

HARSÁNYI ZOLTÁN

XHGMYE

2021.01.26

① DOM

## Document Object Model

A DOM API egy programozott API a memóriában  
A SAX-nál több lehetőséget ad a módosításra  
Elem csomópont, szülőes, csomópontok hálójában  
Leírások, kérés, módosítás

Elemek:

Document

Node

Attribute

CharacterData

Element

NodeList

DomException

HasChildNodes adathoz

Document: documentElement: a legfelső elem

Node: removeChild()  
appendChild()  
replaceChild()  
insertBefore

NodeType: nodeName  
nodeValue  
childNodes  
parentNode  
nextSibling  
textContent

Element: createElement()  
createTextNode()  
createAttribute()  
getElementsByTagName()

HARSÁNYI ZOLTÁN

XKGTME

2021.01.26

(2.)

## XPATH

Az XML Dokumentumot jól ismert, hasonló a DOM-hoz.

Az XML jól az elemek hierarchiájára használható.

XSLT használja

SQL beépített függvény

Kifejezés: Milyenval történik, eseménypontok amelyek eléréséhez  
a mintára azok vannak kiíratva

operátorok: + - > \* div !

!= < <= >= > and or not

tengelyek: self : maga a csomópont

child: a gyermek csomópontok

descendant: a gyermek és azok gyermekei tetszőleges mélyséig

descendant-or-self: descendant és self node egyikén

ancestor: összes csomópont amely befoglalja a node-t

ancestor-or-self: ancestor és self node egyikén

preceding: azok a csomópontok amelyek előtt vannak a dokumentumban

preceding-sibling: preceding node-ok hibérve az ancestor befoglaló elemi node-okkal

following: dokumentumban következzen ezt követően

following-sibling: following hibérve az ancestor, befoglaló elemi elemekkel

~~attribute~~ attribute: a kontextus csomópontján tartozó elem

namespace: a kontextusban tartozó elemek

HARSÁNYI ZOLTÁN  
XK9HYE  
2021.01.26

- ③ XML Schema alapján validálhatunk XML dokumentumokat  
Szorapra hogy ~~meta~~ megadja hogyan nézhet ki az XML dokumentum. Korlátozza a tartalt előfordulások hirt, előfordulások számát, helyességét. Integritás <sup>örv</sup>

hírlön függben adjuk meg majd az xml-be amit validálunk megadjuk elírás: utvondit

```
<?xml version="1.0" ?>  
<xs:schema xmlns:xs=" " ?>  
</xs:schema>
```

Elem: <xs:element elemnév>  
tartalom  
</xs:element>

Összetett:  
<xs:complexType>  
<xs:sequence>  
...  
</xs:sequence>  
</xs:complexType>

xs:sequence → sorrendben  
tetszőleges sorrend esetén xs:all  
opcionális lista xs:choice  
is element ~~xs~~: nem fontos megadni

Névtér: feldolgozó program azonosítójá

2021.09.16

## ④ JSON

JavaScript Object Notation

Legelterjedtebb adat formátum az interneten

Hozható egy JS objektumhoz kevés eltéréssel  
a key-ek "..."-ban vannak

key: value alapú

adattípusok: {} - object

[] - array

" " - string

~~not~~ number - float, int.

leírás: képesek zárgatni nem

```
{
  "key1": "value1",
  "key2": {
    "key3": "val3",
    "key4": "val4"
  },
  "key5": [1, 2, 3, 4, 5]
}
```

}

elem olvasás: képesek zárgatni akár pl. json["key2"]["key3"]

YAML: JSON alapján készült, de ehhez  
whitespace jelöl a scope-ot  
nyilván az adattípusok vannakXML vs JSON: XML: hierarchikus  
fogalmat használ

JSON: rövidebb, gyorsabban  
könnyebben olvasható  
könnyebb használni az interneten  
könnyebb feldolgozni parserrel