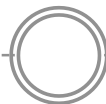




A nyelvtechnológia alapjai



4. Gyakorlat

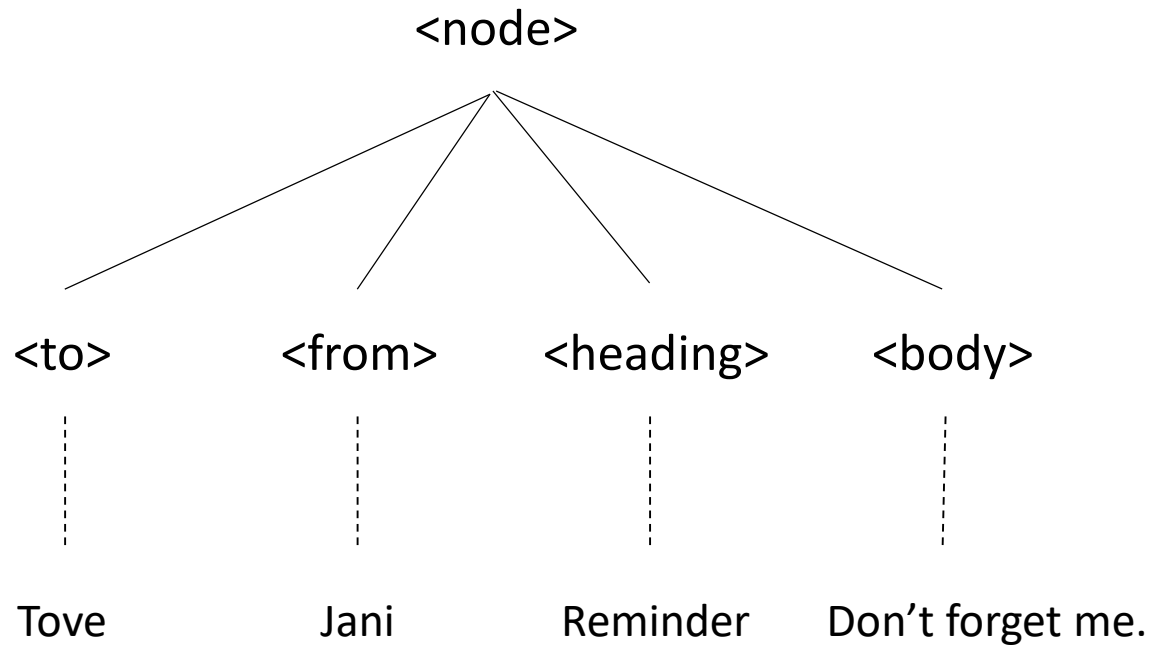
XML: SAX, DOM



- Kétféle megközelítés
 - Az XML sima szöveg.
 - Végigolvassuk és közben feldolgozzuk
 - Elemzés: hol tartunk éppen
 - **SAX** (Simple API for XML)
 - Az XML fa-struktúra.
 - Felépítjük a fát (a memóriában), azt vizsgáljuk, és feldolgozzuk
 - Elemzés: a fa felépítése
 - **DOM** (Document Object Model)

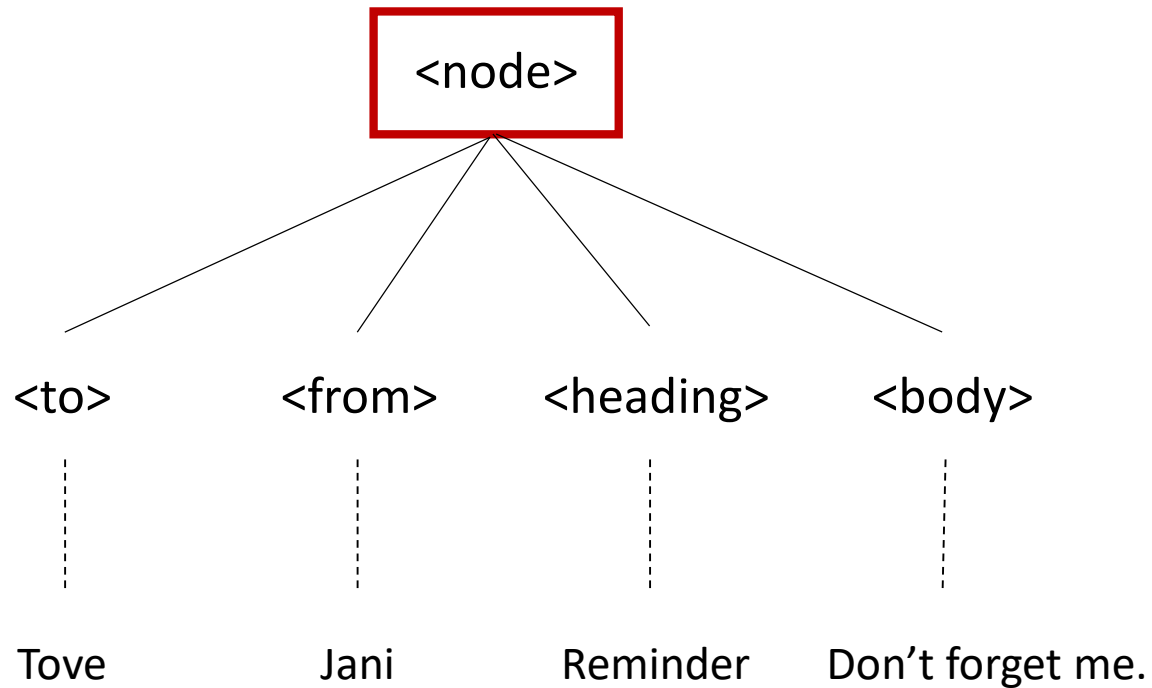


<node><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>



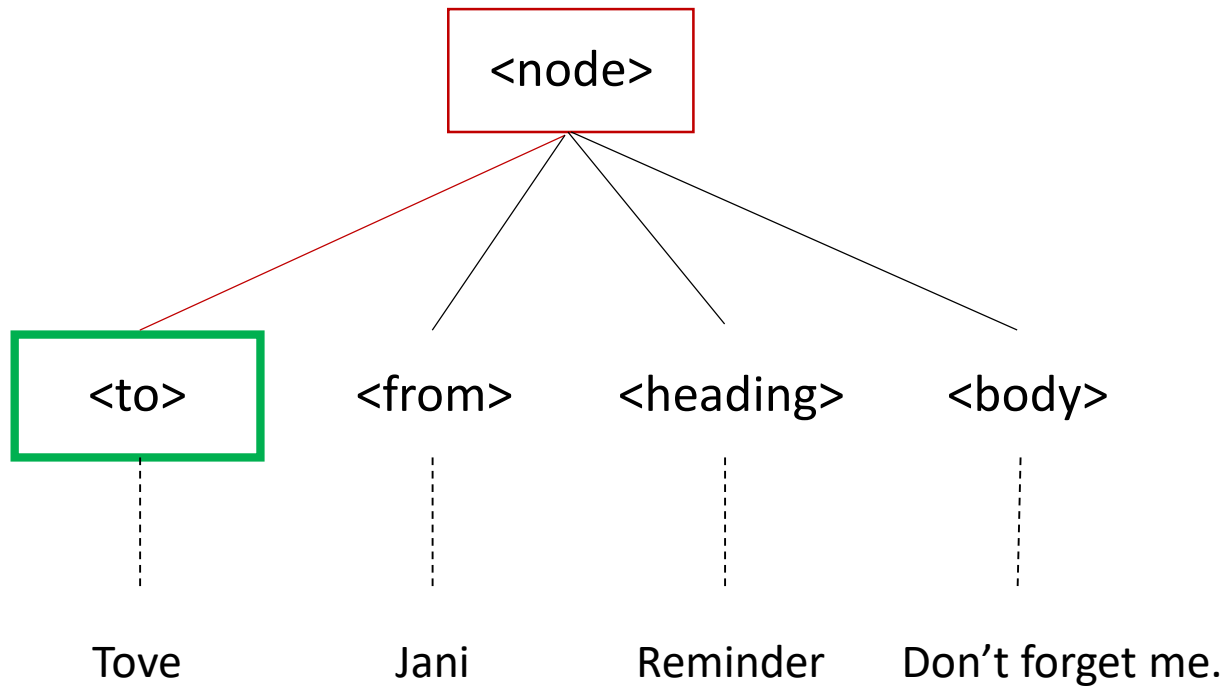


<node> <to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>



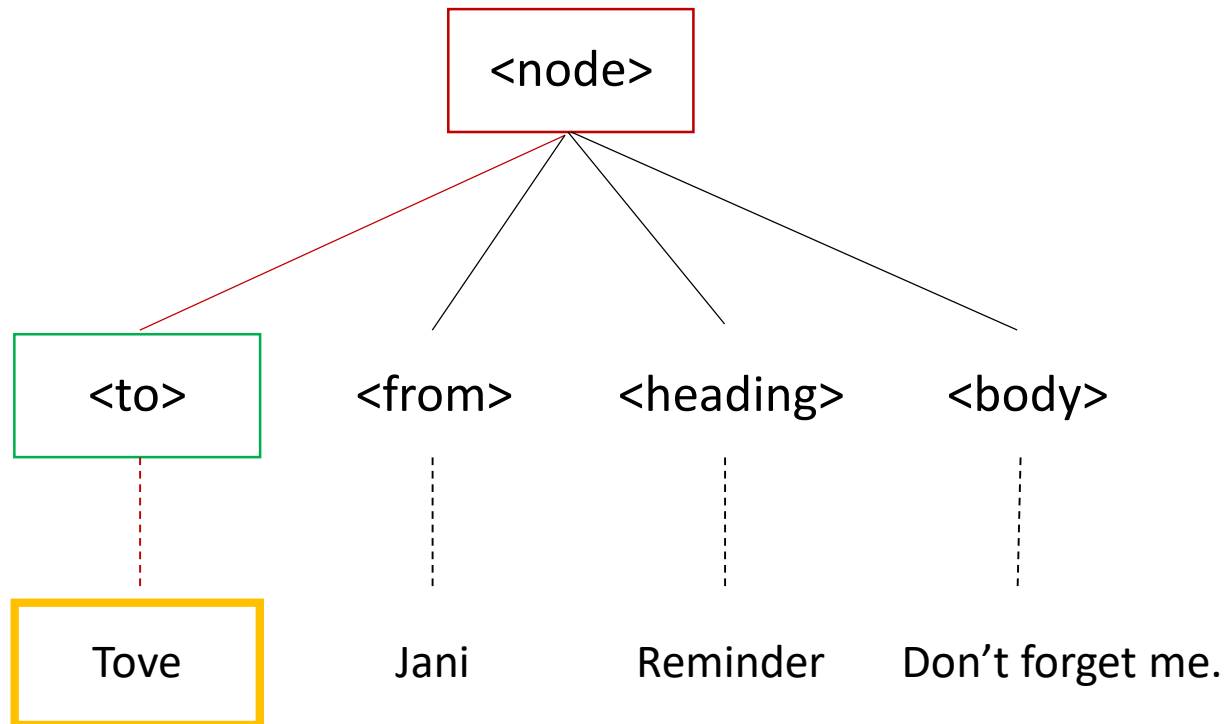


`<node><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>`



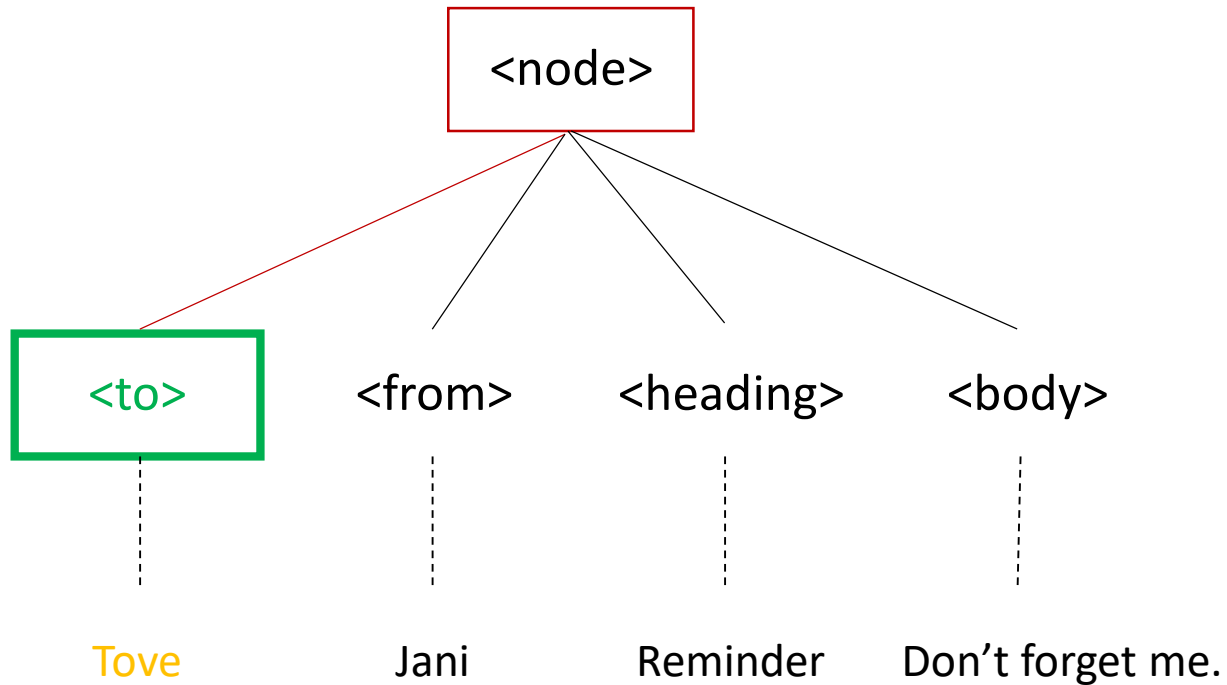


`<node><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>`



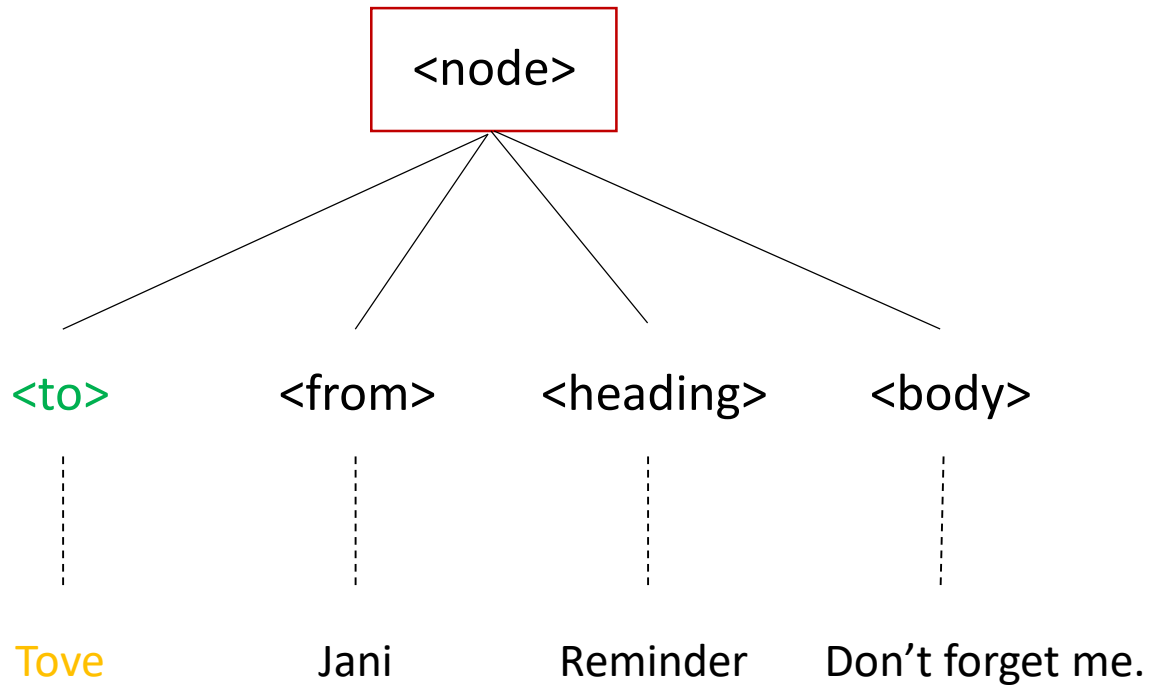


`<node><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>`



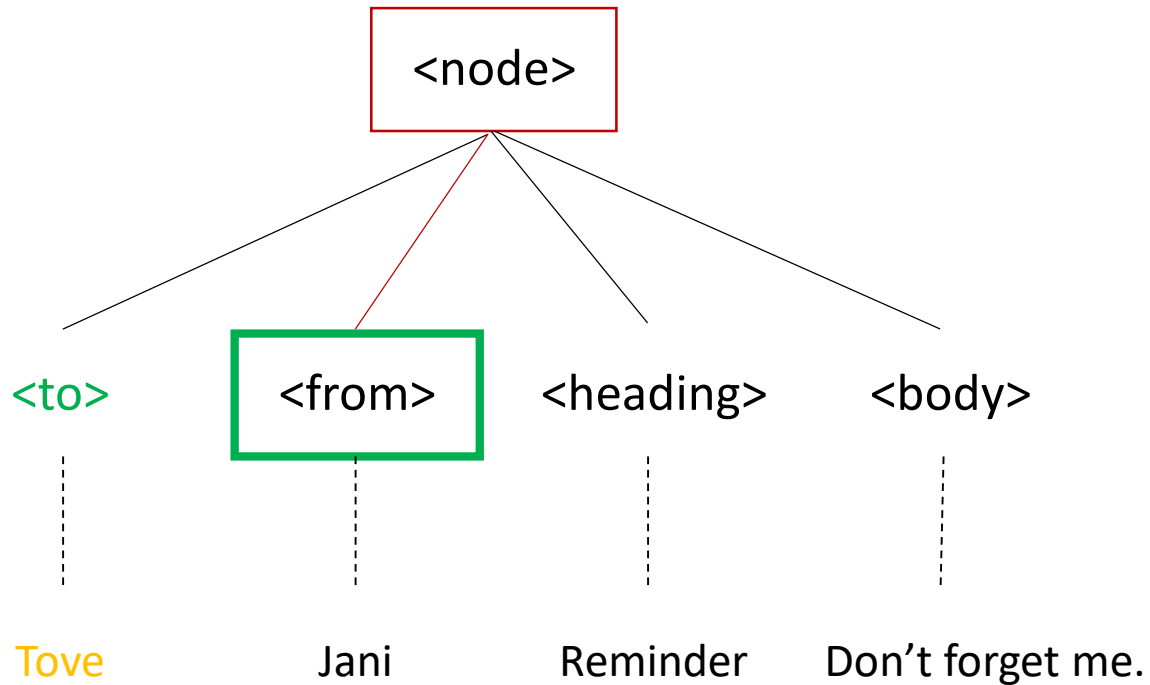


`<node><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>`



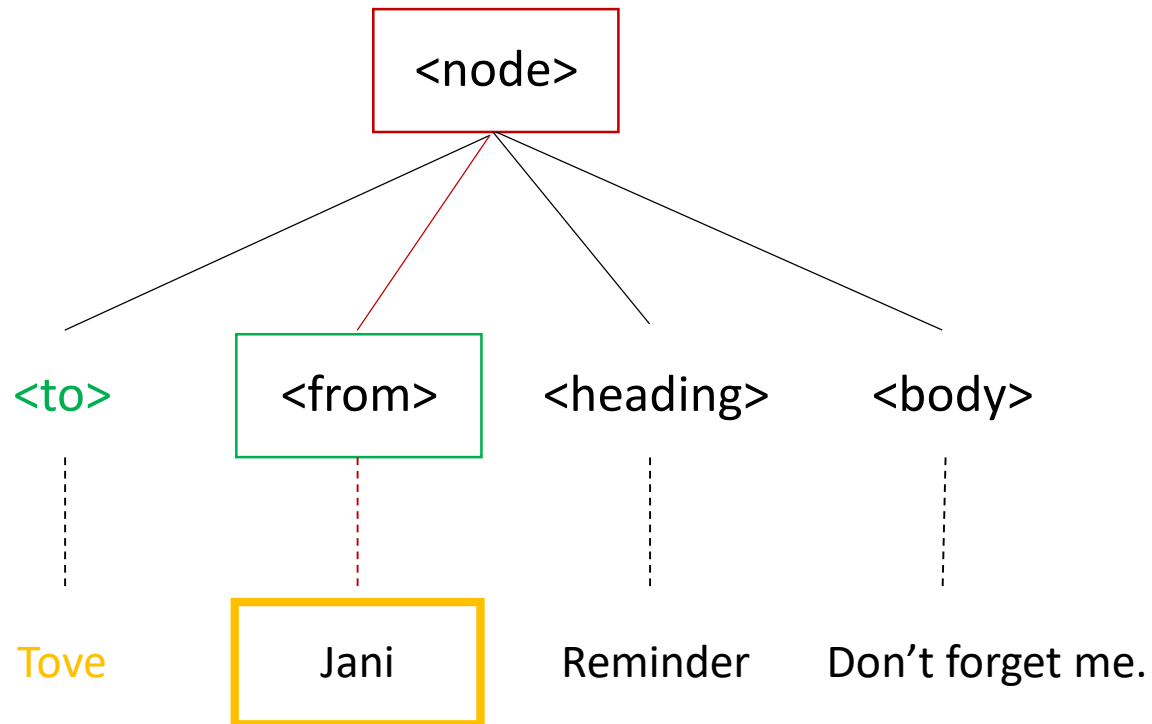


`<node><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>`



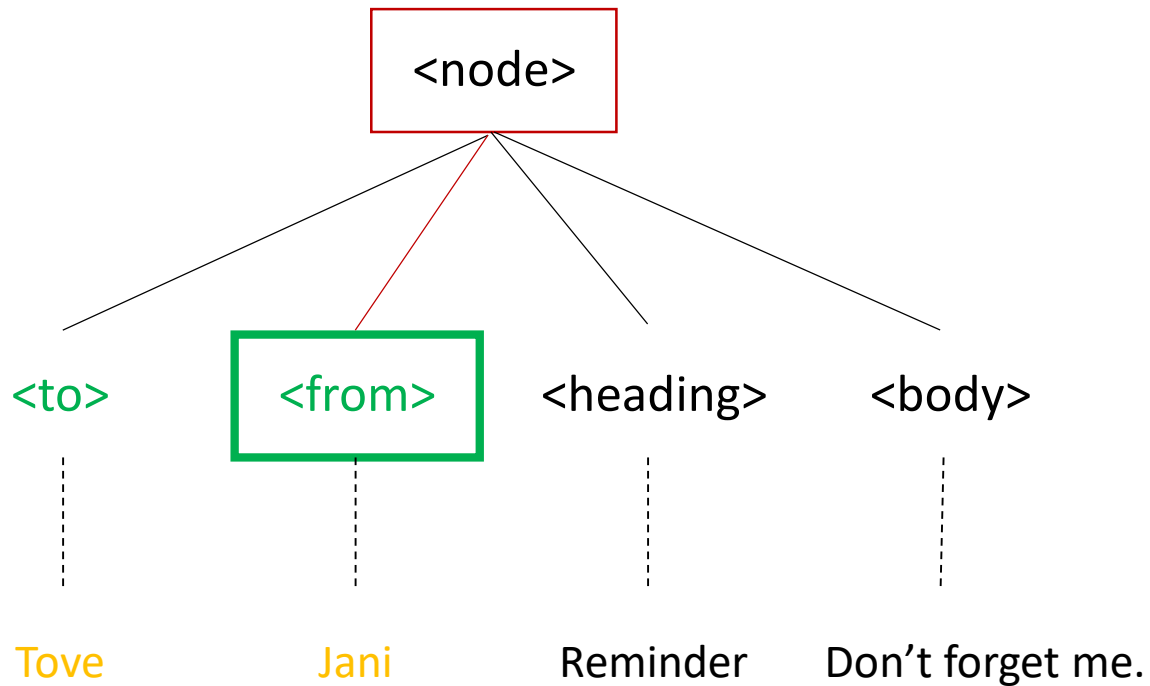


`<node><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>`



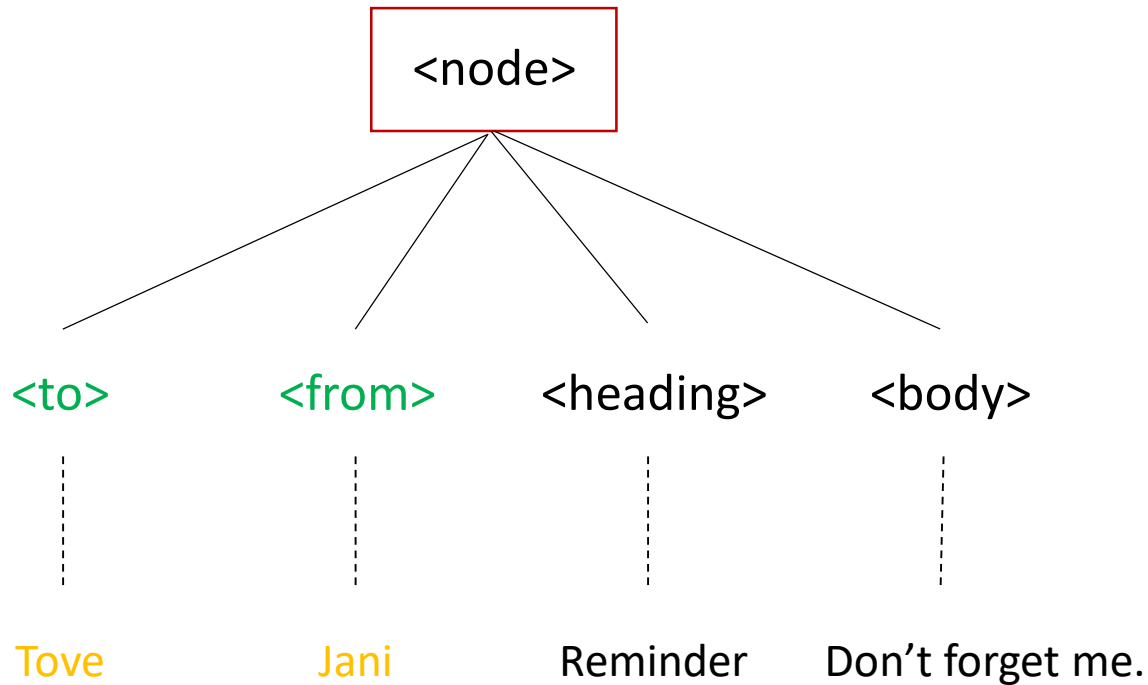


`<node><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>`



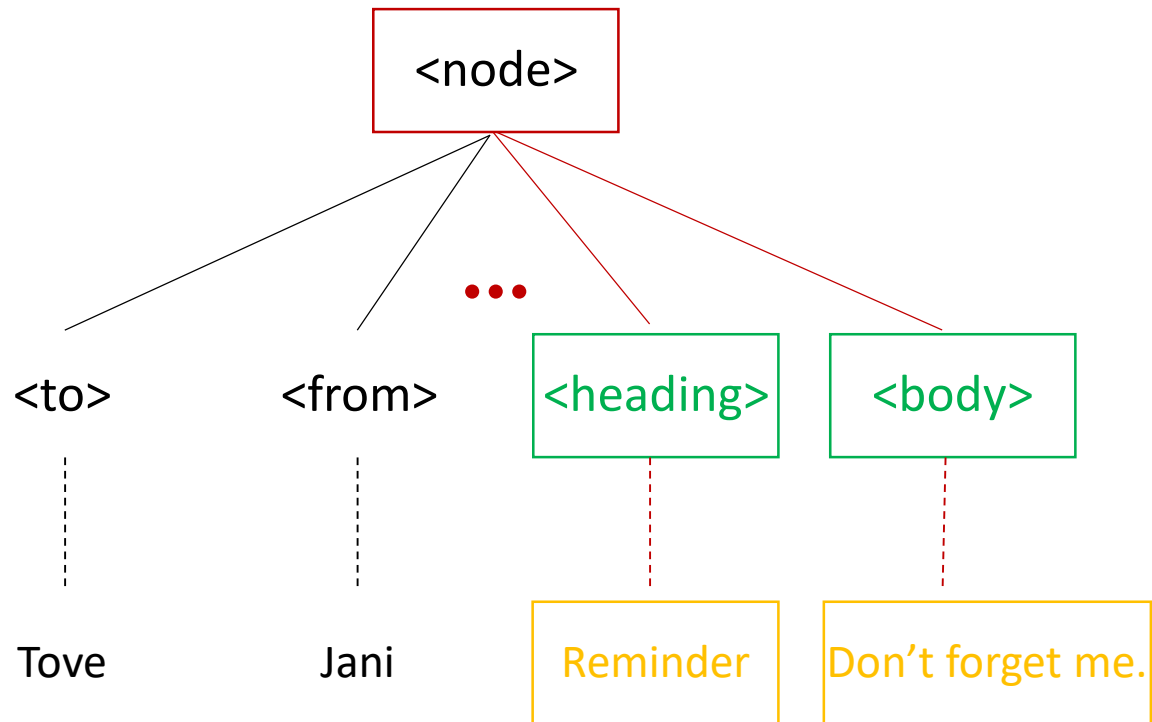


`<node><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>`



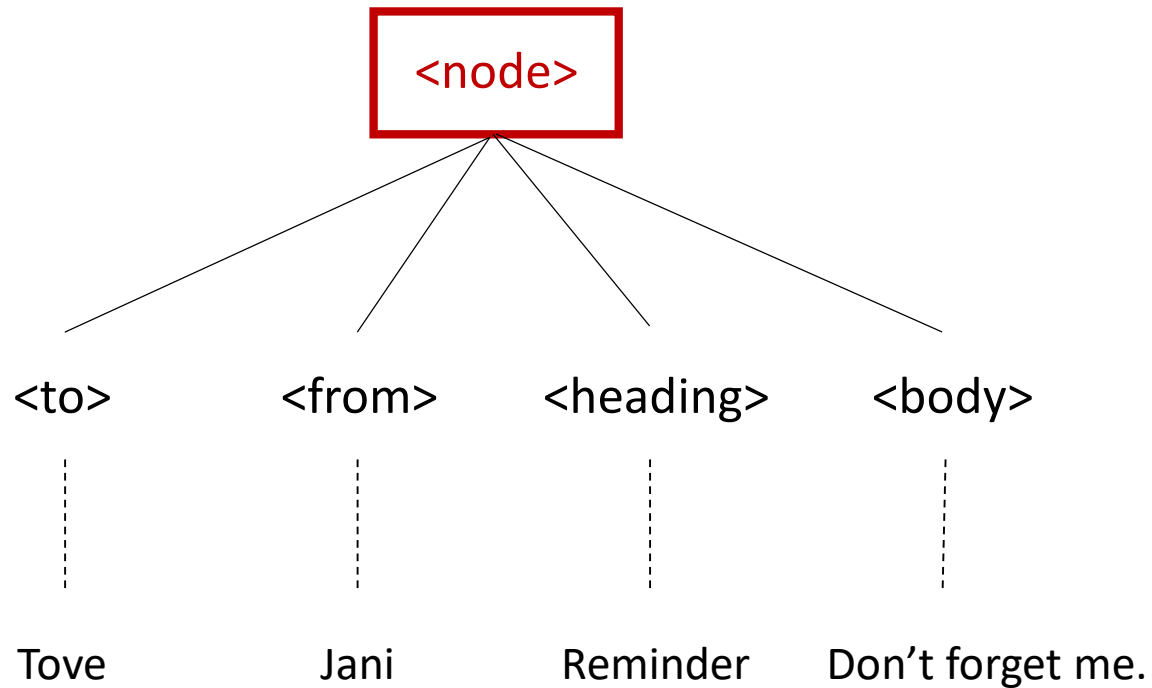


`<node><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>`





`<node><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me. </body></node>`



SAX	DOM
• <i>sequential access</i> („pergamentekercs”) • számon kell tartani, hogy hol tartunk, csak az aktuális elem érhető el közvetlenül • jó, ha a dokumentum kis része (egy elem) kell egyszerre	• <i>random access</i> („könyv”) • a teljes dokumentum rendelkezésre áll, tetszőleges eleme bármikor elérhető • jó, ha a dokumentum nagy része (az egész) kell egyszerre
• Az XML dokumentum <i>esemény-stream</i>. • meghívódó, <i>callback</i> metódusok • „<i>push</i>” típusú feldolgozás	• Az XML dokumentum <i>adatszerkezet</i>: • egy csomópontokból álló fa-struktúrában tárolódik. • „<i>pull</i>” típusú feldolgozás

SAX	DOM
- Az XML fájlt egyszer végigolvassa, <i>nem</i> tárolja. <i>Események</i>: nyitó címke, záró címke, karakteres adat. <i>Események kezelése</i>: egyedileg megírt, automatikusan	
- memóriaigény: kicsi (gyakorlatilag konstans) - nagy dokumentumok feldolgozására is alkalmas - bonyolult (mély) struktúrák kezelése nehéz <i>gyors</i>	- memóriaigény: $O(\text{csomópontok száma ill. dokumentum mérete})$ - csak kis dokumentumok kezelhetők vele - bonyolult (mély) struktúrák kezelése viszonylag egyszerű <i>lassú</i>

SAX	DOM
• Szöveges kimenetet könnyű előállítani. • XML dokumentum előállítása nehezebb.	• A fa-struktúrát tetszőlegesen módosíthatjuk. • A módosított DOM fa egyszerűen kiírható XML formába.

- `<i>dőlt` `<b>dőlt` és félkövér`</i>` félkövér`</b>`
- Példa:
 - Szavak és oldalak bejelölése egy dokumentumban.
 - Oldal alján elválasztott szó → átfedés!
- Példa
 - `<w>Kossuth</w>` `<w>utcai</w>` illetve
 - `<NE>Kossuth utca</NE>i`



1. lezárni és kinyitni újból

- `<i>dőlt </i><b><i>dőlt és félkövér</i></b><b>félkövér</b>`

2. attribútumban kódolni

- `<x ot="open:i,close:b"/>`

3. standoff annotáció

- `<s><w i="1">Miért</w> <w i="2">nem</w><w i="3">lesz</w></s>`
- `<strucs><struc beg="1" end="2"/><struc beg="2" end="3"/></strucs>`



Köszönöm a figyelmet



ध्यान देने के लिए आपका धन्यवाद
आप अपना ध्यान के लिए धन्यवाद
थैंक यू फॉर योर अटेंशन

Thank you for your attention
Thank you for your attention
Thank you for the attention

Merci de votre attention
Je vous remercie pour votre attention
Merci pour l'attention

感谢您的关注
谢谢你的关注
谢谢您的注意