

# PHP a XML

Jiří Kosek

EDICE  
**PROFESIONAL**

GRADA

- Využití XML na webu a podpora XML v PHP5
- Čtení dokumentů pomocí SimpleXML, DOM, SAX a XMLReader
- XML schémata, XPath a XSLT
- Webové služby (SOAP, REST) a AJAX

## Upozornění pro čtenáře a uživatele této knihy

Všechna práva vyhrazena. Žádná část této tištěné či elektronické knihy nesmí být reprodukována a šířena v papírové, elektronické či jiné podobě bez předchozího písemného souhlasu nakladatele. Neoprávněné užití této knihy bude **trestně stíháno**.

*Používání elektronické verze knihy je umožněno jen osobě, která ji legálně nabyla a jen pro její osobní a vnitřní potřeby v rozsahu stanoveném autorským zákonem. Elektronická kniha je datový soubor, který lze užívat pouze v takové formě, v jaké jej lze stáhnout s portálu. Jakékoliv neoprávněné užití elektronické knihy nebo její části, spočívající např. v kopírování, úpravách, prodeji, pronajímání, půjčování, sdělování veřejnosti nebo jakémkoliv druhu obchodování nebo neobchodního šíření je zakázáno! Zejména je zakázána jakákoliv konverze datového souboru nebo extrakce části nebo celého textu, umístování textu na servery, ze kterých je možno tento soubor dále stahovat, přitom není rozhodující, kdo takovéto sdílení umožnil. Je zakázáno sdělování údajů o uživatelském účtu jiným osobám, zasahování do technických prostředků, které chrání elektronickou knihu, případně omezují rozsah jejího užití. Uživatel také není oprávněn jakkoliv testovat, zkoušet či obcházet technické zabezpečení elektronické knihy.*





Copyright © Grada Publishing, a.s.



Copyright © Grada Publishing, a.s.

# Stručný obsah

Předmluva .....	13
Úvod .....	15
1. Syntaxe XML .....	17
2. Přehled podpory XML v PHP5 .....	43
3. (Ne)podpora Unicode v PHP .....	67
4. SimpleXML .....	87
5. SAX .....	105
6. DOM .....	123
7. XMLReader .....	163
8. XPath .....	171
9. Schémata .....	195
10. Validace .....	265
11. XSLT .....	277
12. Webové služby .....	323
13. Zápis XML .....	345
Závěr .....	353
Použitá literatura .....	355
Rejstřík .....	357



# Obsah

<b>Předmluva .....</b>	<b>13</b>
<b>Úvod .....</b>	<b>15</b>
<b>1. Syntaxe XML .....</b>	<b>17</b>
1.1 Elementy a struktura dokumentu .....	17
1.2 Datový model dokumentu .....	19
1.3 Atributy .....	20
1.4 Zápis vyhrazených znaků .....	21
1.5 Názvy elementů a atributů .....	22
1.6 Deklarace XML .....	22
1.7 Komentáře .....	23
1.8 Sekce CDATA .....	23
1.9 Instrukce pro zpracování .....	24
1.10 Automatická kontrola syntaxe .....	25
1.11 Jmenné prostory .....	25
1.12 Práce s bílými znaky .....	30
1.13 Skládání dokumentů .....	32
1.13.1 Entity .....	33
1.13.2 XInclude .....	35
1.14 Katalogové soubory .....	37
1.15 Speciální atributy .....	39
1.15.1 xml:lang .....	39
1.15.2 xml:space .....	39
1.15.3 xml:id .....	40
1.15.4 xml:base .....	40
<b>2. Přehled podpory XML v PHP5 .....</b>	<b>43</b>
2.1 SimpleXML – jednoduše na věc .....	46
2.2 SAX – čteme pěkně popořádku .....	49
2.3 DOM – načteme to do paměti .....	55
2.4 XPath – rychle to najdeme .....	58

2.5	XSLT – jazyk budoucnosti .....	60
2.6	XMLReader – když se zamotáme do SAX .....	62
2.7	Webové služby .....	64
2.8	Závěr .....	66
<b>3.</b>	<b>(Ne)podpora Unicode v PHP .....</b>	<b>67</b>
3.1	Znakové sady, kódování a Unicode .....	67
3.1.1	Znaková sada .....	67
3.1.2	Kódování .....	68
3.1.3	Unicode .....	68
3.2	PHP a práce s řetězci .....	73
3.2.1	Ruční překódování .....	75
3.2.2	Knihovna mbstring .....	78
3.3	Další problémy .....	81
3.3.1	BOM a UTF-8 .....	82
3.3.2	PHP a UTF-16 .....	84
3.3.3	Unicode a porovnávání řetězců .....	84
<b>4.</b>	<b>SimpleXML .....</b>	<b>87</b>
4.1	Načtení dokumentu .....	87
4.2	Konfigurace parseru .....	88
4.3	Čtení hodnot .....	90
4.4	Práce se jmennými prostory .....	93
4.5	Smišený obsah .....	95
4.6	Využití XPathu .....	96
4.7	Modifikace dokumentu .....	97
4.8	Rozšiřování třídy SimpleXMLElement .....	99
4.9	Příklady využití .....	101
<b>5.</b>	<b>SAX .....</b>	<b>105</b>
5.1	Události .....	106
5.1.1	Začátek elementu .....	106
5.1.2	Konec elementu .....	106
5.1.3	Znaková data .....	106
5.1.4	Instrukce pro zpracování .....	107
5.1.5	Začátek mapování prefixu jmenného prostoru .....	107
5.1.6	Konec mapování prefixu jmenného prostoru .....	107
5.1.7	Externí entita .....	108
5.1.8	Další události .....	108
5.2	Vytvoření parseru .....	109



5.3	Konfigurace parseru .....	109
5.4	Registrace obsluhy událostí .....	110
5.5	Čtení dat .....	110
5.6	Obsluha chyb .....	111
5.7	Zapouzdření obsluhy událostí do objektu .....	114
5.8	Přepínání obsluhy událostí .....	117
<b>6.</b>	<b>DOM .....</b>	<b>123</b>
6.1	Objektová reprezentace dokumentu .....	123
6.2	Načtení dokumentu .....	124
6.3	Čtení dokumentu .....	126
6.3.1	Informace o uzlu .....	127
6.3.2	Pohyb po stromu .....	129
6.3.3	Výběr elementů na základě jména .....	131
6.3.4	Průchod celým dokumentem .....	133
6.3.5	Výběr elementu na základě ID .....	139
6.3.6	Čtení atributů .....	139
6.4	Modifikace dokumentu .....	141
6.4.1	Vytváření nových uzlů .....	141
6.4.2	Připojování a odpojování uzlů .....	142
6.4.3	Kopírování a klonování uzlů .....	147
6.4.4	Práce s atributy .....	149
6.4.5	Manipulace s textovými uzly .....	150
6.4.6	Vytvoření nového DOM stromu .....	151
6.4.7	Práce s fragmenty XML .....	151
6.4.8	Uložení dokumentu .....	153
6.5	Konfigurace parseru .....	153
6.6	Obsluha chyb .....	154
6.6.1	Ošetření chyb při načítání XML .....	155
6.6.2	Ošetření výjimek při práci s DOM stromem .....	156
6.7	Zpracování HTML .....	157
6.8	Rozšiřování objektů DOM .....	160
6.9	Další možnosti DOM .....	161
<b>7.</b>	<b>XMLReader .....</b>	<b>163</b>
7.1	Vytvoření a inicializace parseru .....	163
7.2	Konfigurace parseru .....	164
7.3	Čtení dat .....	165
7.3.1	Čtení obsahu elementů .....	166
7.3.2	Přeskočení elementu .....	167

7.3.3 Čtení atributů .....	168
<b>8. XPath .....</b>	<b>171</b>
8.1 Základní struktura výrazu .....	171
8.2 Datový model .....	172
8.3 Testování výrazů .....	174
8.4 Identifikátory osy .....	175
8.5 Testy uzlu .....	177
8.6 Zkrácený zápis .....	178
8.7 Predikáty .....	180
8.8 Příklady dotazů .....	181
8.9 Operátory .....	182
8.9.1 Matematické operátory .....	182
8.9.2 Relační operátory .....	183
8.9.3 Logické spojky .....	184
8.9.4 Sjednocení seznamů uzlů .....	184
8.10 Funkce .....	184
8.10.1 Funkce pro práci s uzly .....	185
8.10.2 Řetězcové funkce .....	186
8.10.3 Logické funkce .....	189
8.10.4 Funkce pro práci s čísly .....	189
8.11 Využití XPathu v DOM .....	190
8.12 Podpora XPath v dalších rozhraních .....	194
<b>9. Schémata .....</b>	<b>195</b>
9.1 Význam a historie schémat .....	195
9.1.1 Význam schémat .....	196
9.1.2 Historický vývoj jazyků pro popis schématu dokumentu .....	196
9.1.3 Srovnání nejpoužívanějších jazyků pro popis schématu dokumentu .....	198
9.2 DTD .....	202
9.2.1 Deklarace elementů .....	203
9.2.2 Deklarace atributů .....	205
9.2.3 Připojení DTD k dokumentu .....	207
9.2.4 Validace dokumentů oproti DTD .....	208
9.3 XML schémata .....	209
9.3.1 Datové typy .....	210
9.3.2 Jednoduché datové typy .....	210
9.3.3 Komplexní datové typy .....	217
9.3.4 Jmenné prostory .....	224
9.3.5 Připojení schématu k dokumentu a validace .....	225

<b>9.4</b>	<b>RELAX NG</b> .....	<b>226</b>
9.4.1	Základní vzory .....	228
9.4.2	Pokročilé vzory .....	233
9.4.3	Datové typy .....	241
9.4.4	Modularizace schématu .....	248
9.4.5	Jmenné prostory .....	250
9.4.6	Validace oproti RELAX NG schématu .....	252
<b>9.5</b>	<b>Schematron</b> .....	<b>252</b>
9.5.1	Validace pomocí XSLT .....	253
9.5.2	Vložení Schematronu do XML Schema .....	254
9.5.3	Vložení Schematronu do RELAX NG .....	255
9.5.4	Pokročilá validace .....	256
<b>9.6</b>	<b>Best-practices pro návrh schémat</b> .....	<b>258</b>
9.6.1	Elementy nebo atributy .....	258
9.6.2	Jmenné prostory .....	259
9.6.3	Názvy elementů a atributů .....	260
9.6.4	Defaultní a fixní hodnoty .....	261
9.6.5	Verzování schémat .....	261
9.6.6	Rozšiřitelnost schémat .....	261
<b>10.</b>	<b>Validace</b> .....	<b>265</b>
10.1	Validace pomocí rozhraní DOM .....	266
10.2	Validace pomocí rozhraní SimpleXML .....	268
10.3	Validace pomocí rozhraní XMLReader .....	268
10.4	Schematron .....	272
10.5	Praktické využití validace .....	273
<b>11.</b>	<b>XSLT</b> .....	<b>277</b>
11.1	Základy XSLT .....	277
11.2	Cykly – iterativní zpracování .....	282
11.3	Řazení dat .....	285
11.4	Podmíněně zpracování .....	288
11.5	Generování výstupu .....	291
11.5.1	Generování elementů a atributů .....	292
11.5.2	Generování textového výstupu .....	292
11.5.3	Generování elementů a atributů s předem neznámým názvem .....	293
11.5.4	Generování speciálních konstrukcí .....	294
11.6	Zpracování dokumentů se jmennými prostory .....	294
11.7	Předávání parametrů .....	297
11.8	Podpora XSLT v PHP .....	300

11.9	Volání PHP funkcí z XSLT .....	305
11.10	Funkce přidané do XPathu .....	308
11.10.1	document() .....	308
11.10.2	key() .....	311
11.10.3	format-number() .....	311
11.10.4	current() .....	313
11.10.5	unparsed-entity-uri() .....	313
11.10.6	generate-id() .....	314
11.10.7	system-property() .....	317
11.10.8	element-available() .....	318
11.10.9	function-available() .....	318
11.11	Spouštění transformací na klientovi .....	318
12.	Webové služby .....	323
12.1	Webové služby à la SOAP .....	324
12.1.1	Pod pokličkou SOAP .....	325
12.1.2	Pod pokličkou WSDL .....	326
12.1.3	PHP jako klient webové služby .....	330
12.1.4	Webová služba .....	333
12.2	Webové služby à la REST .....	335
12.3	AJAX .....	337
13.	Zápis XML .....	345
13.1	Ruční generování XML .....	345
13.2	Generování pomocí DOM .....	348
13.3	Využití třídy XMLWriter .....	350
	Závěr .....	353
	Použitá literatura .....	355
	Rejstřík .....	357

# Předmluva

Když jsem před více než deseti lety psal předmluvu k dnes již legendární knize *PHP – tvorba interaktivních internetových aplikací* [11], slíbil jsem v ní, že na doprovodný web knihy umístím informace o práci s formátem XML, které se do knihy již nevešly. Z časových důvodů k naplnění tohoto slibu nikdy nedošlo. Jako satisfakci proto přijměte tuto knihu. Její náplní je pouze XML a jeho použití v PHP.

Deset let je dlouhá doba, ale myslím, že čekání se vyplatilo. Podpora XML byla v PHP až do jeho verze 5.0 poměrně partyzánská. A teprve od verze 5.2 lze PHP považovat za jazyk, ve kterém se dá s XML rozumně pracovat.

O významu XML dnes již není potřeba nikoho přesvědčovat. Ať se nám to líbí nebo ne, XML je zkrátka všude a ve webových aplikacích je potřeba s tímto formátem pracovat. Ať už se jedná o importy a exporty dat, transformaci dat pro prezentační vrstvu nebo backend pro AJAXovou aplikaci. Cílem této knihy je naučit vás používat všechna existující rozhraní pro práci s technologiemi XML, která PHP nabízí. Popsán a vysvětlen je však i samotný jazyk XML, jeho syntaxe a navazující technologie jako XML schémata, dotazovací jazyk XPath a transformační jazyk XSLT. Pro pochopení výkladu tak není nutná žádná velká předchozí zkušenost s XML – vše potřebné je průběžně vysvětleno.

V dnešní době začínají být papír a knihy považovány za příliš konzervativní médium. Asi jsem staromilec a mám tištěné knihy rád. Nicméně na adrese <http://www.kosek.cz/knihy/phpxml/> najdete další informace související s knihou – příklady ke stažení, opravy chyb apod. Máte-li ke knize nějaké připomínky, uvítám je na mojí emailové adrese <jirka@kosek.cz>.

Na vzniku knihy má zásluhu mnoho lidí. Nevyčerpatelnou trpělivost prokázali šéfredaktoři počítačové redakce Miroslav Lochman a Daniel Vrba, kteří vydrželi pět let čekat na dokončení knihy. Pokud v knize nebude příliš chyb, je to zásluha korektorky Zuzany Vrbové a redaktorky Evy Steinbachové. Největší dík však patří mé ženě Lence a dětem, kteří trpělivě snášely mé úteky k počítači při psaní knihy.

Přeji vám příjemné čtení knihy.

*Jirka Kosek*

Liberec, 10. dubna 2009



# Úvod

Vývoj moderních webových aplikací klade na vývojáře vysoké nároky, je potřeba znát široké spektrum technologií. Počínaje jazyky HTML a CSS pro definici samotné stránky a jejího vzhledu, přes Javascript pro vytváření vysoce interaktivních aplikací, po nějaký serverový jazyk jako je PHP. Každá větší aplikace navíc potřebuje někam ukládat data, typicky do databáze. K tomu je potřeba znát principy protokolu HTTP a vědět, jak obcházet jeho limity. A aby toho nebylo málo, na mnoha místech se vývojář webové aplikace setká i s formátem XML.

Když XML v polovině 90. let minulého století vznikalo, původní smělé plány byly, že zcela nahradí jazyk HTML při doručování obsahu do prohlížeče. Tato myšlenka se však ukázala jako příliš revoluční a navíc problémy spojené s jazykem XHTML a jeho podporou v prohlížečích v očích mnoha webových vývojářů nevrhly na XML příliš růžové světlo. Nicméně technologie XML jsou dnes pevnou součástí mnoha webových technologií, formátů a protokolů, takže je potřeba práci s tímto formátem ovládat.

Kde se na webu s XML může vývojář setkat? Syntaxi XML využívají mnohé prezentační formáty – počínaje jazykem XHTML, přes stále populárnější vektorový grafický formát SVG, až po jazyky pro definici uživatelského rozhraní v moderních RIA (Rich Internet Application) prostředích, jako je XAML v Silverlightu a MXML ve Flashi. Pokud tedy vaše skripty v PHP v minulosti generovaly převážně HTML kód, časem budou přicházet požadavky na dynamickou tvorbu modernějších, na XML založených, formátů.

XML dnes zcela dominuje na poli publikování metainformací. Jedná se například o formáty pro publikování přehledů nových článků, jako jsou RSS či Atom. Protože začleňování sémantiky ve strojově čitelné podobě přímo do webových stránek je stále v plenkách, mnoho vyhledávačů nabízí vlastní formáty, ve kterých jim můžete předávat informace vylepšující vyhledávání – například Google Sitemap nebo Google Base.

Další využití XML je pro komunikaci a předávání dat. XML se využívá jednak pro výměnu dat mezi backendy jednotlivých aplikací a dále pak v AJAXových aplikacích pro zasílání aktualizací dat do prohlížeče. „Enterprise“ aplikace pak pro samotnou komunikaci nevyužívají prostě XML, ale komplexnější mechanismus webových služeb.

Diverzita koncových zařízení používaných pro přístup k webu se také stále zvyšuje. Webové stránky se už neprohližejí jen z klasického počítače, ale i z chytrých telefonů nebo různých PDA. Mnohé aplikace potřebují uspokojivě řešit možnost kvalitního tisku. Už nestačí vytvořit aplikaci, která výstupy generuje jen jako HTML. Pro jednotlivá koncová zařízení je potřeba generovat odlišné formáty výstupu a webovou aplikaci je potřeba obohatit o flexibilní prezentační vrstvu. Pro vytvoření takové vrstvy lze využít i jazyk XML a stylové technologie jako XSL.

Výše uvedený výčet toho, kde se na webu můžeme setkat s XML, jistě není úplný. Pouze potvrzuje to, že webový vývojář se dnes neobejde bez znalosti tohoto formátu a práce s ním. Tato kniha vás naučí vše potřebné o formátu XML a navazujících technologiích a o tom, jak lze s tímto formátem pracovat v PHP.

První kapitola je určená zejména pro čtenáře, kteří ještě neznají formát XML. Seznámí se zde se syntaxí jazyka a naučí se ji kontrolovat.

Druhá kapitola pak stručně shrnuje a ukazuje, jaké možnosti pro práci s XML nabízí PHP. Je to ideální místo pro porovnání jednotlivých přístupů pro práci s XML. Nemusíte tak číst celou knihu, ale v této kapitole zjistíte, jaký způsob práce s XML je pro vás nejhodnější a ten si dále podrobněji nastudujete v odpovídající samostatné kapitole.

Třetí kapitola přímo nesouvisí s XML, ale ukazuje, jak lze v PHP částečně obejít chybějící podporu znakové sady Unicode, kterou využívá i jazyk XML.

Následují čtyři kapitoly, které podrobně popisují jednotlivá rozhraní pro práci s XML – SimpleXML, SAX, DOM a XMLReader. Výběr vhodného rozhraní záleží na povaze dat, která čtete, a na tom, jak je potřebujete zpracovat.

Osmá kapitola seznamuje s dotazovacím jazykem XPath, který nabízí jednoduchou a zároveň mocnou metodu pro vyhledávání a výběr dat v dokumentech XML. Kromě samotného dotazovacího jazyka je zde samozřejmě popsáno, jak jej používat v kombinaci s rozhraními DOM a SimpleXML.

Další dvě kapitoly se věnují kontrole (validaci) dokumentů XML. Zvláště v otevřeném prostředí internetu je potřeba počítat s nejhorším a všechny vstupy do aplikace pečlivě kontrolovat. Pro dokumenty XML takovou kontrolu nabízejí schémata, která dokáží popsat povolenou strukturu a datové typy dokument XML. Devátá kapitola tak popisuje nejpoužívanější schémové jazyky a v desáté kapitole je pak ukázáno, jak pomocí nich prakticky v PHP kontrolovat data uložená v XML.

Jedenáctá kapitola vysvětluje základy jazyka XSLT a jeho použití v PHP. XSLT je nejvhodnější prostředek pro transformace dokumentů XML do dalších formátů, včetně formátu HTML. Nalezne tak uplatnění například v prezentační vrstvě webové aplikace.

Následující dvanáctá kapitola se pak věnuje komunikaci mezi aplikacemi s využitím XML. Popsány jsou jak klasické webové služby, tak jednodušší mechanismy jako REST a AJAX.

Poslední třináctá kapitola pak ukazuje možnosti pro generování dokumentů XML na výstupu skriptu.

Kniha je zaměřena zejména na vysvětlení principů a na ukázky využití jednotlivých technologií a knihoven PHP. Ve většině případů jsou popsány všechny možnosti jednotlivých knihoven PHP. Nicméně kniha primárně neslouží jako referenční příručka, pro tyto účely je vhodné nahlížet i do dokumentace PHP na adrese <http://docs.php.net/manual/en/>.

Všechny příklady byly testovány s PHP ve verzi 5.2 a v současné době nic nenasvědčuje tomu, že by se v blízké době mělo v jazyce PHP měnit něco, co by způsobilo nefunkčnost skriptů. Většina použitých knihoven je standardní součástí PHP. Chcete-li používat XSLT, je potřeba do PHP přidat modul `php_xsl`, pro webové služby je zapotřebí modul `php_soap` a kapitola o Unicode využívá některé funkce z modulu `php_mbstring`.

Dlouhé řádky ve výpisech, které musely být rozděleny, jsou označeny pomocí znaku `▶`.



# 1.

## Syntaxe XML

Chceme-li pracovat s dokumenty XML, musíme samozřejmě vědět, jak tyto dokumenty vypadají. V této kapitole se proto seznámíme se syntaxí jazyka XML a dalšími jeho rysy, které bychom měli znát. Znáte-li již XML dobře, a zajímá vás jen, jak se s ním pracuje v PHP, můžete tuto kapitolu směle přeskočit.

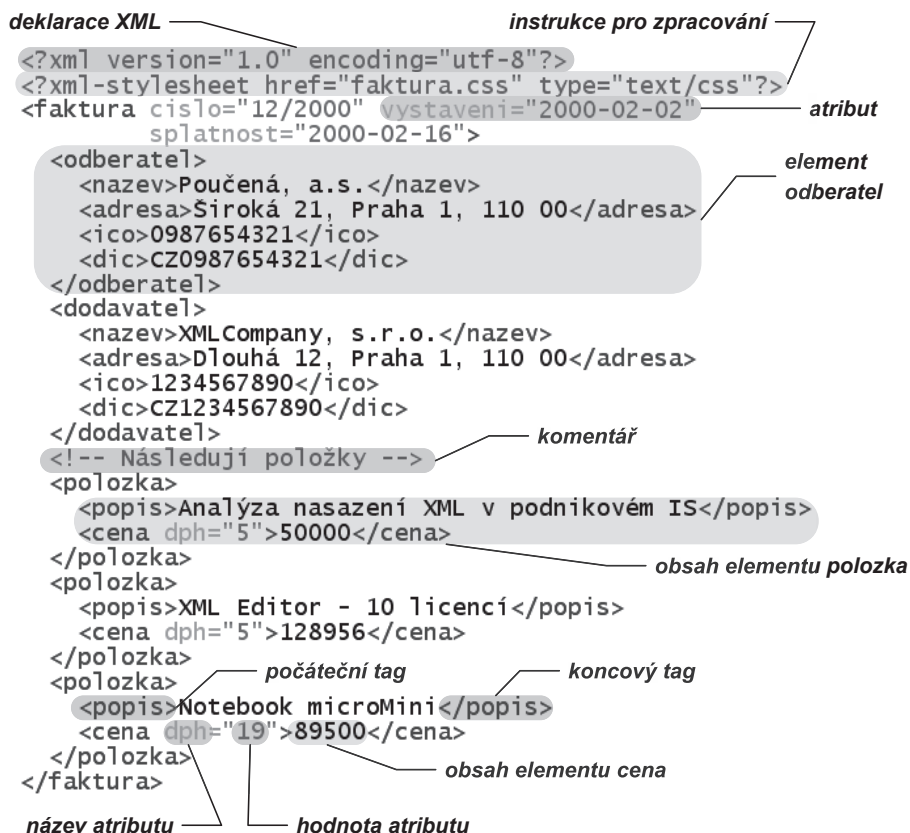
### 1.1 Elementy a struktura dokumentu

Každý XML dokument se skládá z *elementů*. Elementy se v textu vyznačují pomocí tzv. *tagů*. Většinou elementů odpovídají dva tagy – počáteční a koncový.

```
<para>Toto je obsah elementu para.</para>
```

Ukázka obsahuje jeden element para. Jeho obsah je vyznačen pomocí tagů `<para>` (počáteční tag) a `</para>` (ukončovací tag). Jen na okraj poznamenejme, že výše uvedená ukázka je asi nejjednodušším dokumentem XML, který vůbec můžeme vytvořit.

Názvy tagů se zapisují mezi znaky `<` a `>`. Ukončovací tag má před svým názvem ještě znak `/`, aby se odlišil od počátečního.



Obrázek 1.1: Základní části dokumentu XML

Některé elementy nemusí mít žádný obsah. Můžeme je samozřejmě zapisovat tak, že za počátečním tagem uvedeme hned ten koncový.

```
<para>Toto je obsah elementu para.<br></br> A tohle také.</para>
```

Není to však příliš pohodlné, a proto můžeme v XML použít ještě jednu variantu tagu, která říká, že element nemá žádný obsah. Počáteční tag ukončíme dvojicí znaků ,/>' místo pouhého ,>' a koncový tag vynecháme.

```
<para>Toto je obsah elementu para.<br/> A tohle také.</para>
```

Každý dokument XML musí obsahovat pro všechny počáteční tagy odpovídající koncový tag, nebo musí být počáteční tag zapsán jako element s prázdným obsahem. Následující ukázky jsou ukázkami špatných dokumentů, které nevyhovují specifikaci XML.

```
<para>Toto je obsah elementu para.<br> A tohle také.</para>
```

Ukázka je chybná, neboť tag <br> není ukončen.

```
<para>Toto je obsah elementu para.<br/> A tohle také.</oara>
```