

RUSSIL PETER
QOHYCR

- 1) DOM parser: Complete list of nodes and attributes.
SAX parser: Event-driven XML document parser. ~~It~~ It does not store the whole document in memory.
XPath parser: XSLT based XML query language.

2) XPath: szabvány az XML dokumentum részletek kijelölésére.

általános alak:

tengely: komponenssorozás [szekvencia].

- Navigáció
- Komponenssorozás
- Szekvencia: Tartalomra szűkítés.

3) DOM azaz bármely egy megvalósítás programozható API felület.

DOM = Document Object Model

Az értelmező bekapcsa az XML dokumentumot és egy fát, mint objektumot épít belőle.

Az előállított objektum - fát a metódusain keresztül lehet olvasni és módosítani. Megerőltet hatékonyan implementálni a SAX-ra épül.

Vagy a memória igénye.

Feldolgozás lépései: - Dokumentum olvasás betöltése

- Objektum fa betöltése

- Dokumentum tagok elérése.

- Dokumentum metódusok meghívása

- Dokumentum orieralizálása.

Metódusok: Elem hozzáférése

Elem módosítása

Elem törlése

Adattagjai: Document

PI

Element

Attribute

Node

4) XML Schema:

szerepe: Egy vázlat, mely megmondja az XML dokumentum felépítését.
fő típusai: date, float, string, integer

DTP előnyei:

- elemi szerkezetű elemek
- korlátozott integrációs elemek
- nincs adattípus meghatározás
- globális nevek
- névspace típusok
- szoros kapcsolat az XML-hez
- entity támogatás

hátrányai:

- nincs lehetőség összetett szerkezetű elem def.
- típus ábrázolásának nincs.

főbb elemek:

- <! ELEMENT név szerkezet >
- <! ATTLIST elem attr-név adattípus meghatározás >

5.) XSLT komponens orientáltság: megkeres egy adott komponens

```
<xsl:choose>
  <xsl:when test="if1">
    <!-- tevékenység -->
    <xsl:when>
  <xsl:when test="if2">
    <!-- tevékenység -->
    <xsl:when>
</xsl:choose>
```

Görgetés:

```
<xsl:for-each-group select="if" group-by="group name">
  <!-- tartalom -->
</xsl:for-each-group>
```

rendezés:

```
<xsl:sort>
  <!-- tartalom -->
</xsl:sort>
```