

①

Az xsd fájl definiálja hogy milyen elemek és attribútumok jelenhetnek meg a xml fájlban

Elemi típusok: ~~sz~~ atom, lista, union

Összetett típusok: ~~gond~~ gyerek elemeket tartalmaznak

```
<xs:complexType> *  
  <xs:sequence>  
  + /xs:sequence  
</xs:complexType>
```

xs:sequence-ba felsorolni gyerek elemeket xs:element-ben

Származtatás: <xs:complexType name="name1"/>
complexType-nek definiálunk egy struktúrát, azután adunk majd <xs:extension base="name1"/>-al tudjuk származtatni a típust

Hivatkozás: ref attribútummal lehet egy már definiált struktúrára hivatkozni,

```
<xsd:element name='p'>  
  xsd:element name='pname'/>  
</xsd:element>
```

```
<xsd:element>  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element name='age'>  
      <xsd:element ref='p'>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

2.

XML + fa struktúráként kezel, DOM+ra hasonlítan
xml elemek, jellemzők keresése, hivatkozására használják,
sok beépített függvény

Kifejezés: Menteinek listáját

Görögpaták amelyek ellenkezősége a mintára, azok közül
hivatkozik rá.

függvények self: maga a node

child: gyerek node-ok

descendants: gyerek is azok gyerekei, teljes egészében

parent: a nodeot tartalmazó node

ancestor: olyan nodeok amelyek fölé tartoznak

ancestor-or-self: self is ancestor uniója

operatorok: + - = * div != < <= > >= and or not

pl:

```
<books>
```

```
<book>
```

```
  <name> <price>
```

```
</book>
```

```
...
```

```
</books>
```

abszolút relatív utvonál //books/book[1]name → relatív utvonál

//book[price > 50] → abszolút utvonál

⑤ XSL

A2 xsl xml-Göl xml-Gc konvertáló, transformáló nyelv

- xml formátum
- xpath - út használat, választás
- nyelvnek van kérése