

**Univerzita Jana Evangelisty Purkyně  
v Ústí nad Labem  
Přírodovědecká fakulta**

UNIVERZITA J. E. PURKYNĚ V ÚSTÍ NAD LABEM

Přírodovědecká fakulta

**Programování pro internet  
Příprava na zkoušku**

**Vypracoval:** Kopyto

**Studijní program:** Aplikovaná informatika

**Studijní obor:** Informační systémy

ÚSTÍ NAD LABEM 2024

# Obsah

<b>Obsah</b>	<b>1</b>
<b>1 XML</b>	<b>2</b>
1.1 základní syntaxe . . . . .	2
1.2 Jmenný prostor . . . . .	2
1.3 obsah elementu . . . . .	3
1.4 XML schéma . . . . .	3
1.5 XML datové typy . . . . .	4
<b>2 DTD a XSD</b>	<b>5</b>
2.1 DTD . . . . .	5
2.1.1 Datové typy obsahu . . . . .	5
2.1.2 Datové typy atributů . . . . .	5
2.2 XSD . . . . .	6
2.2.1 Primitivní datové typy . . . . .	6
2.2.2 Odvozené datové typy . . . . .	7
2.2.3 Speciální datové typy . . . . .	7

# 1. XML

XML (eXtensible Markup Language) je značkovací jazyk, který umožňuje uživatelům definovat svá vlastní strukturovaná data. Základní syntaxe XML vyžaduje, aby každý dokument obsahoval jeden kořenový element, který obaluje všechny ostatní elementy. XML soubory musí být "well-formed", což znamená, že musí správně dodržovat strukturu značek a pravidla syntaxe, jako je správné uzavírání značek a uvozování atributů hodnotami v uvozovkách.

pod tu to otázky spadaj pod otázky: základní syntaxe, jmenný prostor, schéma a datatypes

## 1.1 základní syntaxe

Pravidla pro psaní XML souboru zahrnují použití správného prologu, což je volitelná deklarace na začátku souboru specifikující verzi XML a použitou značkovou sadu (např. `<?xml version="1.0" encoding="UTF-8"?>`). Jména elementů a atributů by měla být srozumitelná a relevantní k obsahu, který reprezentují, a neměla by obsahovat mezery ani speciální znaky. Co se týče entit, XML definuje několik vestavěných entit pro speciální znaky (např. `&lt;` pro znak menší než, `&gt;` pro znak větší než, `&amp;` pro ampersand, atd.), které umožňují vkládat do dokumentů speciální znaky. Výše uvedené aspekty jsou klíčové pro správné psaní a porozumění XML.

## 1.2 Jmenný prostor

Jmenný prostor (namespace) v XML je metoda, která umožňuje rozlišovat elementy a atributy, které by mohly mít stejná jména, ale různé významy, v závislosti na kontextu, ve kterém jsou použity. Jmenné prostory jsou definovány pomocí atributu `xmlns` v základním nebo jiném elementu XML dokumentu, čímž se určuje URI (Uniform Resource Identifier), které slouží jako unikátní identifikátor pro daný jmenný prostor.

Pro přidání jmenného prostoru do elementu v XML dokumentu je běžně používán atribut `xmlns`, například `<knihy xmlns="http://www.example.com/knihy">`. Tento přístup označuje, že element `knihy` a všechny jeho dětské elementy patří do jmenného prostoru `http://www.example.com/knihy`. Když chcete použít elementy z více jmenných prostorů v jednom dokumentu, můžete definovat prefixy, jako je `xsl:template`, kde `xsl` je prefix odkazující na jmenný prostor definovaný například takto: `xmlns:xsl="http://www.w3.org/1999/XSL/Transform"`.

Tato struktura umožňuje XML dokumentům být flexibilní a zároveň přesně definované, což je klíčové pro jejich správnou funkčnost a integraci v různých aplikacích a systémech.

## 1.3 obsah elementu

Obsah elementu v XML se týká všeho, co je umístěno mezi otevírací a zavírací značkou tohoto elementu. Tento obsah může zahrnovat text, další elementy (což umožňuje vytvářet hierarchické struktury), komentáře, a entity (které umožňují zahrnutí speciálních znaků, jako jsou například &, <, >). Kromě toho může obsah zahrnovat data v různých formátech, jako jsou čísla, řetězce nebo datумы, v závislosti na specifikaci a účelu daného XML dokumentu. Význam a struktura obsahu dokumentu je definována schematem nebo typem dokumentu, který XML používá, což umožňuje flexibilní a přesnou manipulaci s daty.

## 1.4 XML schéma

XML schema, obvykle definované ve W3C XML Schema Definition Language (XSD), umožňuje návrhářům specifikovat pravidla a omezení pro strukturu XML dokumentu, včetně typů dat pro jednotlivé elementy a atributy, jejich výskyt v dokumentu, a vztahy mezi různými elementy. Pomocí XML schema můžete například definovat, že určitý element musí obsahovat pouze číselné hodnoty, nebo že jiný element může obsahovat jiné elementy v určitém pořadí. XML schema také podporuje vytváření vlastních datových typů a použití omezení, jako jsou minimální a maximální hodnoty nebo specifické formáty pro řetězce. Tato schémata jsou klíčová pro automatizovanou validaci XML dokumentů, což zajišťuje, že data vyhovují definovaným standardům a jsou konzistentní napříč různými systémy a aplikacemi.

```
<?xml version="1.0" encoding="UTF-8"?>
<knihovna>
  <kniha id="001">
    <nazev>Ůvodce galaxií pro řstopae</nazev>
    <autor>Douglas Adams</autor>
    <vydano>1979</vydano>
  </kniha>
  <kniha id="002">
    <nazev>1984</nazev>
    <autor>George Orwell</autor>
    <vydano>1949</vydano>
  </kniha>
</knihovna>
```

Obrázek 1.1: Ukázka XML souboru

```

<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.someweb.com"
xmlns="http://www.someweb.com"
elementFormDefault="qualified">
. . . . .
</xs:schema>

```

Obrázek 1.2: Ukázka přidání schematu

## 1.5 XML datové typy

XML Schema poskytuje širokou škálu vestavěných datových typů, které umožňují přesně specifikovat a omezovat data v XML dokumentech. Základní datové typy zahrnují string pro textové řetězce, integer pro celá čísla, boolean pro pravdivostní hodnoty, a date pro data. Kromě těchto základních typů, XML Schema definuje i složitější datové typy jako decimal pro čísla s desetinnými místy, duration pro časové úseky, nebo gYearMonth pro specifikaci roku a měsíce. Uživatelé také mohou vytvářet vlastní datové typy pomocí omezení (restriction) na stávající typy, což umožňuje velmi specifickou validaci dat podle potřeb aplikace. Tato flexibilita a přesnost jsou klíčové pro efektivní využití XML v různých aplikacích a systémech.

## 2. DTD a XSD

### 2.1 DTD

DTD (Document Type Definition), což je jeden z prvních a základních mechanismů pro definování struktury a pravidel pro XML dokumenty. DTD umožňuje specifikovat, které elementy, atributy a entity mohou v dokumentu existovat, jaké jsou jejich vztahy a jak často se mohou v dokumentu objevit. Toto je klíčové pro zajištění konzistence dat mezi různými systémy a pro validaci struktury XML dokumentů před jejich zpracováním.

DTD je obvykle zapsáno buď přímo v XML dokumentu, nebo jako externí soubor, který je poté odkazován z XML. K definování pravidel v DTD se používají deklarace elementů a atributů. Například, můžete definovat, že každý dokument musí obsahovat kořenový element <knihovna>, který může obsahovat několik <knihy> elementů, přičemž každý <knihy> element musí obsahovat <nazev>, <autor>, a <vydano>.

#### 2.1.1 Datové typy obsahu

1. **#PCDATA** - Znamená "Parsed Character Data", což umožňuje vložení běžného textu. Tento typ se používá, když chcete, aby element obsahoval jen text.
2. **Prázdný** - Pomocí klíčového slova EMPTY můžete specifikovat, že element nesmí obsahovat žádný obsah ani žádné další elementy.
3. **Smíšený obsah** - Můžete definovat element, který může obsahovat kombinaci textu a dalších elementů. To se specifikuje pomocí modelu obsahu kombinujícího #PCDATA s názvy dalších elementů.

#### 2.1.2 Datové typy atributů

1. **CDATA** - Pro běžné textové řetězce.
2. **ID** - Unikátní identifikátor v rámci dokumentu.
3. **IDREF/IDREFS** - Odkaz na element s typem ID.
4. **NMTOKEN/NMTOKENS** - Token, který musí odpovídat XML názvovým konvencím (např. musí začínat písmenem).
5. **ENUMERATION** - Umožňuje definovat seznam povolených hodnot.
6. **NOTATION** - Odkaz na notaci definovanou v DTD.

```

<!DOCTYPE knihovna [
<!ELEMENT knihovna (kniha+)>
<!ELEMENT kniha (nazev, autor, vydano)>
<!ELEMENT nazev (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT vydano (#PCDATA)>
]>

```

Obrázek 2.1: Ukázka DTD souboru

## 2.2 XSD

XSD je mocný nástroj pro definování struktury a validaci XML dokumentů. Na rozdíl od DTD poskytuje XSD podrobnější specifikace pro datové typy, podporuje jmenné prostory a umožňuje vytváření složitých a omezujících pravidel pro XML dokumenty. XSD se používá pro definování pravidelné struktury XML dokumentů, což zajišťuje, že data splňují definované formáty a omezení. To je nezbytné pro aplikace, které vyžadují vysokou úroveň integrity dat, jako jsou finanční systémy, zdravotnické informační systémy, a webové služby, kde je důležitá správná a konzistentní interpretace dat mezi různými systémy a platformami. Hlavní kategorie datových typů

1. **Primitivní datové typy** - To jsou základní typy, ze kterých jsou odvozeny všechny ostatní typy.
2. **Odvozené datové typy** - Tyto typy jsou odvozeny z primitivních typů a mohou zahrnovat dodatečná omezení.
3. **Speciální datové typy**

### 2.2.1 Primitivní datové typy

- **'xs:string'** - Pro textové řetězce.
- **'xs:boolean'** - Pro pravdivostní hodnoty (true/false).
- **'xs:decimal', 'xs:integer', 'xs:float', 'xs:double'** - Pro číselné hodnoty, kde 'xs:decimal' a 'xs:integer' jsou pro celá a desetinná čísla bez ztráty přesnosti, zatímco 'xs:float' a 'xs:double' jsou pro plovoucí desetinné čárky s možnou ztrátou přesnosti.
- **'xs:date', 'xs:time', 'xs:dateTime', 'xs:duration'** - Pro datum, čas a jejich kombinace.

## 2.2.2 Odvozené datové typy

- **'xs:byte', 'xs:short', 'xs:long', 'xs:unsignedShort', atd.** – Různé formáty číselných typů pro specifické potřeby, jako jsou velikosti a znaménka.
- **'xs:nonNegativeInteger', 'xs:positiveInteger'** – Pro nezáporná a kladná celá čísla.

## 2.2.3 Speciální datové typy

- **'xs:ID', 'xs:IDREF', 'xs:ENTITY', atd.** – Speciální účelové typy pro unikátní identifikátory, odkazy na identifikátory a jména entit.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="knihovna">
    <xs:complexType>
      <xs:sequence>
        <xs:element
          name="kniha"
          maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element
                  name="nazev"
                  type="xs:string"/>
                <xs:element
                  name="autor"
                  type="xs:string"/>
                <xs:element
                  name="vydano"
                  type="xs:date"/>
              </xs:sequence>
              <xs:attribute
                name="id"
                type="xs:string"
                use="required"/>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

Obrázek 2.2: Ukázka XSD souboru



# Seznam použité literatury

1. SY-RAHMADI. Resale HDB Flat Prices 2012 - 2023 [<https://www.kaggle.com/datasets/syrahmadi/resale-hdb-flat-prices-2000-2022/data>]. 30. prosinec 2023.
2. CLICKHOUSE. ClickHouse Docs. ClickHouse. 2024. Dostupné také z: <https://clickhouse.com/docs/>.