

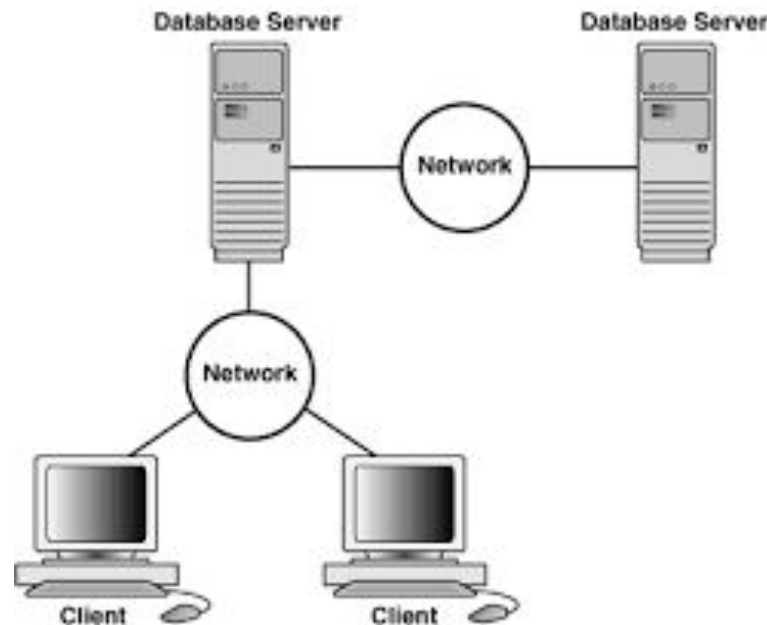
# Web-es kliens-szerver rendszerek

Kovács László, ME

# Kliens-szerver architektúra

Külön számítógépeken fut az alkalmazás háttér oldala és megjelenítési oldala

- a felhasználó a kliens előtt ül
- a kliens gép lehet vékony (csak megjelenítés) vagy vastag
- a kliens rész kéréseket küld a szerver felé
- a szerver több kliens kérését is kiszolgálja
- hálózati réteg, szoftver réteg



# Kliens-szerver architektúra

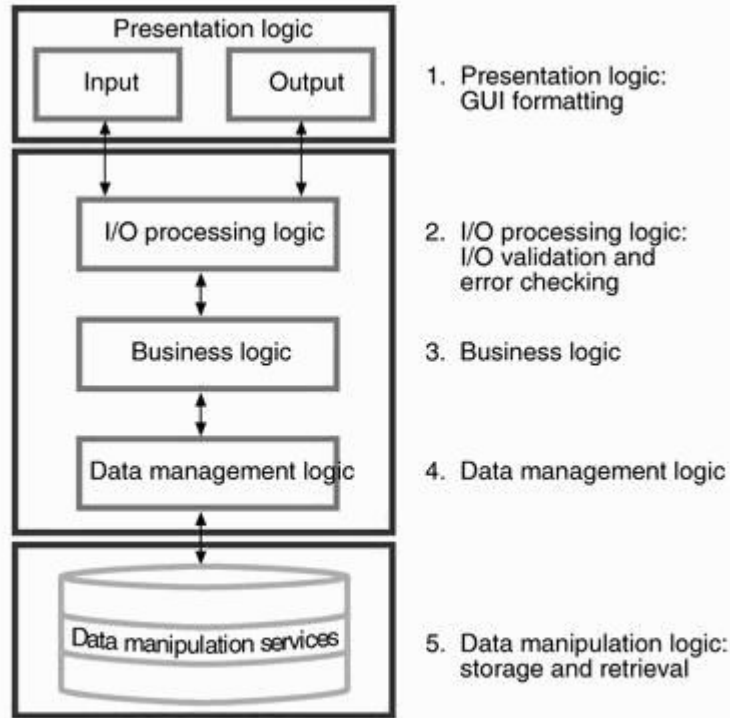
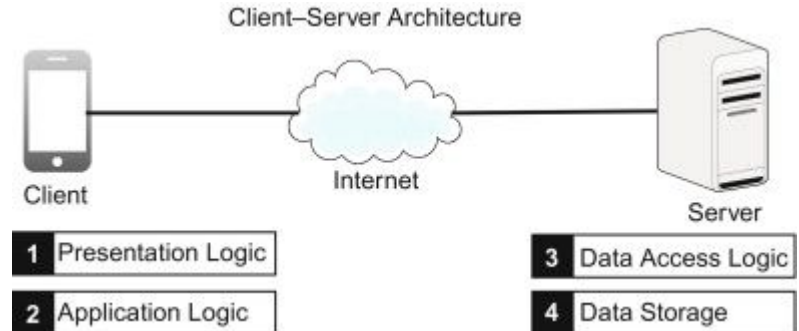
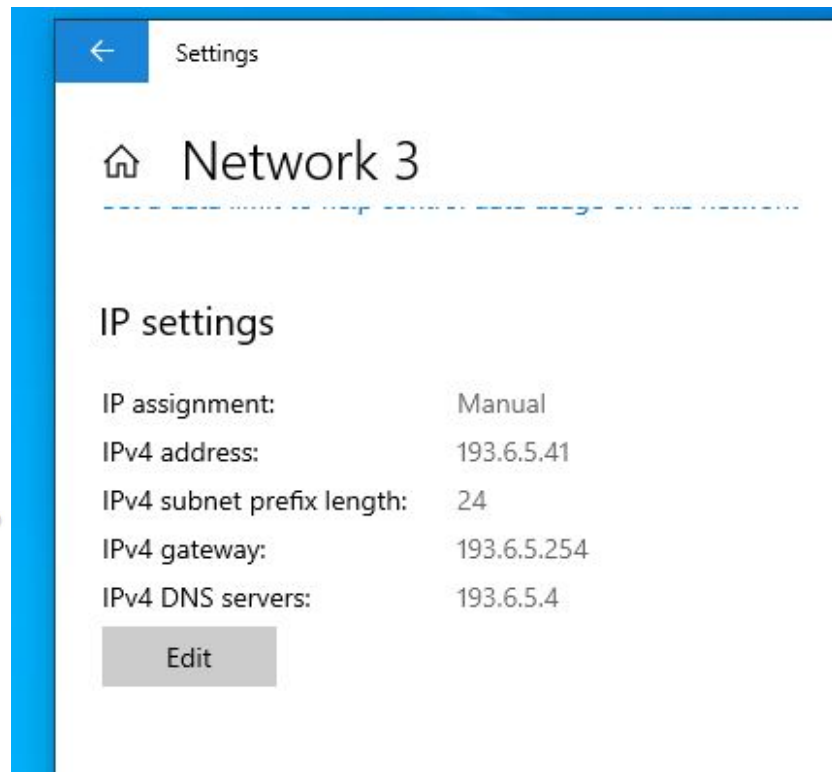
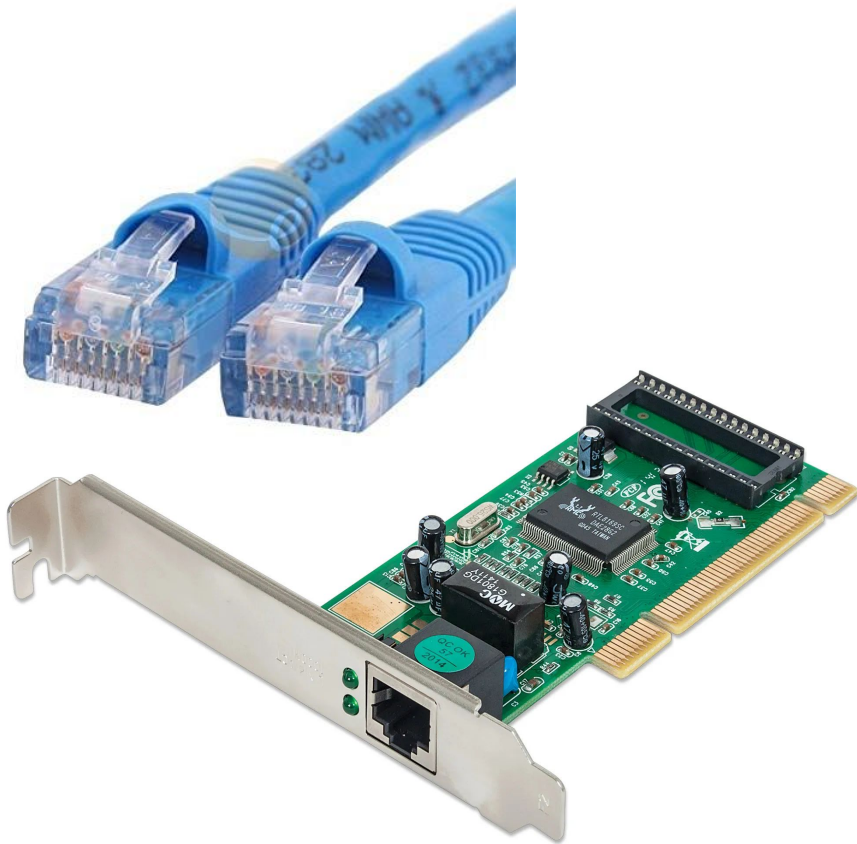


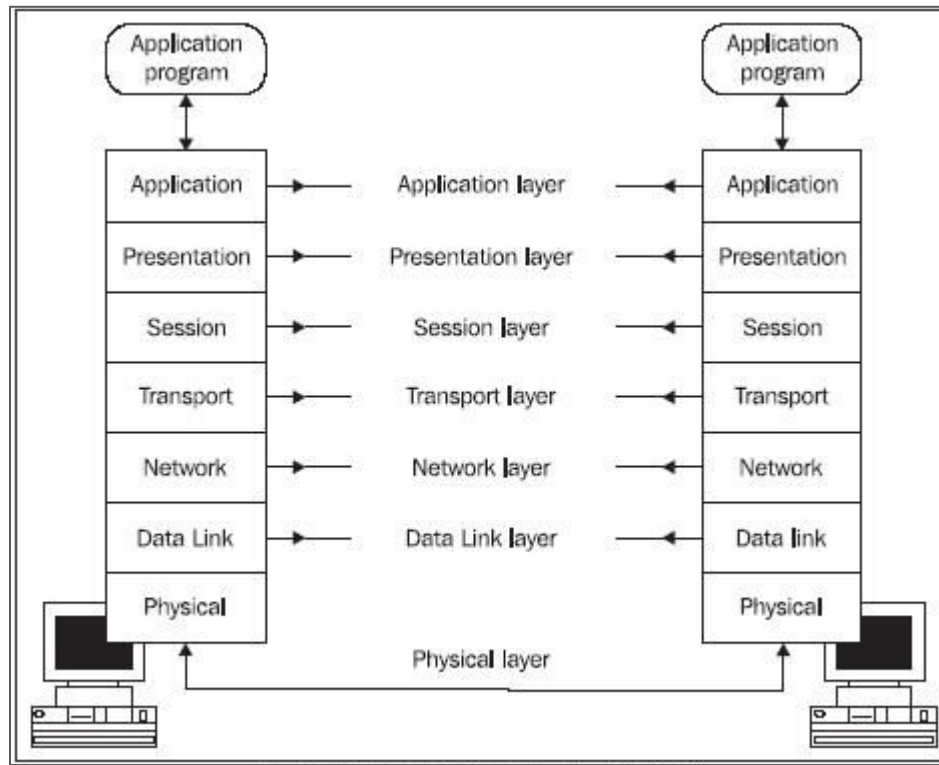
FIGURE 12.15 APPLICATION FUNCTIONAL LOGIC COMPONENTS



# Hálózati réteg



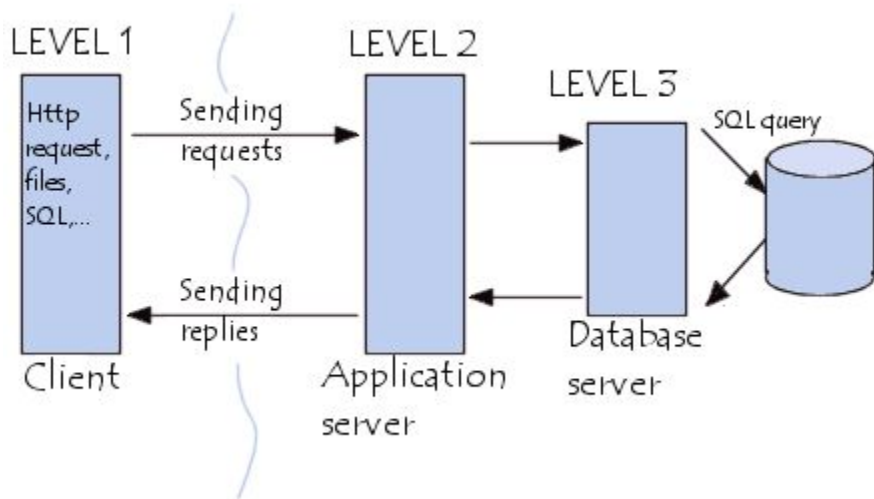
# OSI ISO hálózati modell



# Többrétegű architektúra

Több szervergép együtt működése

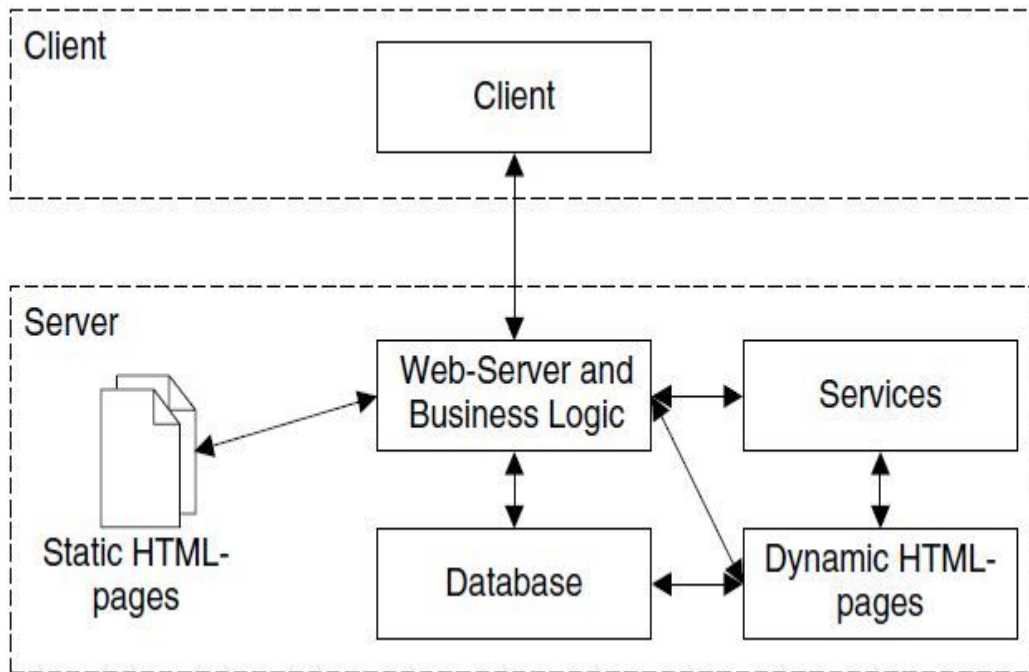
- alkalmazás szerver
- adatbázis szerver
- transactions szerver
- egy funkció több gépen is futhat
- teljesítmény kiegyenlítés



# Web-szerver architektúra

A kliens és szerver HTTP protokollban kommunikálnak

- a szerver dokumentumokat küld vissza
- statikus vagy dinamikus dokumentumok
- rendszerint HTML
- URL azonosítás
- hyperlink lehetőség



# URL cím

1 2 3 4 5 6 7 8  
`https://www.example.com:3000/path/resource?id=123#section-id`

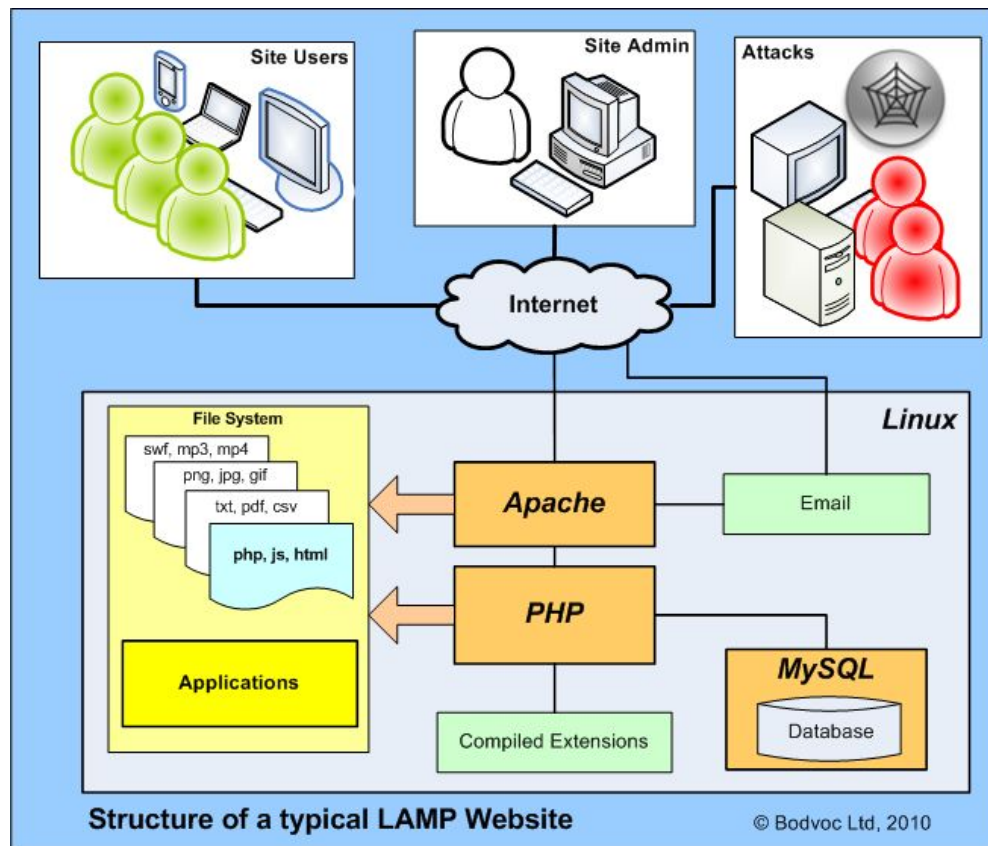
## Key

- 1 Scheme - defines how the resource will be obtained.
- 2 Subdomain - www is most common but not required.
- 3 Domain - unique value within its top-level domain.
- 4 Top-level Domain - hundreds of options now exist.
- 5 Port - if omitted HTTP will connect on port 80, HTTPS on 443.
- 6 Path - specify and perhaps find requested resource.
- 7 Query String - data passed to server-side software, if present.
- 8 Fragment Identifier - a specific place within an HTML document.



# Web-szerver modul

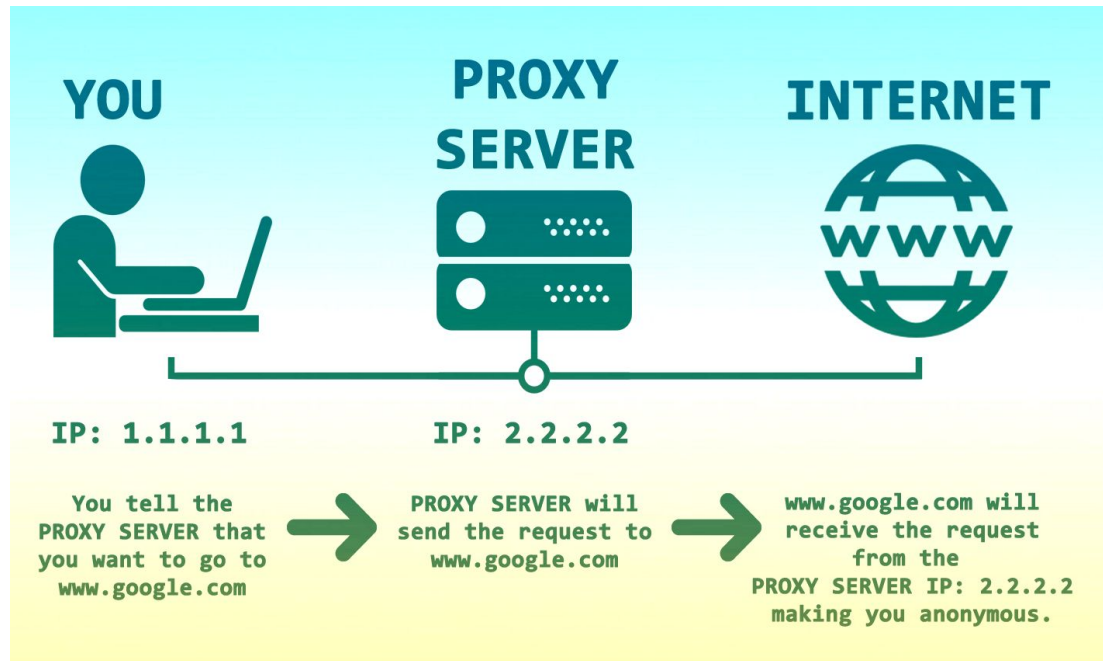
- URL kérések kiszolgálása
- logikai - fizikai cím
- statikus lap elérése
- dinamikus lap előállítása
- vezérlése
- hozzáférés védelem
- naplózás
- cache (gyorsítótár)



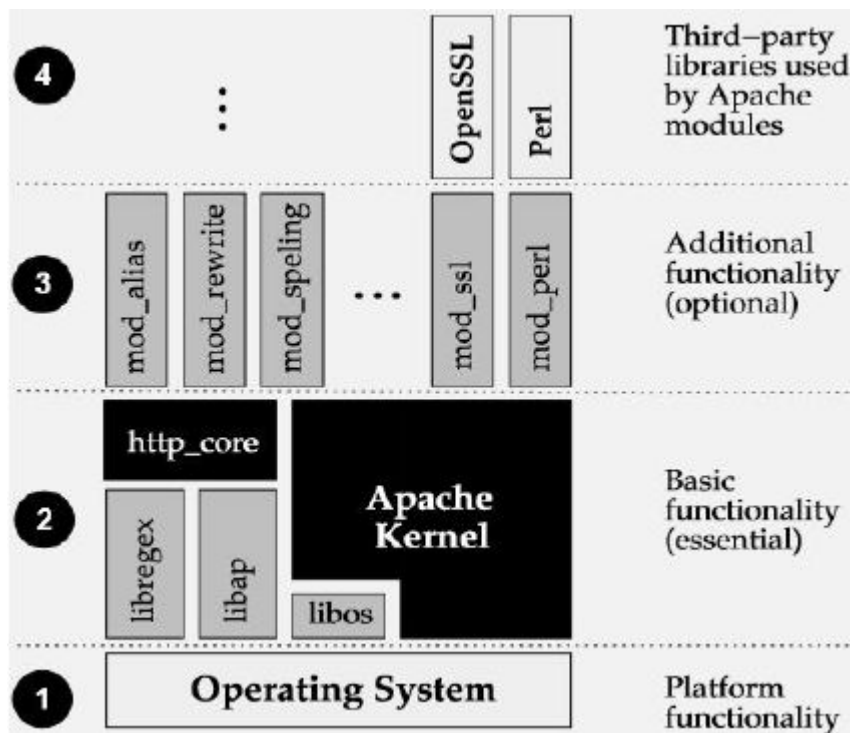
# Proxy-szerver modul

## Köztes szerver

- kérés átirányítása
- tűzfal, védelem
- tartalomszűrés
- gyorsítótár



# Apache web server modul



# HTML leíró nyelv

## Jelölő nyelv

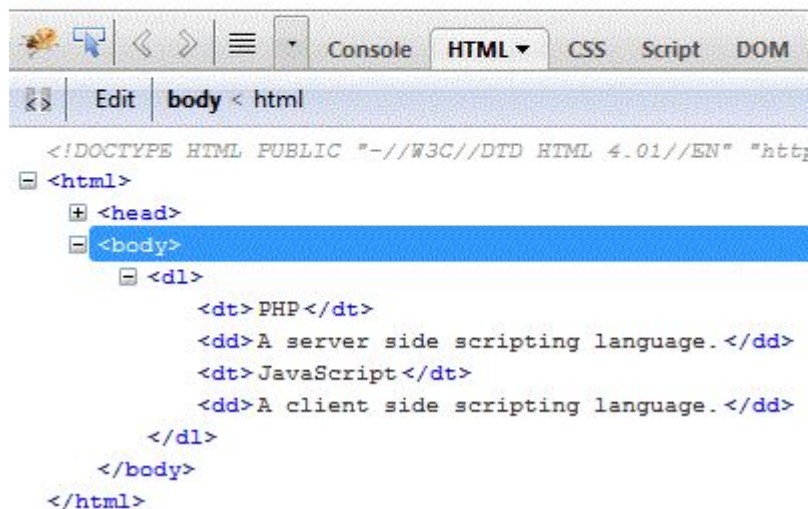
- <nyitótag>tartalom</zárótag>
- <üres\_elem/>
- <nyitótag attr=érték >
- megjelenítés orientált
- rugalmas szerkezet
- web böngésző megjeleníti

### PHP

A server side scripting language.

### JavaScript

A client side scripting language.



# HTML nyelv alapelemei

gyakorlási oldal: <https://jsfiddle.net/>

tanuló anyag: <http://www.w3schools.com/html/>

<html> : dokumentum gyökér eleme  
<head> : a böngésző fejlécének tartalma  
<body> : a böngésző törzs tartalma  
<p> : bekezdés  
<h1> : címsor  
<a href=""> : link  
<image src=""> : kép elem  
<br/> : soremelés  
<input> : beviteli elem

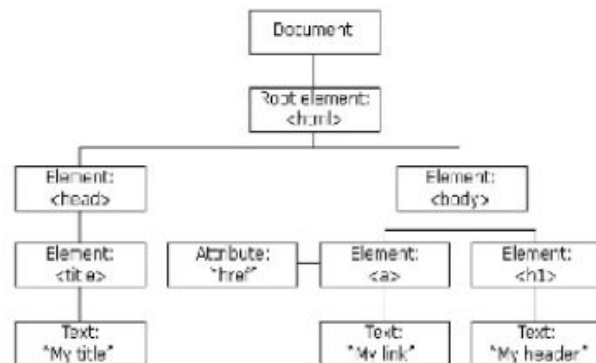
Az elemekhez társíthatóak attributumok

<pre>1 &lt;head&gt; 2 &lt;title&gt;Page Title&lt;/title&gt; 3 &lt;/head&gt; 4 &lt;body&gt; 5 6 &lt;h1&gt;Főcím&lt;/h1&gt; 7 &lt;p&gt;Helló világ&lt;/p&gt; 8 9 &lt;/body&gt; 10</pre>	<div>HTML ⚙️</div> <div>1</div> <div><h1>Főcím</h1><p>Helló világ</p></div>
---	---

# HTML nyelv alapelemei

Ügyeljünk az elemek egymásba ágyazására

`<mark>` : kiemelés  
`<i>` : dőlt betű  
`<b>` : vastagított



```
<h1>Minta szöveg</h1>

<p>
  <mark>Figyelem!</mark> <br/>
  <b>Ez egy fontos üzenet</b>
  <br/> <i>A belépés előtt
  <b> kopogjon</b> !</i>
</p>
<p>
  Hiba esetén : <a href="www.google.com"> Tanácsadás</a>
</p>
```

## Minta szöveg

**Figyelem!**

**Ez egy fontos üzenet**

*A belépés előtt kopogjon !*

Hiba esetén : [Tanácsadás](#)

# HTML nyelv alapelemei

Lista:

`<ul>` : nem számozott lista  
`<ol>` : számozott lista  
`<li>` : lista elem

Táblázat:

`<table>` : táblázat elem  
`<table border=>` : keret mérete  
`<tr>` : egy sor megadása  
`<td>` : egy mező megadása  
  
`<div>` : tartalmi egység a lapon

```
1 <body>
2
3 <H2>
4 Táblázat
5 </H2><br/><br/>
6 <table border="2">
7 <tr>
8   <td>11</td><td>12</td><td>13</td>
9 </tr>
10 <tr>
11   <td>21</td><td>22</td><td>23</td>
12 </tr>
13 <tr>
14   <td>31</td><td>32</td><td>33</td>
15 </tr>
16 </table>
17
18 </body>
19
```

HTML ⚙️

1

## Táblázat

11	12	13
21	22	23
31	32	33



# HTML nyelv alapelemei

Stílusok megadása

`<... style="..." >`

szín kijelölése

`color:blue;`

`color:rgb(c1,c2,c3);`

`background-color:`

karakter-készlet:

`font-family:arial`

`font-size:20px`

szöveg-igazítás:

`text-align: center;`

keret:

`border:2px solid Tomato;`

```
<h1>Minta szöveg</h1>
<mark style="font-size:26px">Figyelem!</mark> <br/>
<p style="color:white; background-color:gray" >
  <b>Ez egy fontos üzenet</b>
</p>
<p style="font-size:21px">
  <br/> <i>A belépés előtt <b>kopogjon</b> !</i>
</p>
<p>
  Hiba esetén : <a href="www.google.com"> Tanácsadás</a>
</p>
```

## Minta szöveg

**Figyelem!**

Ez egy fontos üzenet

*A belépés előtt kopogjon !*

Hiba esetén : [Tanácsadás](#)



# JavaScript nyelv alapelemei

A HTML tartalom dinamikus, kliens oldali kezelése

elemei: hivatkozás a HTML elemekre; eseményekhez kötés; műveletek és számítások

```
1 <body>
2 <h1>Próba Javascript</h1>
3
4 <button type="button"
5 onclick="document.getElementById('demo').innerHTML = Date()">
6 Az idő megjelenítése</button>
7 <p id="demo"></p>
8 </body>
9 |
```

HTML ⚙️

1

## Próba Javascript

Az idő megjelenítése

# JavaScript nyelv alapelemei

A JS kód a kliens oldalt fut le.

Módosíthatjuk, vezérelhetjük a megjelenítést, viselkedést

Programnyelv: adatok (változók) + vezérlési szerkezet

Változó

var vn = érték;

```
var x = 3;  
var y = 5;  
var z = x + y;  
document.write("<html><h1 style='color:white'>" + z + "</h1></html>")
```

A dokumentum tartalmának átírása

```
document.write("kód")  
document.write(vn)  
document.write("szöveg" + vn + "szöveg")
```

8

# JavaScript nyelv alapelemei

A JS kód megadásának módjai a HTML lapon

```
<script>  
    JS-kód  
</script>
```

A JS-kód eseményre fut le.

```
.onclick  
.onchange  
.onload  
.onmouseover
```

beolvasó ablak: `var x = window.prompt("..")`

```
1 <body>  
2 <h1>Próba Javascript</h1>  
3  
4 <button type="button"  
5 onclick="document.getElementById('demo').innerHTML = Date()">  
6 Az idő megjelenítése</button>  
7 <p id="demo"></p>  
8 </body>  
9 |
```

# JavaScript nyelv alapelemei

Minden elem egy objektum  
(van saját tulajdonsága, metódusa)

document

.getElementById()

.write(s)

.createElement(név)

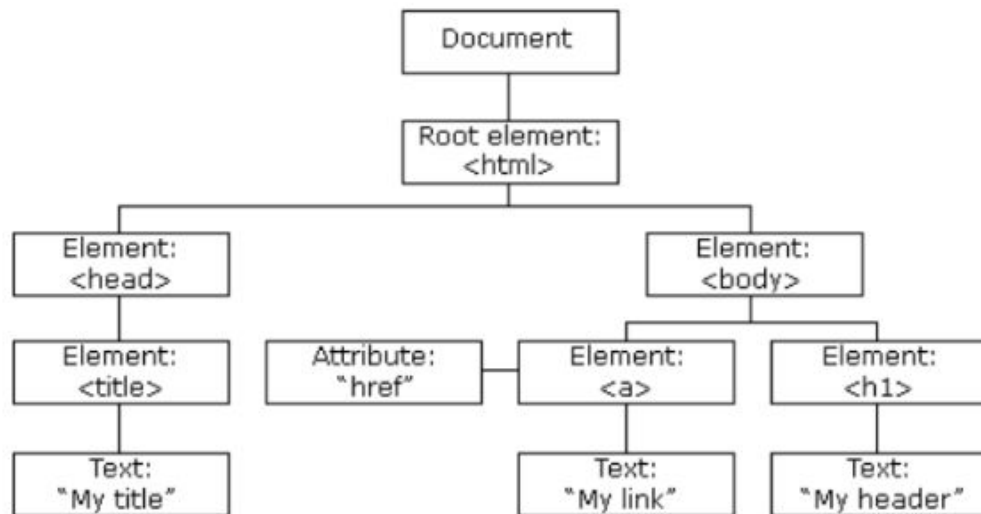
element:

.innerHTML

node:

.appendChild(node)

.removeChild(node)



# JavaScript nyelv alapelemei

Függvény: egy adott feladatot végrehajtó egység

jellemzői:

- név
- paraméterek
- kód

meghívása:

- név
- aktuális paraméterek

```
function szam_novel() {  
  var szam = document.getElementById("s1").innerHTML;  
  szam = Number(szam) + 1;  
  document.getElementById("s1").innerHTML = szam;  
}  
<button type="button" onclick="szam_novel()">Leptet</button>
```

```
<body>  
<script>  
function szam_novel() {  
  var szam = document.getElementById("s1").innerHTML;  
  szam = Number(szam) + 1;  
  document.getElementById("s1").innerHTML = szam;  
}  
  
</script>  
  
<H2>  
Számológép  
</H2><br /><br />  
<form>  
szam=<span id="s1">23</span> <br />  
<br />  
<button type="button" onclick="szam_novel()">Leptet</button>  
</form>  
</body>
```

## Számláló

szam=40

Leptet

# JavaScript nyelv alapelemei

Feltételes végrehajtás:

- egy igaz ág
- egy igaz és egy hamis ág
- több ág

```
if (felt) {  
}  
else {  
}
```

```
switch () {  
  case n:  
    break;  
  
  default:  
}
```

```
<body>  
<script>  
function szam_konvert() {  
  var szam = document.getElementById("s1").value;  
  var szov = " ";  
  szam = Number(szam) + 1;  
  if (szam > 100) {  
    szov = "sok";  
  } else {  
    szov = "kevés";  
  }  
  document.getElementById("pki").innerHTML = szov;  
  alert("Beállítva");  
}  
</script>  
  
<h2>  
Konverter</h2><br/><br/>  
<form>  
szam = <input type="text" id="s1" >  
<br/>  
<button type="button" onclick="szam_konvert()">Leptet</button>  
</form>  
<p id="pki"></p>  
</body>
```

## Konverter

szam = 123

Leptet

sok

# JavaScript nyelv alapelemei

Ciklusok:

- leszámlálós
- iteratív
- elől tesztelős
- hátul tesztelős

```
for (i=e; i<f; i++) {  
  }  
}
```

```
while (f) {  
  break;  
}
```

```
<body>  
<script>  
function rajzol() {  
  var szam = document.getElementById("s1").value;  
  var szoveg = "";  
  for (var i=0; i < szam; i++) {  
    szoveg = szoveg + "=".repeat(i) + "<br/>";  
  }  
  for (i=0; i<4; i++) {  
    szoveg = szoveg + "***" + "<br/>";  
  }  
  document.getElementById("pki").innerHTML = szoveg;  
}  
  
</script>  
  
<H2>  
Konverter</H2><br/><br/>  
<form>  
szint = <input type="text" id="s1" >  
<br/>  
<button type="button" onclick="rajzol()">Leptet</button>  
</form>  
<p id="pki" style="text-align:center"></p>  
</body>
```

## Konverter

szint =

```
  *  
 **  
 ***  
 ****  
 *****  
 ******  
 *******  
 *******  
 *******  
  **  
  **  
  **  
  **
```