

# JEGYZŐKÖNYV

Web technológia alapjai

Node.js 101, azaz egy gyors weboldal  
készítése Express backendel

Készítette: **Kiss Konrád Soma**

Neptunkód: **CNY8MP**

Dátum: 2024. május 04.

**Miskolc. 2024.**

# Tartalomjegyzék

<b>Bevezetés</b>	<b>3</b>
<b>1. HTML</b>	<b>4</b>
1.1 Tartalom	4
1.2 Szerkezet megtervezése	4
1.3 Felhasznált elemek	4
1.4 Űrlap	5
1.5 Videó	5
<b>2. CSS</b>	<b>5</b>
2.1 – 2.3 Megfelelő formázások alkalmazása	5
2.4 Táblázat formázása	5
2.5 Menü kialakítása	5
2.6 – 2.8 Háttérszín, linkek, űrlap elemek és gombok formázása	5
2.9 Külső CSS fájlban, és beágyazva is legyen CSS alkalmazva	5
<b>3. JavaScript, jQuery, JSON és AJAX</b>	<b>6</b>
3.1 Form ellenőrzés	6
3.2 jQuery animáció	6
3.3 Elemek kiválasztása	6
3.4 – 3.6 Új html elem készítése, meglévő módosítása, JSON adattárolás, AJAX alkalmazásával a JSON fájl megjelenítése az oldalon	6
<b>4. Szorgalmi</b>	<b>6</b>

## Bevezetés

A feladatom egy HTML5 alapokon nyugvó weboldal tervezése és megvalósítása volt, amely egy választott témára épülve teljesíti a meghatározott követelményeket. Az oldalnak eleget kell tennie bizonyos kritériumoknak, amelyek magukban foglalják a használt technológiákat, az elrendezést és a szükséges tartalmi elemeket. A projekt témája a "Node.js 101" volt, ami azt jelenti, hogy egy rövidebb, oktatói jellegű oldalt kellett elkészítenem.

A kész projekt fájlrendszeri elrendezése az alábbi lett:

- A projekt gyökérkönyvtára: **WebTechCNY8MP**
  - A statikus fájlokat tároló mappa: **public**
    - Az interaktivitást biztosító fájlokat tároló mappa: **lib**
    - A **.html**, **.css** és **egyéb** tartalommal rendelkező fájl
  - A Node.js backend fő programfájlja: **index.js**
  - A felhasznált könyvtárakat beállító, és a projekt működését elősegítő **package.json** fájl

Minden más a mappában található fájl ideiglenes, vagy a működéshez elengedhető:

- **node\_modules** mappa: a könyvtárakat tartalmazó mappa, rendkívül nagy és sok fájlból áll, ezért nem szokás a projekttel együtt feltölteni, hanem a **package.json** -ban feljegyezzük hogy mely könyvtárakat használtuk, és az **npm install** paranccsal letöltjük őket.
- **.gitignore**: a projekt feltöltéséhez szükséges fájl, az ideiglenes/nem feltöltendő mappákat és fájlokat szűri ki.
- **.github** mappa: a projekt esetleges külső szerveren történő futtatásához szükséges folyamatok leírását tartalmazza, lokális futtatás esetén szükségtelen.

# 1. HTML

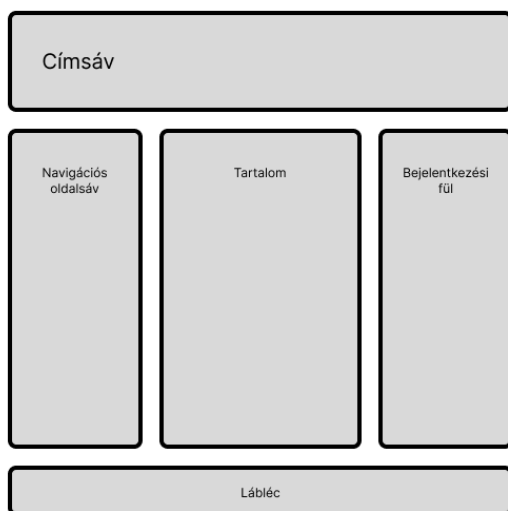
## 1.1 Tartalom

A feladat teljesítéséhez egy eléggé tisztes hosszúságú tartalmat készítettem el, amely leírja a legfontosabb tudnivalókat a HTML, CSS, JavaScript és Node.js elkezdéséhez.

- index.html – főoldal
- html.html – HTML leírás
- css.html – CSS leírás
- js.html – JavaScript és jQuery leírás
- nodejs.html – Node.js leírás
- express.html – Express könyvtár leírás
- contact.html – Kapcsolatlétesítéshez készült űrlap
- about.html – A készítő elérhetőségei

## 1.2 Szerkezet megtervezése

Az oldal szerkezetének a megtervezése során arra figyeltem, hogy az elrendezés egy tiszta, jól megszokott rendszerű legyen. Megterveztem egy egyszerű elképzelést, és gyorsan megvalósítottam a kódban CSS Grid-ek használatával.



Az elrendezést kódbéli megvalósítás részéről az alábbi módon kell elképzelni:

```
<body>
  <main>
    <header></header>
    <kk-sidebar></kk-sidebar>
    <section></section>
    <footer></footer>
  </main>
</body>
```

## 1.3 Felhasznált elemek

Az oldal az alábbi elemeket alkalmazza:

- **div** – rengeteg részen, ahol előre kész elem nincs a feladatra, megfelelő elválasztásra használtam. Példa: a css.html oldalon a margó példát ezzel készítettem el.
- **span, p, címsorok** – az oldalak tartalma ezekkel az elemekkel van elrendezve, kiemelve.
- **képek**: a html.html-en, illetve minden oldalon a címsoron alkalmazva van egy szép megjelenés eléréséhez.
- **táblázat**: az about.html-en az adataim ebben vannak elrendezve.
- **linkek**: a cikkek során többször is alkalmaztam őket további hasznos információk megjelölésére.
- ...

## 1.4 Űrlap

Minden gyakran használt elem alkalmazásra került az alábbi módon:

The form contains the following elements from top to bottom:

- A text input field labeled "Név".
- A text input field labeled "E-mail".
- A dropdown menu labeled "Kapcsolatfelvétel oka" with a downward arrow icon.
- A large text area labeled "Üzenet".
- A text input field labeled "Korcsoport".
- A checkbox labeled "Hírlevél".
- A radio button group labeled "Nem:" with options "Tartózkodom", "Férfi", and "Nő".
- A color picker labeled "Kedvenc szín:" with a red color swatch.
- A date input field labeled "Chapta, válaszd ki a mai dátumot:" with a placeholder "mm / dd / yyyy".
- A blue submit button labeled "Küldés".

## 1.5 Videó

Egy beágyazott videó található a főoldalon, amely a témával kapcsolatos és ellátam egyedi irányításhoz használt gombokkal.

## 2. CSS

### 2.1 – 2.3 Megfelelő formázások alkalmazása

Attól függően, hogy milyen módon volt a legegyszerűbb megoldani, úgy alkalmaztam a megfelelő formázási kijelöléseket. Példák: a kódblokkok style attribútum segítségével vannak színezve, a főcím és pár elrendezési elem azonosító alapján van kijelölve, és pár ismétlődő elem osztály alapján.

### 2.4 Táblázat formázása

Az about.html oldalon látható egy megformázott táblázat az adataimmal.

### 2.5 Menü kialakítása

A bal oldali oldalsávon látható a menü, benne az összes elérhető oldallal.

### 2.6 – 2.8 Háttérszín, linkek, űrlap elemek és gombok formázása

Mindezen formázások megtalálhatóak a **style.css** fájlban, és láthatóak az oldalon.

### 2.9 Külső CSS fájlban, és beágyazva is legyen CSS alkalmazva

Szinte a projekt minden oldalán található példa a beágyazott CSS alkalmazására, és egy közös **style.css** fájlal lett elkészítve úgy az oldal, hogy végig konzisztens kinézete legyen.

## 3. JavaScript, jQuery, JSON és AJAX

### 3.1 Form ellenőrzés

A **contact.html** oldalon csak akkor engedi az oldal, hogy elküldjünk egy üzenetet, amennyiben minden helyen megfelelő adat található. Ha valahol hibás a bemenet, letiltásra kerül a küldés gomb, és egy üzenettel, illetve piros kerettel kiemeli a hibát az oldal.

Bár a HTML5 ad pár alapvető eszközt ezeknek az adatoknak a leellenőrzésére, egy kicsit robusztusabb, teljesen JavaScript alapú hibaellenőrzést készítettem.

### 3.2 jQuery animáció

A jQuery segítségével három helyen készítettem apróbb animációkat. Az oldalsávok kinyílása, bezárása jQuery segítségével történik, illetve a **js.html** oldalon egy konkrét példát egy jobbra-balra animált elemmel mutatok az utasításokról.

### 3.3 Elemek kiválasztása

A feladat azt kérte, hogy a jQuery kijelölőket használjam HTML tag névvel, osztállyal és azonosítóval. Ezekre található példa a **kk.js** és **login.js** fájlban:

- HTML tag névvel kelülnek kijelölésre a **code** blokkok, ezzel kijelölve az összeset és alkalmazva rájuk a szintaxis színezést.
- Osztállyal és azonosítóval: a jobb oldali navigációs sáv gombját egy osztály és azonosító kombinációjával sikerül a legpontosabban kijelölni.

### 3.4 – 3.6 Új html elem készítése, meglévő módosítása, JSON adattárolás, AJAX alkalmazásával a JSON fájl megjelenítése az oldalon

Az összes oldalon megtalálható bal oldali oldalsáv mindhárom kritériumot kitölti az alábbi módon:

- Egy JSON fájlban található az összes hivatkozás, ezzel biztosítva, hogy megegyeznek az oldalakon keresztül.
- A meglévő, de még csak nyitó/csukó gombbal felszerelt oldalsávot **szerkesztem** az **újonnan létrehozott** navigációs gombok hozzáadásával.
- Mind ehhez a szükségét **content.json** fájlt a jQuery AJAX metódusaival töltöm be.

## 4. Szorgalmi

A szorgalmi feladat az volt, hogy az oldalt tegyük elérhetővé egy saját Node.js backend segítségével. Ezt az **index.js** fájlban belül értem el, amelyet a mappán belül lehet elindítani az **npm start** paranccsal.