** obsahovat skoro libovolná data (ne např. < &), CDATA je Character Data

< aaa ccc=“50.8”) > text </aaa>

XML

- Atributy

<! ATTLIST aaa ddd NMTOKEN >

Tato varianta stanovuje, že hodnota atributu **ddd** je složena z písmen, číslic, teček, pomlček, podtržítek a dvojteček, tj. vyhovuje syntaxi tvorby jmen xml elementů (ale neplatí omezení pro první znak písmeno).

Pokud chceme i mezery mezi znaky, tak je atribut NMTOKENS.

< aaa ddd=“4VB”) > nějaký text </aaa>

< aaa ddd=“A B Ceda”) > nějaký text </aaa>

XML

- Další typy atributů
  - ID, IDREF, IDREFS
  - Výčet možných hodnot, případně implicitní hodnota (řetězec v uvozovkách za výčtem)
    - Atribut bbb může nabývat hodnot 1,2,3,4, pokud neuvedeme, tak má hodnotu 1
    - <! ATTLIST aaa bbb (1|2|3|4) “1” >
    - Příklad:
    - < aaa bbb=“3”) > nějaký text </aaa>

XML

- Další typy atributů
- Můžeme ještě použít slovo #REQUIRED (atribut je povinný) nebo #IMPLIED (atribut je volitelný) nebo #FIXED před implicitní hodnotou (určená hodnota je jediná, kterou může atribut nabýt).

<! ATTLIST aaa bbb CDATA #REQUIRED
ccc CDATA #IMPLIED
ddd CDATA #FIXED “ABC” >

< aaa bbb=“nic” ddd=“ABC” > text </aaa>

XML

Následující ukázka je převzatá z existujícího systému na Mendelově zemědělské a lesnické univerzitě v Brně. Jde o část DTD tabulky pro vytváření studijních podkladů.

XML

```
<!ELEMENT kapitola     (nazev_kapitoly, cil_kapitoly, kap_klicova_slova?, cas_studia?,
    uvod_kapitoly?,
                       sekce+, shrnuti, seznam_pojmu?, seznam_otazek?,  vzor_priklady?,
                       priklady_s_klicem?, cvic_priklady?, test?, zdroje?)>
<!ATTLIST kapitola
    %spol_atr;
>

<!ELEMENT nazev_kapitoly    (#PCDATA | %styl_prvky; | pojem | odkaz | poznamka)*>
<!ATTLIST nazev_kapitoly
    %spol_atr;
>

<!ELEMENT cil_kapitoly      (%blok_prvky;)+>
<!ATTLIST cil_kapitoly
    %spol_atr;
>

<!ELEMENT cas_studia      EMPTY >
<!ATTLIST cas_studia
    %spol_atr;
    minut NMTOKEN #REQUIRED
```

XML

```
<!ELEMENT odstavec     (#PCDATA | %popis_prvky; | %styl_prvky; )*>
<!ATTLIST odstavec
              odsazeni     (ano | ne) "ne"
              zarovnani    (vpravo | vlevo | stred | blok) "blok"
         %spol_atr;
>

<!ELEMENT seznam       (polozka)+>
<!ATTLIST seznam
          typ        (1 | I | i | a | A | circle | square | disc) "1"
          %spol_atr;>

            <!ELEMENT polozka      (#PCDATA | %styl_prvky; | %popis_prvky; |
%blok_prvky;)*>
            <!ATTLIST polozka
                        %spol_atr;
            >
```

XML

- Jak připojím DTD k dokumentu XML
  - Zapíšeme na začátek xml dokumentu
    - Viz další snímek
  - Nebo uložíme DTD jako samostatný soubor a na začátku xml dokumentu je cesta k DTD.
    - <!DOCTYPE studenti SYSTEM "XMLKompletnibezDTD.dtd">
    - Viz další snímek
  - Pokračování prezentace po ukázkách

XML

```
<?xml version="1.0" encoding="windows-1250" ?>
<!DOCTYPE studenti [
<!ELEMENT studenti (student+)>
<!ELEMENT student (jmeno, adresa, kontakt)>
<!ELEMENT jmeno (#PCDATA)>
<!ELEMENT adresa (ulice, mesto)?>
<!ELEMENT kontakt (telefon, e-mail)?>
<!ELEMENT ulice (#PCDATA)>
<!ELEMENT mesto (#PCDATA)>
<!ELEMENT telefon (#PCDATA)>
<!ELEMENT e-mail (#PCDATA)>
]>
<studenti>
<student>
        <jmeno> Karel Novák </jmeno>
        <adresa>
                <ulice> Kameničky 1</ulice>
                <mesto> Brno </mesto>
```

Zpět na poslední zobrazený snímek 

XML

```
<?xml version="1.0" encoding="windows-1250" ?>
<!DOCTYPE studenti SYSTEM "XMLKompletnibezDTD.dtd">
<studenti>
<student>
        <jmeno> Karel Novák </jmeno>
        <adresa>
                <ulice> Kameničky 1</ulice>
                <mesto> Brno </mesto>
        </adresa>
        <kontakt>
                <telefon> 444 555 666 </telefon>
                <e-mail> novakkarel@seznam.cz </e-mail>
        </kontakt>
</student>
```

XML

- Soubor XMLKompletníbezDTD.dtd je uložen ve stejném adresáři jako xml soubor, ve kterém je DTD používáno (relativní adresování)

XML

- **Zajímavé odkazy**
- http://www.xml.com
  - Zpravodajský server věnovaný XML. Komentáře k nově vyvíjeným standardům, recenze programů atd.
- http://www.biztalk.org
  - Server firmy Microsoft zamýšlený jako repozitory schémat využívaných v různých oblastech.
- http://www.xml.org
  - Další server, který se chce stát repozitory DTD, schémat atd.
- http://www.oasis-open.org/cover/
  - Jeden z nejlepších informačních zdrojů o XML a SGML.
- http://www.zvon.org
  - Výukové materiály o XML, XSL a dalších.
- http://www.xmlsoftware.com/
  - Velice obsáhlý přehled programů, které umožňují pracovat s XML. V přehledu jsou uvedeny komerční i „free“ produkty.
- http://www.w3.org/XML/
  - Stránky konsorcia W3C věnované jazyku XML obsahují zajímavé odkazy a samozřejmě specifikaci XML a dalších souvisejících jazyků.

XML

- Literatura
  - Kosek, J.: XML pro každého. Grada Publishing, Praha 2000. ISBN 80-7169-860-1
  - Grusová, J.: XML pro úplné začátečníky. Computer Press, Praha 2002. ISBN 80-7226-697-7
  - www.w3.org