

Web Programozás Előadás Házi Feladat Dokumentáció

Tóth Szabolcs Botond (TB8BGC)

Botka Gergő (BT80Y0)

Tartalom

- Bevezetés
- Fejlesztési környezet
- Weboldal felépítése
- Tartalom
- Táblázat – CRUD funkciók
- HTML5 API-k
- Chart.js
- AJAX-API
- OOJS
- React
- Részletek a programból
- Hozzáférés
- Csoportmunka

1. Bevezetés

Ez a dokumentáció a Web-programozás-1 házi feladat elkészítésének részleteit tartalmazza. A feladat célja egy többoldalas, modern webes technológiákat alkalmazó honlap elkészítése, különféle funkciókkal: CRUD, AJAX, ChartJS, React, OOJS stb.

2. Fejlesztési környezet

- **Szerkesztő:** Visual Studio Code
- **Böngésző:** Google Chrome
- **Frontend:** HTML5, CSS3, JavaScript, React
- **Verziókövetés:** Git + GitHub
- **Tárhely:** Nethely (<http://gamf.nhely.hu>)

3. Weboldal felépítése

Minden oldal közös elemei:

- **Fejléc (header):** címsor Web-programozás-1 Előadás Házi feladat
- **Menüsor (nav):** vízszintes, aktív menüpont kiemelve
- **Oldalsáv (aside):** oldalspecifikus információ
- **Tartalom (div.content):** aktuális oldal tartalma
- **Lábléc (footer):** készítők nevei és Neptun-kódjai

4. Tartalom

4.1. Kezdőlap – index.html

Statikus információ, egységes struktúrával. Aktív menüpontként megjelölve a "Kezdőlap".



5. Táblázat – CRUD funkciók (tabla.html, tabla.js)

- **Táblázat:** legalább 4x4-es
- **CRUD műveletek:** hozzáadás, szerkesztés, törlés
- **Validáció:** kötelező mezők, max hosszúság
- **Szűrés:** kereső mező alapján
- **Rendezés:** oszlopfejlécre kattintva

Web-programozás-1 Előadás Házi feladat

Kezdőlap Táblázat HTML5 ChartJS AJAX OOJS React

Táblázat oldalásháv

CRUD táblázat

Név	Kor	Város	Email	Hozzáadás
Keresés...				

Készítők: Botka Gergő (BT80Y0), Tóth Szabolcs Botond (TB8BGC)

6. HTML5 API-k (html5.html, html5.js)

- Web Storage
- Web Worker
- Geolocation API
- Drag and Drop
- Canvas és SVG

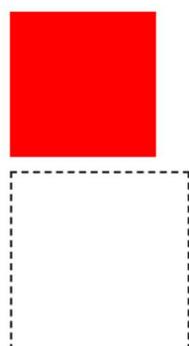
HTML5 API példák

Web Storage

Web Worker

Geolocation

Drag and Drop



Készítők: Botka Gergő (BT80Y0), Tóth Szabolcs Botond (TB8BGC)

7. Chart.js (chart.html, chart.js)

- 5x5-ös táblázat
- Bármelyik sorra kattintva vonaldiagramon jelenik meg az adat
- Chart.js használata

Web-programozás-1 Előadás Házi feladat

Kezdőlap Táblázat HTML5 ChartJS AJAX OOJS React

ChartJS oldalsáv

Adattábla és diagram

#	Érték 1	Érték 2	Érték 3	Érték 4	Érték 5
1	17	99	172	11	23
2	15	11	10	14	13
3	7	22	88	10	6
4	25	18	19	17	40
5	44	62	14	82	12

Készítők: Botka Gergő (BT80Y0), Tóth Szabolcs Botond (TB8BGC)

Adattábla és diagram

Érték 1 Érték 2 Érték 3 Érték 4 Érték 5

1	17	99	172	11	23
2	15	11	10	14	13
3	7	22	88	10	6
4	25	18	19	17	40
5	44	62	14	82	12



8. AJAX – API használat (ajax.html, ajax.js)

- API URL: <http://gamf.nhely.hu/ajax2/>
- **Műveletek:**
 - Read: rekordok lekérése és megjelenítése
 - Create: validált rekord felvitelle
 - Update: ID alapján adatok módosítása
 - Delete: rekord törlése
- **Validáció:** max 30 karakter, nem lehet üres

Web-programozás-1 Előadás Házi feladat

AJAX oldalsáv

Adatok API-val

Új adat

Név	Magasság	Súly	Hozzáadás
-----	----------	------	-----------

Adat módosítása

ID	Adatok lekérése		
Név	Magasság	Súly	Frissítés

Adat törlése

ID	Törlés
----	--------

Adatok listázása

Adatok lekérése

Készítők: Botka Gergő (BT80Y0), Tóth Szabolcs Botond (TB8BGC)

AJAX oldalsáv

Adatok API-val

Új adat

Kovács Aladár	192	105	Hozzáadás
---------------	-----	-----	-----------

Adat módosítása

1541	Adatok lekérése		
Kovács	Tamás	95	Frissítés

Adat törlése

ID	Törlés
----	--------

Adatok listázása

Adatok lekérése

ID: 1541, Név: Kovács, Magasság: Tamás, Súly: 95
ID: 1542, Név: Kovács Aladár, Magasság: 192, Súly: 105

9. OOJS – Objektumorientált példa (oojs.html, oojs.js)

- Person és Student osztály
- extends, super használat
- DOM manipuláció appendChild segítségével

The screenshot displays a web application interface. At the top, a green header bar contains the title "Web-programozás-1 Előadás Házi feladat". Below this is a dark navigation bar with tabs: "Kezdőlap", "Táblázat", "HTML5", "ChartJS", "AJAX", "OOJS" (which is highlighted in blue), and "React". The main content area has a teal sidebar on the left labeled "OOJS oldalsáv". The main content area displays a heading "OOJS példa: Személyek listája" and a button labeled "Új személy hozzáadása". Below the button, there are two entries: "Béla, 40 éves" and "Anna, 20 éves, iskola: GAMF". At the bottom of the page, a green footer bar contains the text "Készítők: Botka Gergő (BT80Y0), Tóth Szabolcs Botond (TB8BGC)".

10. React – Egyoldalas alkalmazások (react.html)

- **1. alkalmazás:** Számláló (useState)
- **2. alkalmazás:** Lista kezelő (hozzáadás, törlés, useState)
- **React CDNs használata**
- **Komponensek:** minden alkalmazás külön funkciókomponensként

Web-programozás-1 Előadás Házi feladat

Kezdőlap Táblázat HTML5 ChartJS AJAX OOJS React

React oldalsáv

React alkalmazások

Számláló

Aktuális érték: 0

Lista kezelő

Készítők: Botka Gergő (BT80Y0), Tóth Szabolcs Botond (TB8BGC)

Web-programozás-1 Előadás Házi feladat

Kezdőlap Táblázat HTML5 ChartJS AJAX OOJS React

React oldalsáv

React alkalmazások

Számláló

Aktuális érték: 3

Lista kezelő

- Valami

Készítők: Botka Gergő (BT80Y0), Tóth Szabolcs Botond (TB8BGC)

10. Részletek a programból

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists files in a tree structure under a 'WEB-PROG' folder. Files shown include ajax.html, ajax.js, chart.html, chart.js, html5.html, html5.js, index.html, oojs.html, oojs.js, react.html, style.css, tabla.html, and tabla.js. The 'index.html' file is currently selected and highlighted in blue. The main pane displays the HTML code for 'index.html'. The code includes a DOCTYPE declaration, a meta tag for UTF-8 charset, a title 'Web-programozás-1 Előadás Házi feladat', a link to 'style.css', and a navigation menu with links to various pages like 'index.html', 'tabla.html', 'html5.html', etc. It also contains a footer with author information.

```
<!DOCTYPE html>
<html lang="hu">
  <head>
    <meta charset="UTF-8">
    <title>Web-programozás-1 Előadás Házi feladat</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <header>
      <h1>Web-programozás-1 Előadás Házi feladat</h1>
    </header>
    <nav>
      <ul>
        <li><a href="index.html" class="active">Kezdőlap</a></li>
        <li><a href="tabla.html">Táblázat</a></li>
        <li><a href="html5.html">HTML5</a></li>
        <li><a href="chart.html">ChartJS</a></li>
        <li><a href="ajax.html">AJAX</a></li>
        <li><a href="oojs.html">OOJS</a></li>
        <li><a href="react.html">React</a></li>
      </ul>
    </nav>
    <aside>
      <p>Előadás beadandó</p>
    </aside>
    <div class="content">
      <h1> Főoldal </h1>
    </div>
    <footer>
      <p>Készítők: Botka Gergő (BT80Y0), Tóth Szabolcs Botond (TB8BGC)</p>
    </footer>
  </body>
</html>
```

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists files in a tree structure under a 'oojs.html' folder. Files shown include oojjs.js, react.html, style.css, tabla.html, and tabla.js. The 'style.css' file is currently selected and highlighted in blue. The main pane displays the CSS code for 'style.css'. The code defines styles for the 'aside' element (font-style: italic, background-color: #22a2b3, padding: 50px, height: 500px, width: 180px, float: left, box-sizing: border-box), the '.content' element (margin-left: 20%, padding: 20px, box-sizing: border-box), and the 'footer' element (clear: both, position: fixed, left: 0, bottom: 0, width: 100%, text-align: center).

```
aside {
  font-style: italic;
  background-color: #22a2b3;
  padding: 50px;
  height: 500px;
  width: 180px;
  float: left;
  box-sizing: border-box;
}

.content {
  margin-left: 20%;
  padding: 20px;
  box-sizing: border-box;
}

footer {
  clear: both;
  position: fixed;
  left: 0;
  bottom: 0;
  width: 100%;
  text-align: center;
}
```

```
# style.css      ◊ tabla.html      JS tabla.js      ◊ html5.html      JS html5.js      ◊ chart.html      JS chart.js      X
JS chart.js > ...
1  let chart;
2
3  function showChart(row) {
4      const cells = row.querySelectorAll("td");
5      const label = `Sor ${cells[0].innerText}`;
6      const values = Array.from(cells).slice(1).map(td => Number(td.innerText));
7
8      const ctx = document.getElementById("lineChart").getContext("2d");
9
10     if (chart) chart.destroy();
11
12     chart = new Chart(ctx, {
13         type: 'line',
14         data: {
15             labels: ["Érték 1", "Érték 2", "Érték 3", "Érték 4", "Érték 5"],
16             datasets: [
17                 {
18                     label: label,
19                     data: values,
20                     borderColor: 'blue',
21                     borderWidth: 2,
22                     fill: false
23                 }
24             ],
25             options: {
26                 responsive: true,
27                 scales: {
28                     y: { beginAtZero: true }
29                 }
30             }
31         });
32     }

```

```
◊ ajax.html > Ⓛ html > Ⓛ body > Ⓛ div.content > Ⓛ p#feedback
1  <!DOCTYPE html>
2  <html lang="hu">
3      <head>
4          <meta charset="UTF-8">
5          <title>AJAX API példa</title>
6          <link rel="stylesheet" href="style.css">
7          <script defer src="ajax.js"></script>
8      </head>
9      <body>
10         <header>
11             <h1>Web-programozás-1 Előadás Házi feladat</h1>
12         </header>
13         <nav>
14             <ul>
15                 <li><a href="index.html">Kezdőlap</a></li>
16                 <li><a href="tabla.html">Táblázat</a></li>
17                 <li><a href="html5.html">HTML5</a></li>
18                 <li><a href="chart.html">ChartJS</a></li>
19                 <li><a href="ajax.html" class="active">AJAX</a></li>
20                 <li><a href="oojs.html">OOJS</a></li>
21                 <li><a href="react.html">React</a></li>
22             </ul>
23         </nav>
```

```

let tableData = [];

document.getElementById("dataForm").addEventListener("submit", function (e) {
  e.preventDefault();
  const name = document.getElementById("name").value.trim();
  const kor = document.getElementById("kor").value.trim();
  const varos = document.getElementById("varos").value.trim();
  const email = document.getElementById("email").value.trim();

  if (!name || !kor || !varos || !email) return alert("Minden mező kötelező!");

  tableData.push({ name, kor, varos, email });
  updateTable();
  this.reset();
});

function updateTable() {
  const tbody = document.getElementById("dataTable").querySelector("tbody");
  tbody.innerHTML = "";
  tableData.forEach((row, index) => {
    const tr = document.createElement("tr");
    tr.innerHTML =
      ` ${row.name}</td>        <td>${row.kor}</td>        <td>${row.varos}</td>        <td>${row.email}</td>        <td>           <button onclick="editRow(${index})">Szerkeszt</button>           <button onclick="deleteRow(${index})">Törölés</button>         </td>`;     tbody.appendChild(tr);   }); } |
```

```

1  class Person {
2
3
4
5
6
7
8
9
10
11  render() {
12    const p = document.createElement("p");
13    p.innerText = this.describe();
14    document.getElementById("personContainer").appendChild(p);
15  }
16
17
18 class Student extends Person {
19   constructor(name, age, school) {
20     super(name, age);
21     this.school = school;
22   }
23
24   describe() {
25     return `${super.describe()}, iskola: ${this.school}`;
26   }
27 }
28
29 function addPerson() {
30   const p1 = new Person("Béla", 40);
31   const s1 = new Student("Anna", 20, "GAMF");
32   p1.render();
33   s1.render();
34 }
35

```

```

65     function updateData() {
66         const id = document.getElementById("updateId").value.trim();
67         const name = document.getElementById("u_name").value.trim();
68         const height = document.getElementById("u_height").value.trim();
69         const weight = document.getElementById("u_weight").value.trim();
70
71         if (!name || !height || !weight || name.length > 30) {
72             return alert("Hibás adat!");
73         }
74
75         fetch("http://gamf.nhely.hu/ajax2/", {
76             method: "POST",
77             headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
78             body: `op=update&id=${id}&name=${name}&height=${height}&weight=${weight}&code=${code}`
79         }).then(res => res.text())
80         .then(data => {
81             document.getElementById("feedback").innerText = "Sikeres frissítés!";
82             readData();
83         });
84     });
85 }
86
87 function deleteData() {
88     const id = document.getElementById("deleteId").value.trim();
89     fetch("http://gamf.nhely.hu/ajax2/", {
90         method: "POST",
91         headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
92         body: `op=delete&id=${id}&code=${code}`
93     }).then(res => res.text())
94         .then(data => {
95             document.getElementById("feedback").innerText = "Sikeres törlés!";
96             readData();
97         });
98 }
99

```

html5.html > html > body > footer > p

```

2   <html lang="hu">
9    <body>
27      <div class="content">
38        <section>
41          <p id="workerOutput"></p>
42          <h3>The h3 element represents a section heading.</h3>
43        <div>MDN Reference</div>
44        <h3>Geolocation</h3>
45        <button onclick="getLocation()">Helymeghatározás</button>
46        <p id="geoOutput"></p>
47      </section>
48
49      <section>
50        <h3>Drag and Drop</h3>
51        <div id="drag1" draggable="true" ondragstart="drag(event)" style="width:100px; height:100px; background-color:red;"></div>
52        <div id="dropZone" ondrop="drop(event)" ondragover="allowDrop(event)" style="width:120px; height:120px; border:2px dashed black;"></div>
53      </section>
54
55      <section>
56        <h3>Canvas</h3>
57        <div>
58          <canvas id="myCanvas" width="200" height="100" style="border:1px solid black; margin-bottom:10px;"></canvas>
59        </div>
60
61        <div>
62          <h3>SVG</h3>
63          <div>
64            <svg width="100" height="100">
65              <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
66            </svg>
67          </div>
68        </div>
69      </div>
70      <div>
71        <p>Készítők: Botka Gergő (BT80Y0), Tóth Szabolcs Botond (TB8BGC)</p>
72      </div>
73    </body>
74  </html>

```

11. Hozzáférés

- GitHub: github.com/szaTB8BGC/hello-world
 - **URL:** www.gamf999.nhely.hu/
 - **FTP hozzáférés:**
 - FTP szerver: [ftp.gamf.nhely.hu](ftp://ftp.gamf.nhely.hu)
 - Felhasználónév: gergo3000
 - Jelszó: Gergo2004
-

12. Csoportmunka

- Botka Gergő – nethely, dokumentáció, Index, CSS design, CHART
 - Tóth Szabolcs Botond – AJAX, CRUD táblázat, HTML5, OOJS, REACT
-

