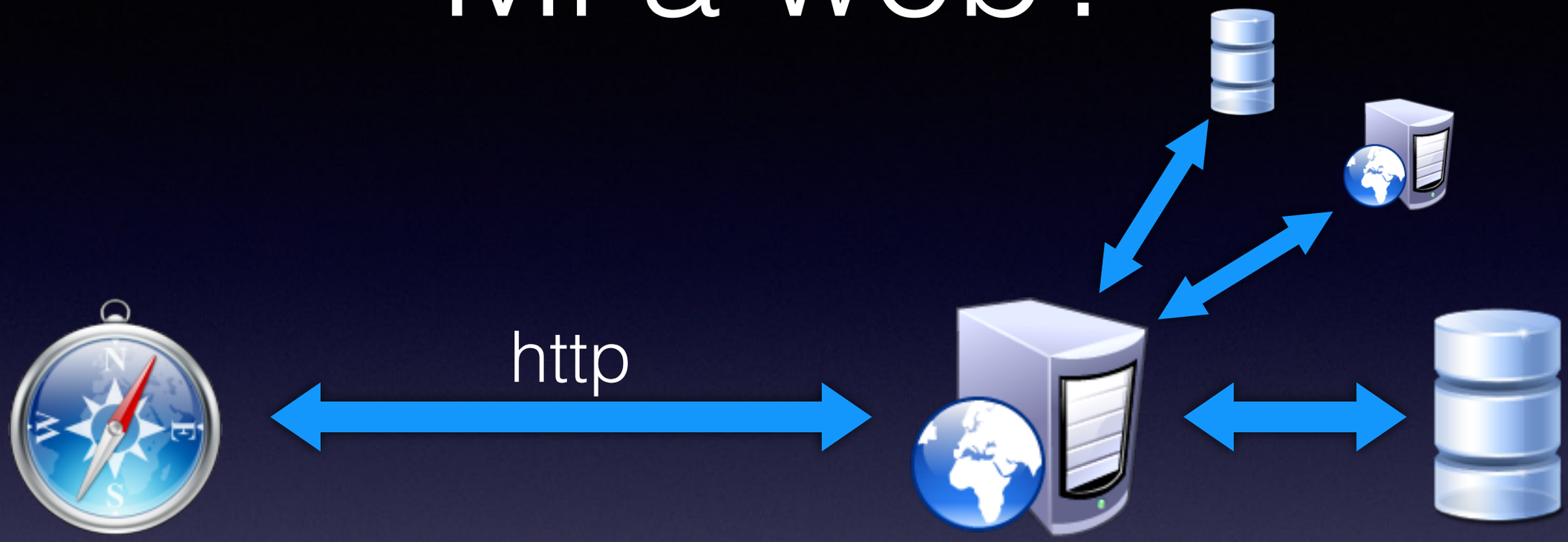
JavaScript

A kliens oldal programozása

Mi a web?



Browser

Webszerver

Apache

DB szerver

MySql

- html + css
- javascript
- flash, silverlight

- Vhost config
- php
- nodejs, ruby, python, java, .net, stb…

- Tárolt eljárások

Mi a JavaScript?

Nagyon Nem Java!

Nincs lefordított állomány, maga a forráskód kerül futtatáskor értelmezésre, script nyelv.

Miért **Java**Script?

- 1995 május - Brendan Eich: **Mocha**
- 1995 szeptember - **LiveScript**

- 1995 december - **JavaScript**
- 1996 március - Netscape Navigator 2.0
- 1996 augusztus - Internet Explorer 3.0 **JScript**
- 1997 június - ECMAScript…

ECMAScript?

- 1997 június - ECMA-262 script szabvány
  (European Computer Manufacturers Association)

- JavaScript
- ActionScript 3 - Adobe Flash, Air
- JScript .Net

A JavaScript jövője

- jQuery
- Node.js
- Coffee script
- Google Dart

Szóval JavaScript

- C/Java szerű szintaxis
- Objektum orientált, de nincsenek osztályok
- Objektumok vannak és prototípusok
- Nem típusos nyelv, de természetesen vannak típusok (number, string, object, function, stb…)
- Nincs explicit belépési pont

JavaScriptet hova?

```
<!DOCTYPE html>
<html>
  <head>
    <title>website</title>
    <style>…</style>
    <script>…</script>
  </head>
  <body>
    <script src="…"></script>
  </body>
</html>
```

Hello világ!!!

```
function kiabal(mit){
  var kialtas = mit + '!!!';
  alert(kialtas);
  console.log('Kiabáltam: ' + mit);
}

var i;
for ( i=0; i<10; i++ ){
  kiabal('Hello világ');
}
```

Mire jó a JavaScript?

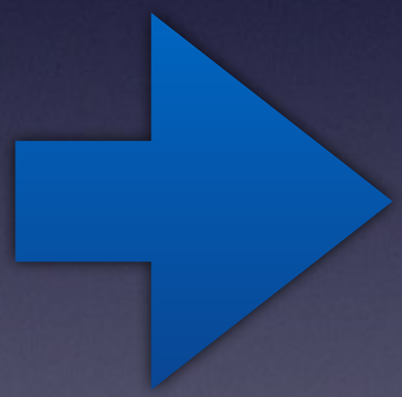
Arra használjuk, amit amúgy egy web oldal nem tud megtenni magától.

1. Módosít a HTML-en (DOM-on)
2. Kommunikál a szerverrel
   - XMLHTTPRequest
   - AJAX

Mi az a DOM?

Document Object Model

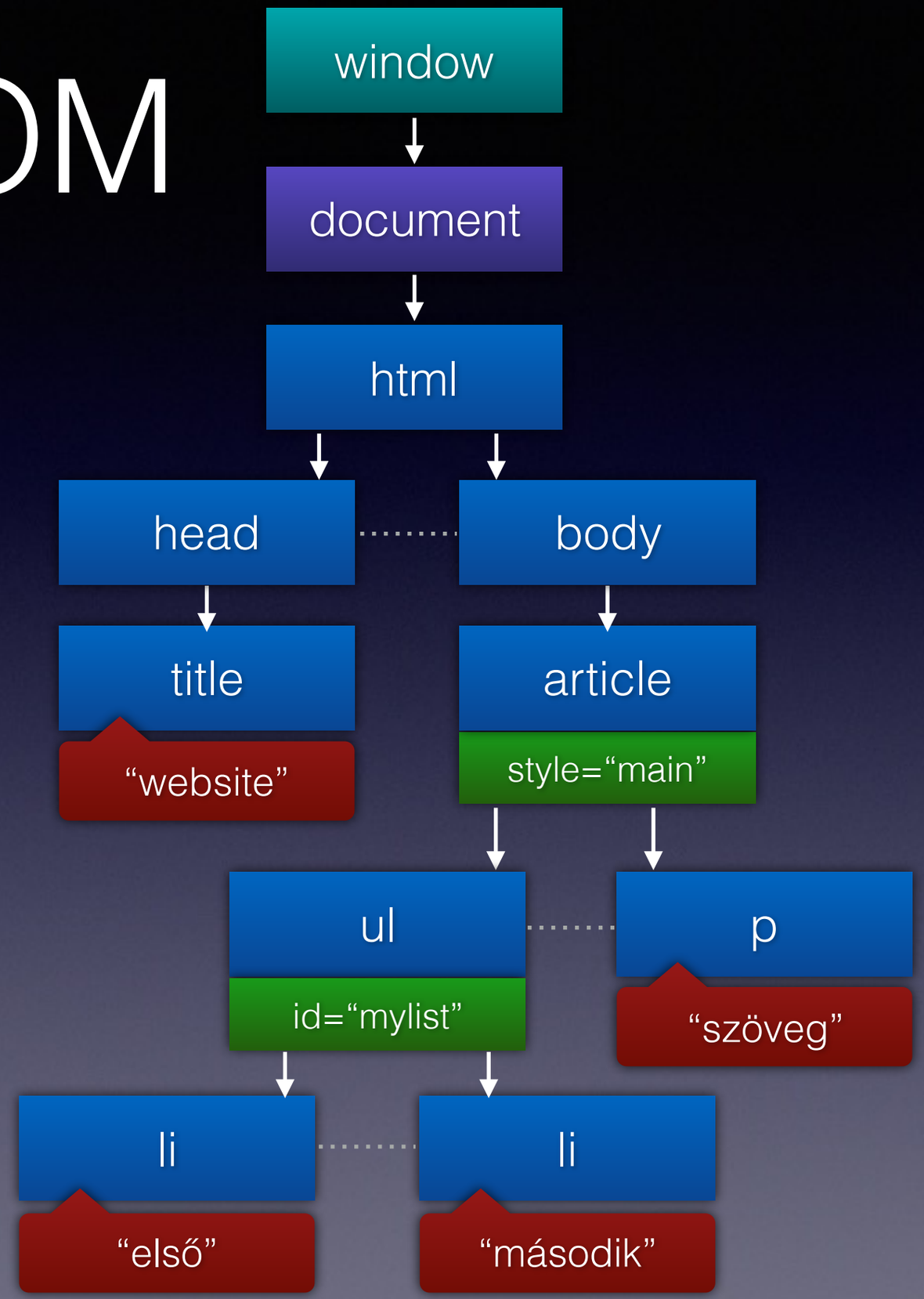
html
szöveg



DOM
hierarchikus objektum struktúra

DOM

```
<!DOCTYPE html>
<html>
    <head>
        <title>website</title>
    </head>
    <body>
        <article style="main">
            <ul id="mylist">
                <li>első</li>
                <li>második</li>
            </ul>
            <p>szöveg</p>
        </article>
    </body>
</html>
```



Keresés a DOM-ban

document.

- getElementById(id)
- getElementsByTagName(tagname)
- getElementsByName(name)
- getElementsByClassName(classname)

Keresés a DOM-ban

element.

- getElementsByTagName(tagname)
- getElementsByClassName(classname)
- childNodes
- parentNode
- firstChild, lastChild
- nextSibling, previousSibling

Új elem(ent) létrehozása

- document.createElement(tagname)
- element.appendChild(newElement)
- element.insertBefore(newElement, element)

```
var article, body, firstP;

article = document.createElement(‘article’);
body = document.getElementsByTagName(‘body’)[0];
body.appendChild(article);

firstP = document.createElement(‘p’);
body.insertBefore(p, body.fistChild);
```

Elem törlése

- parentElement.removeChild(elementToDelete)

```
…<body>
   <p id=“elem”>…</p>
</body>…
```

```
var elemToDelete;

elemToDelete = document.getElementById(‘elem’);
elemToDelete.parent.removeChild(elemToDelete);
```

Elem cseréje

- parentElement.replaceChild(elementToDelete)

```
…<body>
   <p id=“first”>…</p>
   <p id=“second”>…</p>
</body>…
```

```
var newP = document.createElement(‘p’);
var first = document.getElementById(‘first’);
first.parent.replaceChild(newP, first);
newP.id = ‘newFirst’;
newP.className = ‘fontos’;
```

window.onload

```
var myApp = {
   init: function(){
      console.log('inicializálom az applikációt');
   }
}

window.onload = function(){
   myApp.init();
}
```


PLEASE HELP
WILL CODE HTML FOR FOOD
GOD BLESS YOU