WEBFEJLESZTÉS 2. – ŰRLAPOK, KÉPEK, TÁBLÁZATOK, BÖNGÉSZŐ

Horváth Győző

Egyetemi adjunktus

1117 Budapest,
Pázmány Péter sétány 1/C, 2.420
Tel: (1) 372-2500/1816

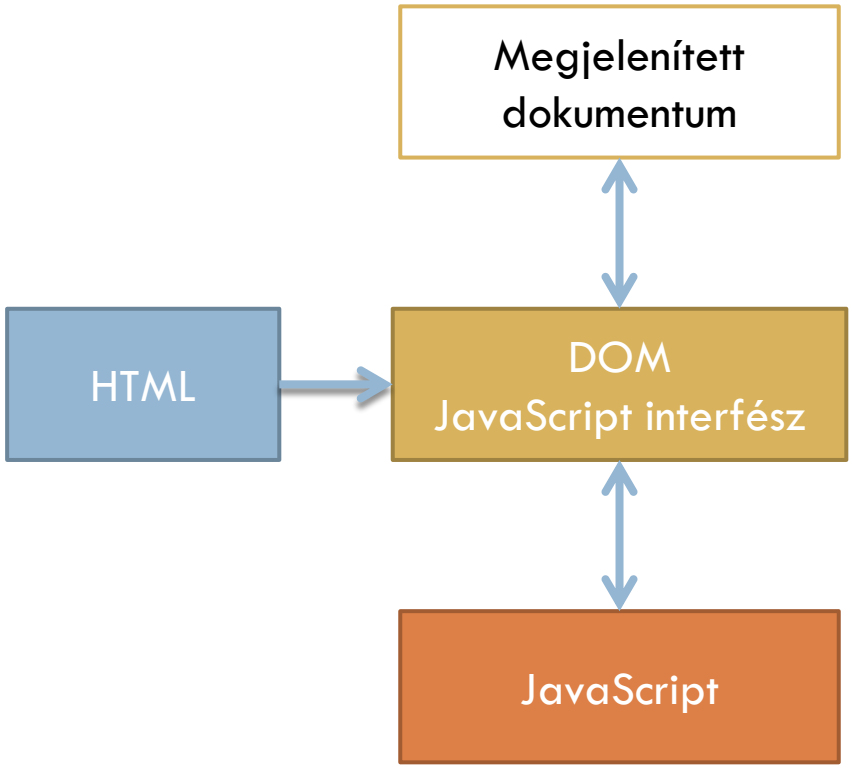
Ismétlés

JavaScript – programozás

- JavaScript – C++
- Vezérlési szerkezetek
- Operátorok
- Függvények
- Adatszerkezetek
  - Elemi típusok
  - Összetett típusok: tömb, objektum → JSON

HTML és JavaScript – webprogramozás

- Elemek elérése
  - document.getElementById
  - $()
- Elemek tulajdonságai
- Eseménykezelés
  - addEventListener
- HTML és JavaScript szétválasztása



Események, nyelvi elemek

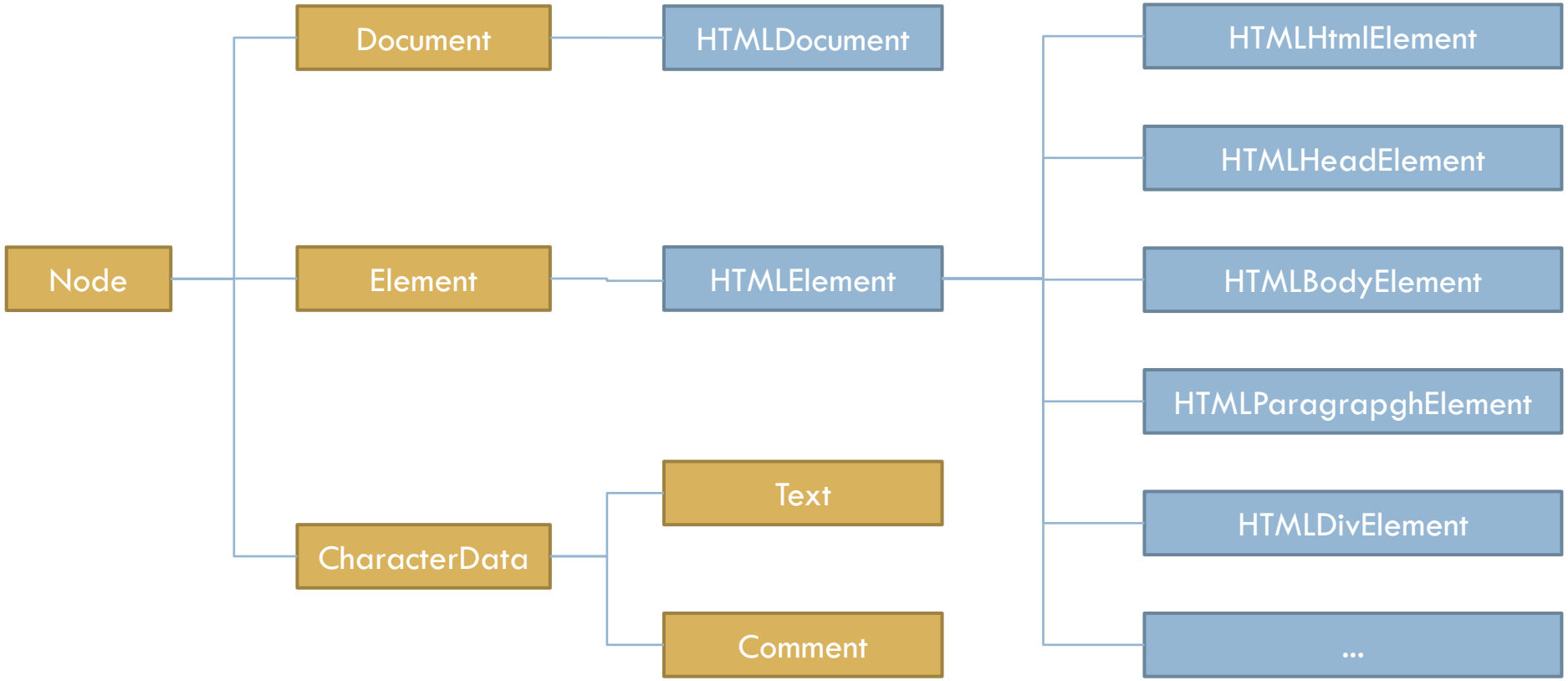
- Eseménykezelés részletei
  - Eseményt jelző objektum
    - this
  - Eseményobjektum
    - megszerzése
    - tulajdonságai
  - Alapértelmezett művelet megakadályozása
  - Események buborékolása
  - Eseményt eredetileg jelző objektum
  - Buborékolás megállítása
- Globális függvények és értékek
  - NaN, isNaN(), Infinity, isFinite()
  - parseInt(), parseFloat()
  - encodeURI(), decodeURI()
- Beépített objektumok
  - Math
  - Date
  - String
  - Array

DOM műveletek

DOM-ok

- DOM Core
  - csomópontok fája
  - ábrázolás
  - művelet
- HTML DOM
  - HTML elemek fája
  - speciális elemek → többletfunkcionalitás
- Speciális elemek
  - Űrlapok
  - Képek
  - Táblázatok

DOM-ok


Node
Document
Element
CharacterData
HTMLDocument
HTMLElement
Text
Comment
HTMLHtmlElement
HTMLHeadElement
HTMLBodyElement
HTMLParagrapghElement
HTMLDivElement
...

DOM

- Document Object Model
- HTML és XML dokumentumok programozási felülete
- Objektumstruktúra, faszerkezet
- DOM csomópontok
  - dokumentum
  - elem
  - attribútum
  - szöveges csomópont

Példa


```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <p id="par1">Bekezdés.</p>
    <p class="aktiv">Még egy bekezdés.</p>
    <ul>
      <li>első</li>
      <li class="aktiv">második</li>
    </ul>
    <form name="form1">
      <input type="radio" name="r1" value="elso">
      <input type="radio" name="r1" value="masodik">
    </form>
  </body>
</html>
```

Példa DOM fája

Műveletcsoportok

- Elemek kiválasztása
- Szerkezet bejárása
- Szerkezet módosítása
  - attribútumok
  - új elem/attribútum létrehozása
  - módosítás
  - törlés

Elemek kiválasztása

- document
  - getElementById(id)
  - getElementsByName(név)
- document/elem
  - getElementsByTagName(elemnév)
  - getElementsByClassName(stílusosztályok)
  - querySelector(css_szelektor)
  - querySelectorAll(css_szelektor)

```
console.log(document.getElementById('par1'));
console.log(document.getElementsByName('r1'));
console.log(document.getElementsByTagName('p'));
console.log(document.getElementsByClassName('aktiv'));
console.log(document.querySelector('ul > li'));
console.log(document.querySelectorAll('ul > li'));

var body = document.body;
console.log(body.getElementsByTagName('p'));
console.log(body.getElementsByClassName('aktiv'));
console.log(body.querySelector('ul > li'));
console.log(body.querySelectorAll('ul > li'));
```

```
<body>
  <p id="par1">Bekezdés.</p>
  <p class="aktiv">Még egy bekezdés.</p>
  <ul>
    <li>első</li>
    <li class="aktiv">második</li>
  </ul>
  <form name="form1">
    <input type="radio" name="r1" value="elso">
    <input type="radio" name="r1" value="masodik">
  </form>
</body>
```

Szerkezet bejárása

- Bejárás (elem.tulajdonság)
  - gyerekek
    - childNodes, firstChild, lastChild
  - szülő
    - parentNode
  - testvérek
    - nextSibling, previousSibling
- Szöveges csomópontok is megjelennek
  - nodeType === 3

Szerkezet módosítása

- Attribútumok (elem)
  - getAttribute(név), setAttribute(név, érték), hasAttribute(név), removeAttribute(név)
- Törlés, beszúrás (elem)
  - removeChild(elem), appendChild(elem), replaceChild(elem, elem)
- Létrehozás
  - document.createElement(elem)
  - (elem.innerHTML)

```
var p = document.createElement('p');
p.innerHTML = 'Új bekezdés';
p.setAttribute('class', 'aktiv');
document.body.appendChild(p);
```

Űrlap és űrlapelemek

Tipikus űrlapműveletek

- Interakció egyik legfontosabb eszköze az űrlap
- Adatok beolvasása, megadása
- Adatok ellenőrzése

- Referencia
  - https://developer.mozilla.org/en-US/docs/Gecko_DOM_Reference

Űrlap tulajdonságai, metódusai, eseményei

- <form> elem
- Tulajdonságai (analóg módszer)
  - action
  - method
  - enctype
  - name
  - target
- Események
  - submit (megállítható)
  - reset
- Metódusok
  - submit()
  - reset()

Input elem

- <input>
- type attribútuma → jellege
- Közös tulajdonságai
  - name
  - form
  - value
  - defaultValue
  - disabled
  - tablndex
- Közös metódusok
  - focus()
  - blur()
  - select()
  - click()
- Események
  - click
  - change
  - focus
  - blur

Szöveges beviteli mező(k)

- type
  - text
  - password
- Tulajdonságok
  - value
  - readOnly
  - size
  - maxLength

Checkbox

- <input type="checkbox">
- Jelölőmező
- Legfontosabb tulajdonság: checked
  - írható, olvasható logikai érték
  - defaultChecked

Radio gombok

- <input type="radio">
- name attribútum fogja össze őket
- Legfontosabb tulajdonsága: checked
  - írható, olvasható logikai érték
  - defaultChecked
- Melyik van kiválasztva?
  - id alapján nehézkes
  - → name alapján kell

Radio gombok


```
<form id="form1">
  <input type="radio" name="alma" value="piros">
  <input type="radio" name="alma" value="sárga">
  <input type="radio" name="alma" value="zöld" checked>
  <input type="radio" name="alma" value="barna">
</form>
```

Radio gombok

- Name alapján
  - document.getElementsByName(nev)

```
function radioErteke(nev) {
  var radiok = document.getElementsByName(nev);
  var ertek;
  var i = 0;
  while (i < radiok.length && !radiok[i].checked) {
    i = i + 1;
  }
  if (i < radiok.length) {
    ertek = radiok[i].value;
  }
  return ertek;
}
```

Radio gombok

- Name alapján
  - document.getElementsByName(nev)

```
function radioErteke(nev) {
  var radiok = document.getElementsByName(nev);
  for (var i = 0; i < radiok.length; i++) {
    if (radiok[i].checked) {
      return radiok[i].value;
    }
  }
  return undefined;
}

//Használata
radioErteke('alma');  //"zöld"
```

Radio gombok

- Name alapján
  - document.getElementsByTagName(nev)
  - elem.getElementsByTagName(nev)

```
function radioErteke(nev, urlapAzon) {
  var tarto = urlapAzon?document.getElementById(urlapAzon):document;
  var inputok = tarto.getElementsByTagName('input');
  var ertek;
  var van = false;
  var i = 0;
  while (!van && i < inputok.length) {
    van = inputok[i].name == nev && inputok[i].checked;
    ertek = inputok[i].value;
    i = i + 1;
  }
  return ertek;
}
```

Legördülő mező

- <select> elem
- Tulajdonságai
  - multiple
  - size
  - value
  - selectedIndex
  - options – tömb
    - options.length
- options[i]
  - options[i].value
  - options[i].text
  - options[i].selected
- Metódusok
  - add(elem[, elemElé])
  - remove(index)

Legördülő mező


```
<select id="select1">
  <option value="ertek1">szöveg1</option>
  <option value="ertek2">szöveg2</option>
  <option value="ertek3" selected>szöveg3</option>
  <option value="ertek4">szöveg4</option>
</select>
```

Legördülő mező

- Kiválasztott opció indexe, értéke, szövege

```
var select = document.getElementById('select1');
select.value;                                    //"ertek3"
select.selectedIndex;                            //2
select.options[select.selectedIndex].value;      //"ertek3"
select.options[select.selectedIndex].text;       //"szöveg3"
```

Legördülő mező

- Kiválasztott opció adatai

```
function kivalasztottOpcio(select) {
  if (select && select.selectedIndex > -1) {
    return {
      value: select.value,
      text:  select.options[select.selectedIndex].text,
      index: select.selectedIndex
    };
  }
  return {};
}

//Használata
var select = document.getElementById('select1');
kivalasztottOpcio(select);
//Object {value: "ertek3", text: "szöveg3", index: 2}
```

Legördülő mező

- Új opció hozzáadása

```
//Új elem hozzáadása DOM műveletekkel
function ujOpcio(select, szoveg, ertek) {
  var ujop = document.createElement('option');
  ujop.value = ertek;
  ujop.text = szoveg;
  select.add(ujop, null);
}
//vagy
//Új elem hozzáadása innerHTML segítségével
function ujOpcio(select, szoveg, ertek) {
  var ujop = '<option value="' + ertek + '">' + szoveg + '</option>';
  select.innerHTML += ujop;
}
//Használata
var select = document.getElementById('select1');
ujOpcio(select, 'új szöveg', 'ujErtek');
```

Legördülő mező

- Opció törlése

```
function torolOpcio(select, i) {
  select.remove(i);
}

//Használata
var select = document.getElementById('select1');
torolOpcio(select, 3);
```

Legördülő mező

- Többszörös választás esetén

```
<select id="select1" multiple size="4">
  <option value="ertek1" selected>szöveg1</option>
  <option value="ertek2">szöveg2</option>
  <option value="ertek3" selected>szöveg3</option>
  <option value=
</select>
```

```
function kivalasztottOpciok(select) {
  var ertekek = [];
  for (var i = 0; i < select.options.length; i++) {
    if (select.options[i].selected) {
      ertekek.push(select.options[i].value);
    }
  }
  return ertekek;
}

//Használata
var select = document.getElementById('select1');
kivalasztottOpciok(select);    //["ertek1", "ertek3"]
```

Többsoros szöveges beviteli mező

- <textarea></textarea>
- Tulajdonságok
  - value
  - defaultValue

Képek

Képek

- <img src="kep.png" alt="szöveg" >
- Legfontosabb tulajdonsága: src
  - Dinamikusan lecserélni a képet
- Minden egyéb általában stílusmanipuláció
- Tipikus képműveletek
  - Kép lecserélése
  - Kép előtöltése

Kép lecserélése

- src attribútumának megváltoztatása

```
<img src="alma.png" alt="Alma" id="kep1">
```

```
var kep = document.getElementById('kep1');
kep.src = 'korte.png';
```

- Hátránya:
  - a kép letöltése a szerverről ekkor kezdődik el
  - lassú lehet
  - felhasználói élmény csökken
- → Kép előtöltése

Kép előtöltése

- Egy memóriabeli kép objektumba előre betöltjük a képet, így cserekor az azonnal rendelkezésre áll.
- Sok képet nem érdemes előre betölteni.
- Inkább folyamatosan töltsük be őket.

```
var mem_kep = document.createElement('img');
mem_kep.src = 'korte.png';
//...
//ha szukseges a csere:
var kep = document.getElementById('kep1');
kep.src = mem_kep.src;
```

Táblázatok

Sorok és cellák kezelése

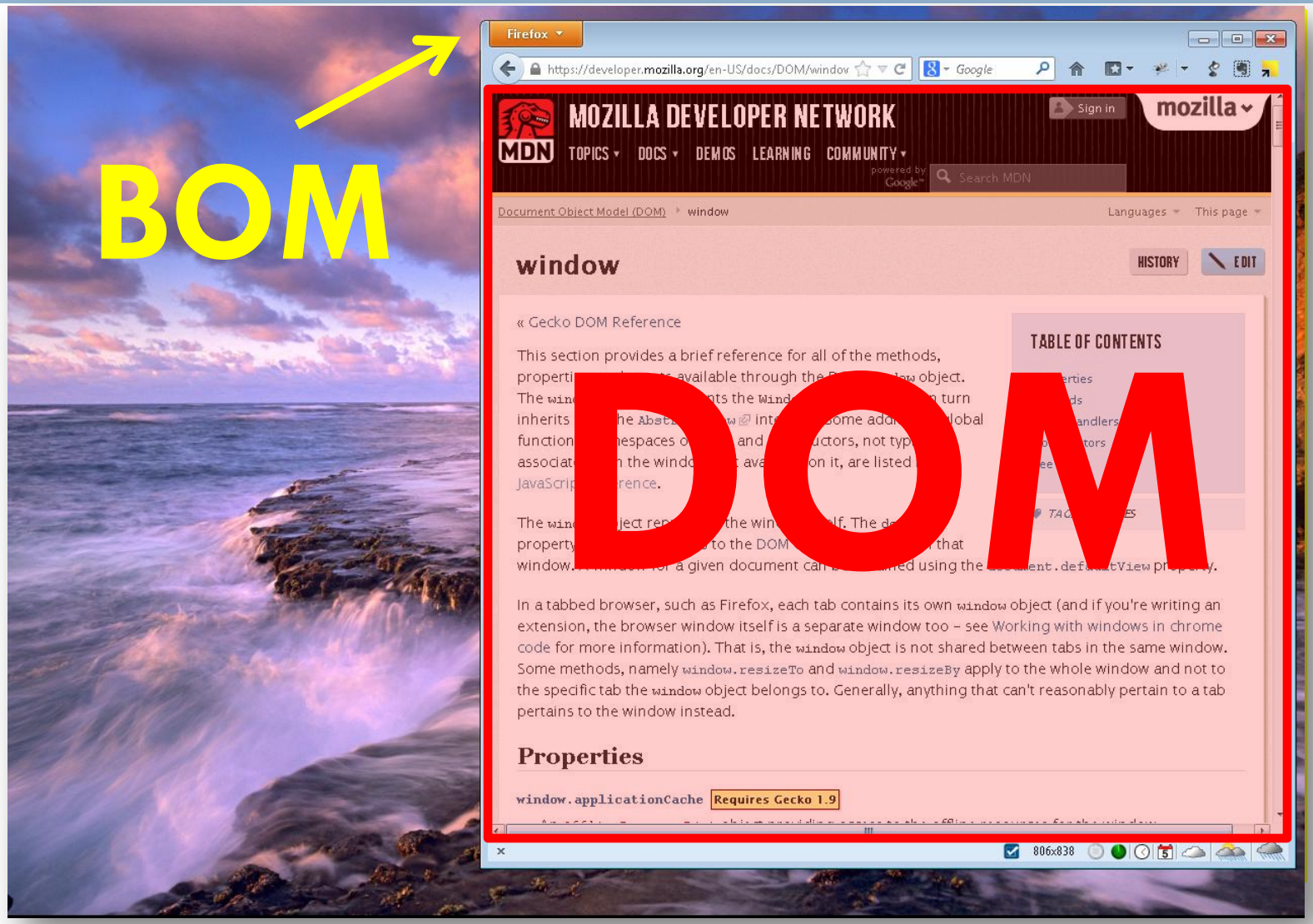
- Táblázat
  - rows
  - insertRow(index)
  - deleteRow(index)
- Sor
  - rowIndex
  - sectionRowIndex
  - cells
  - insertCell(index)
  - deleteCell(index)
- Cella
  - cellIndex
- x-y koordináta:

```
function xyKoord(td) {
  var x =  td.cellIndex;
  var tr = td.parentNode;
  var y =  tr.sectionRowIndex;
  return {
    x: x,
    y: y
  };
}
```

BOM – Browser Object Model

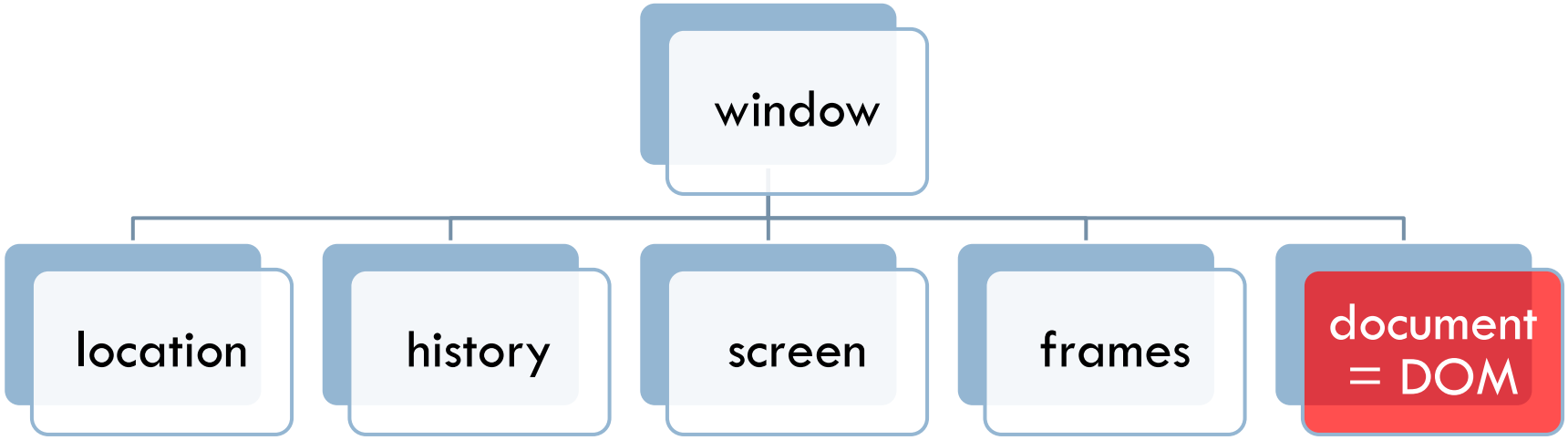
A böngészővel kapcsolatos objektumok, metódusok

A böngészőablak



Browser Object Model – BOM

- BOM: a böngészőablaknak megfelelő objektumhierarchia
- https://developer.mozilla.org/en-US/docs/DOM/window



Méretek és pozíció

- window
  - innerHeight, innerWidth
    - a tartalmi rész magassága és szélessége
  - outerHeight, outerWidth
    - a böngészőablak magassága és szélessége
  - screenX, screenY
    - az ablaknak a főképernyőtől bal felső sarkától való távolsága
  - scrollX, scrollY
    - hány pixelnyire van gördítve a dokumentum
  - scrollMaxX, scrollMaxY
    - legfeljebb hány pixelnyit lehet gördíteni a dokumentumot

Méretek és pozíció

- window.screen (az ablakhoz kapcsolódó képernyő)
  - width, height
    - a képernyő szélessége és magassága
  - availWidth, availHeight
    - a képernyő rendelkezésre álló szélessége és magassága
  - top, left, availTop, availLeft, colorDepth, pixelDepth
    - ld. dokumentáció

Méretek és pozíció

- window metódusok
  - resizeTo(szél, mag), resizeBy(dszél, dmag)
    - ablak átméretezése (megkötésekkel)
  - moveTo(x, y), moveBy(dx, dy)
    - ablak áthelyezése (megkötésekkel)
  - scrollTo(x, y), scrollBy(dx, dy), scrollByLines(l), scrollByPages(p)
    - a dokumentum görgetése adott pozícióra vagy valamennyivel

Új ablak

- window.open

```
var w = window.open(url, név, tulajdonságok);
```

- Paraméterek
  - url: a betöltendő oldal URL-je
    - ha üres → üres ablak
  - név: az új ablak neve
    - ha létezik → ebbe töltődik bele az url
    - "_blank" → új ablak nyílik
  - tulajdonságok: ablaknyitási paraméterek

Új ablak

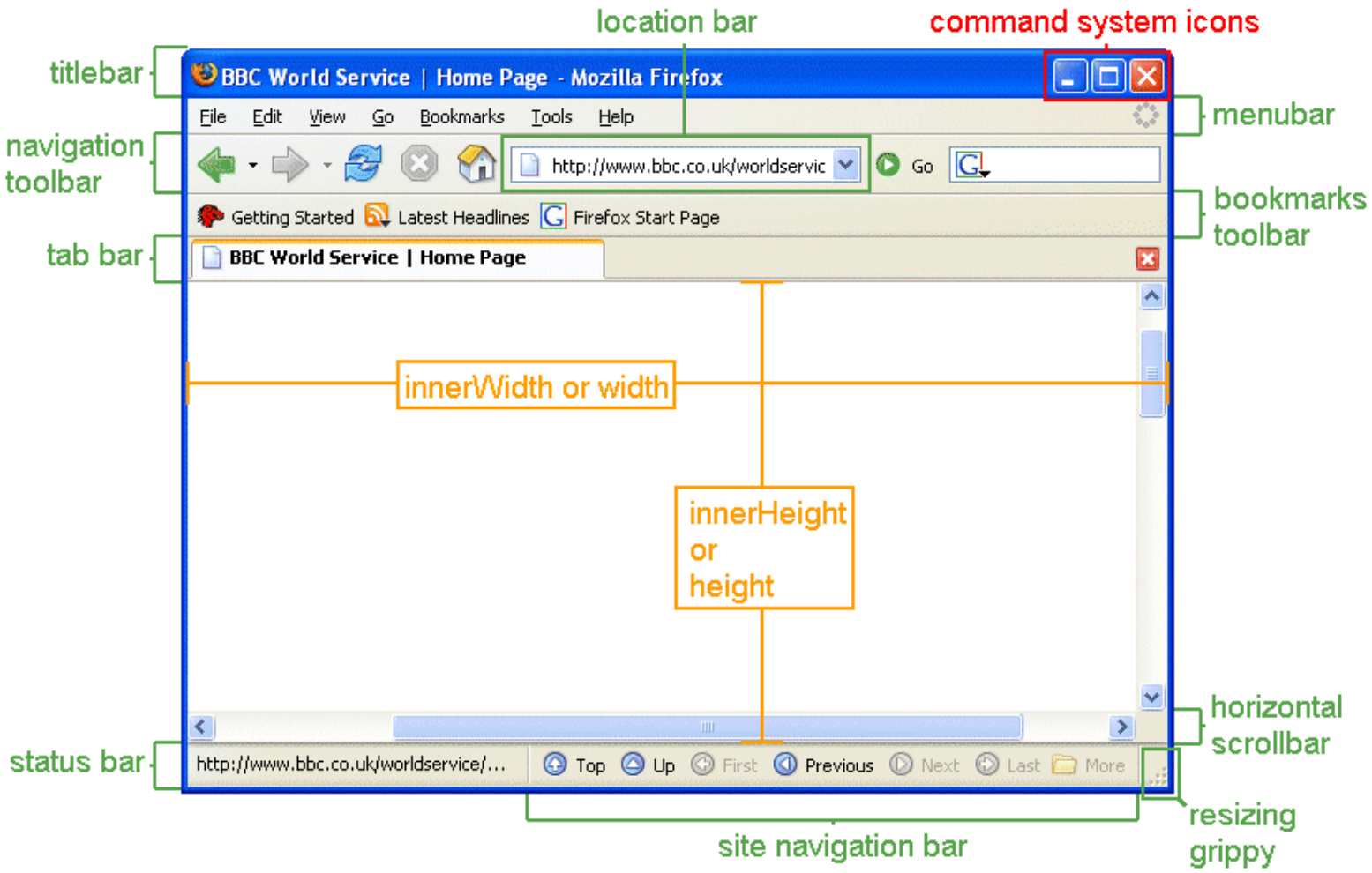
- Ablaknyitási paraméterek
  - https://developer.mozilla.org/en-US/docs/DOM/window.open
  - `"width=800,height=600,scrollbars=yes"`
  - ha üres → új fül vagy alapértelmezett értékek
  - ha nem üres → akkor a nem megadott tulajdonságok automatikusan kikapcsoltak lesznek

```
var w = window.open(
        'http://www.elte.hu',
        'ELTEablak',
        'width=800,height=600,scrollbars=yes'
);
```

Új ablak

- Ablaknyitási paraméterek
  - top, left
  - width, height
  - menubar, toolbar, location, status
  - scrollbars

Új ablak


location bar
command system icons
titlebar
BBC World Service | Home Page - Mozilla Firefox
menubar
File Edit View Go Bookmarks Tools Help
navigation toolbar
http://www.bbc.co.uk/worldservic
Go
bookmarks toolbar
Getting Started
Latest Headlines
Firefox Start Page
tab bar
BBC World Service | Home Page
innerWidth or width
innerHeight or height
horizontal scrollbar
status bar
http://www.bbc.co.uk/worldservice/...
Top
Up
First
Previous
Next
Last
More
resizing grippy
site navigation bar

Új ablak

- A window.open() által visszaadott objektum az új ablak window objektuma

```
var w = window.open(url, név, tulajdonságok);
```

- w mindent tud, amit a window
  - w.document, w.document.writeln()
  - w.resizeTo(szél, mag)
  - stb, stb, stb.

Új ablak

- Szülő ablak elérése a gyerekablakból

```
w.opener
```

- Gyerekablak elérése a szülőablakból
  - megnyitáskor visszaadott változó tárolása (w)
  - hivatkozás újbóli megszerzése

```
var w = window.open('', létező_név);
```

Oldal URL-je

- window.location
- A betöltött oldal URL-jével kapcsolatos információk
  - hash, host, hostname, href, origin, pathname, port, protocol, search

```
http://webprogramozas.inf.elte.hu:8080/webfejl2.html?alma=piros#valami

var l = window.location;
l.hash;      //"#valami„
l.host;      //"webprogramozas.inf.elte.hu:8080„
l.hostname;  //"webprogramozas.inf.elte.hu„
l.href;      //az egész URL
l.origin;    //"http://webprogramozas.inf.elte.hu:8080"
l.pathname;  //"/webfejl2.html„
l.port;      //"8080„
l.protocol;  //"http:„
l.search;    //"?alma=piros"
```

Oldal URL-je

- window.location szöveget kapva értékül megpróbálja a szövegként megadott oldalt betölteni
  - assign(url), replace(url), reload()

```
window.location = "http://www.elte.hu";
window.location.href = "http://www.elte.hu";
window.location.assign("http://www.elte.hu");
window.location.replace("http://www.elte.hu");
window.location.reload();
```

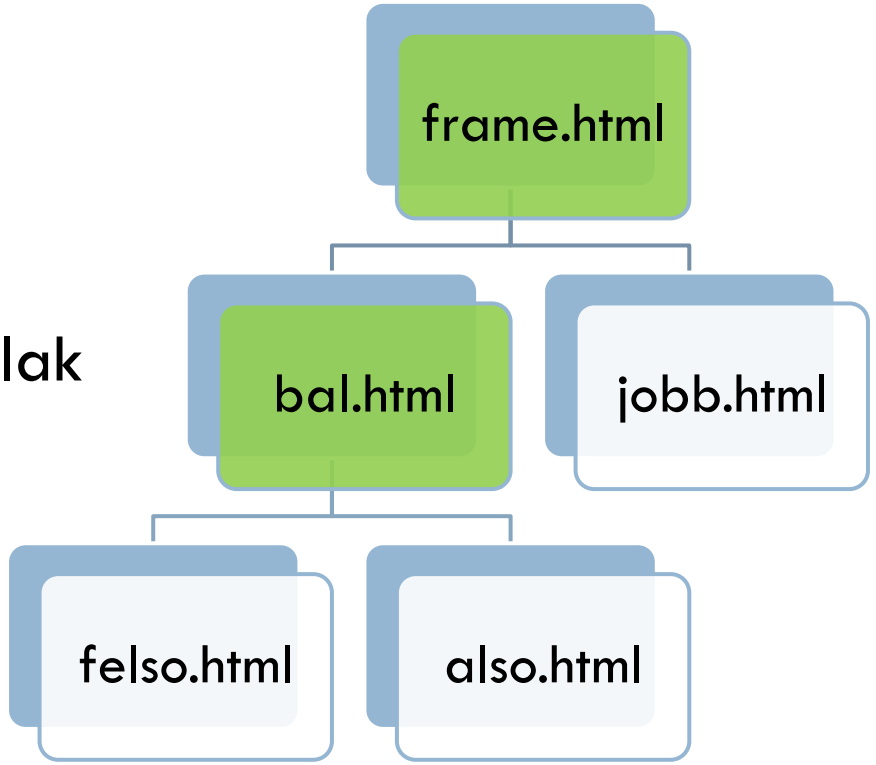
Előzmények

- window.history
  - back()
  - forward()
  - go(n)

```
window.history.back();
window.history.forward();
window.history.go(-3);
```

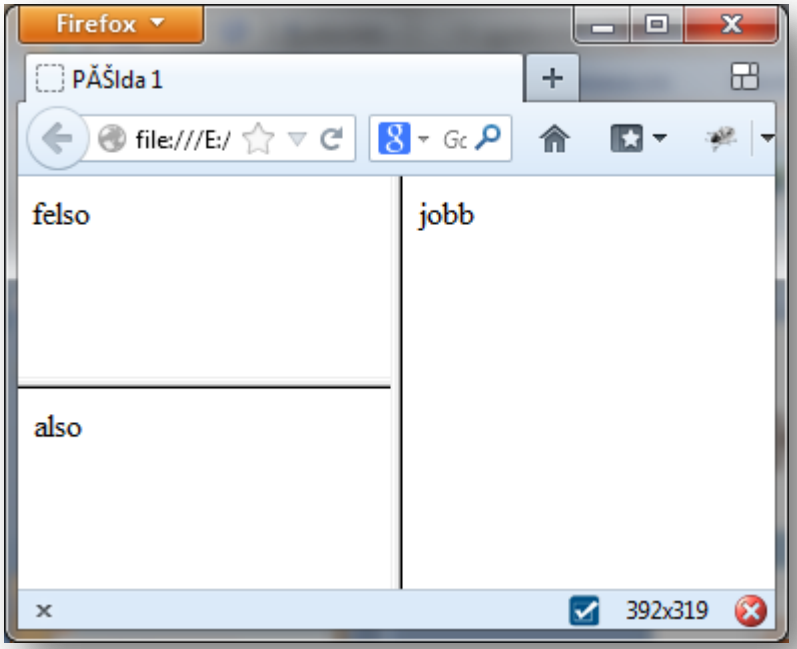
Keretek

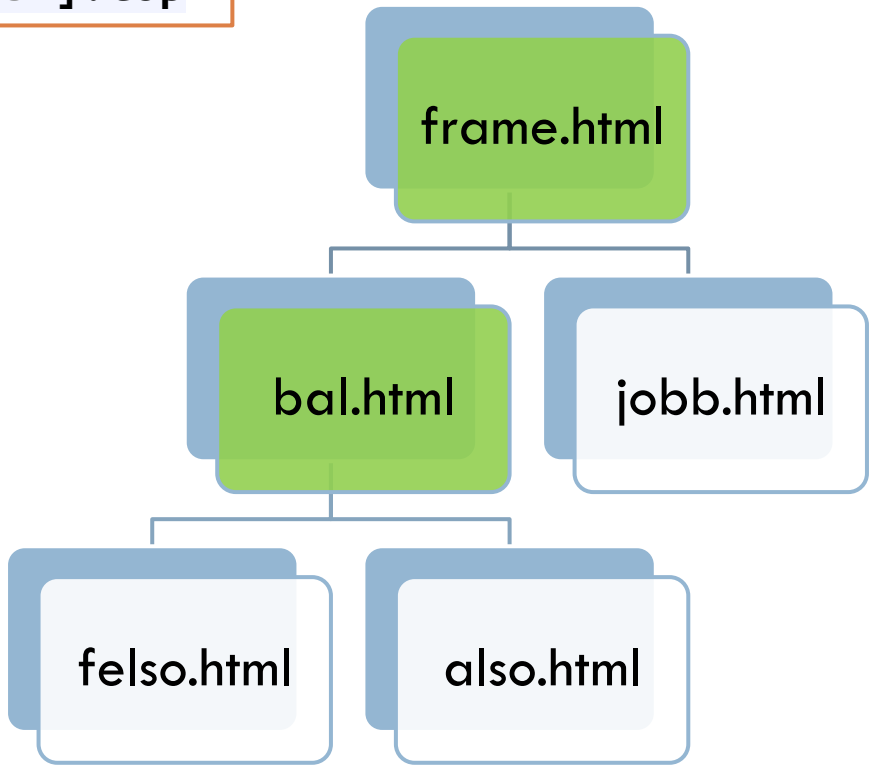
- Keretek: több HTML oldal → több window objektum
- window.frames
  - Adott ablak alá közvetlen tartozó ablakok
  - window.frames[0]
  - window.frames['nev']
- window.parent
  - A közvetlen felette lévő ablak
- window.top
  - A legfelső ablak



Keretek


```
window.frames.length;
window.frames[0];
window.frames['bal'];
window.frames['bal'].frames['felso']
window.frames['bal'].frames['felso'].top
```





További window tulajdonságok

- Tulajdonság
  - name
  - navigator
  - status
- Események
  - onload, onunload
  - onabort , onclose
  - oncontextmenu
  - onresize, onscroll, onselect
- Metódusok
  - alert, confirm, prompt
  - setTimeout, setInterval
  - clearTimeout, clearInterval
  - close
  - print

Összefoglalás

- DOM
  - faszerkezet
  - elérés, bejárás, módosítás
- HTML DOM
  - űrlapok
  - képek
  - táblázatok
- BOM