

# Jegyzőkönyv

Webkönyvtárak

Webprogramozás

Algoritmus gyűjtő platform

Készítette: **Bárdos Triszten Krisztofer**

Neptunkód: **CUNPO1**

Dátum: 2024. május 12.

**Sárospatak, 2024.**

## Tartalomjegyzék

Bevezetés.....	4
1. Feladat - HTML.....	5
1.1 – Legalább 5 HTML fájl .....	5
1.2 – HTML5 szerkezet megtervezése.....	6
1.3 – HTML elemek (div, span, p, címsorok, képek, táblázat, linkek, html5 elemek) .....	9
Div .....	9
Span .....	9
Bekezdés (p) .....	9
Címsorok .....	9
Táblázat (table).....	10
Linkek.....	10
Számozott lista .....	10
1.4 – Űrlap elemek.....	11
Az űrlap felépítése.....	11
Az űrlaphoz tartozó teljes programkód: .....	12
1.5 - Video .....	13
A videóhoz tartozó kód: .....	14
2. Feladat – CSS .....	15
2.1 – style attribútumban formázás .....	16
2.2 – azonosító alapján formázás, táblázat formázása .....	17
2.3 – osztály alapján formázás .....	17
2.4 – menü kialakítása .....	18
Navigációs sáv:.....	18
A menü sáv:.....	18
2.5 – Linkek formázása .....	19
2.6 – Űrlap elemek formázása.....	19
3. Feladat - JavaScript, jQuery, JSON, AJAX .....	21
3.1 – form ellenőrzés .....	22
3.2 – jQuery animáció .....	23

3.2 – elemek kiválasztása html tag név alapján, osztály alapján, azonosító (id) alapján .....	23
3.3 – új elem készítése.....	24
3.3 – beágyazott JSON fájl elkészítése .....	24
3.4 – JSON fájl megjelenítése .....	24
4. Feladat – Server.....	25
Struktúra szerkezet .....	26

# Bevezetés

## A feladat leírása:

Egy internetes felület létrehozása, melyen különböző típusú és nehézségű algoritmusok vannak feltüntetve ezek nevével, kategóriájával, rövid leírásával, futási komplexitásával és a hozzájuk tartozó programok kódjával, ami a C++, illetve JavaScript nyelvekben valósult meg. A JavaScript programozási nyelvben megírt programkódok a P5.js JavaScript könyvtár elemeit és funkcióit tartalmazza.

Az algoritmusok listája rendkívül széleskörű, egészen pontosan 7 különböző kategóriába tartozó és ezen belül 12 algoritmus kerül bemutatásra, mindegyikhez egy-egy rövidebb multimédia hozzáadásával.

A fentebb említett algoritmusok elnevezésének listája:

1. Lineáris keresés
2. Bináris keresés
3. Buborék rendezés
4. Kiválasztásos rendezés
5. Kiválasztás
6. K-középérték
7. Bináris fabejárás
8. N-ágú fabejárás
9. Elárasztásos kitöltés
10. A\* (A star)
11. Futáshossz-kódolás
12. Euklideszi algoritmus

A választott fejlesztői környezet a projekt megvalósítására: Visual Studio Code

Az algoritmusokhoz tartozó C++ programok megvalósítására választott fejlesztői környezet: Code::Blocks 20.03

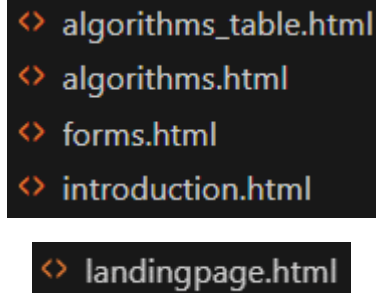
Az algoritmusokhoz tartozó JavaScript programok megvalósítására választott fejlesztői környezet: Visual Studio Code

# 1. Feladat - HTML

## 1.1 – Legalább 5 HTML fájl

A HTML fájlok:

- algorithms\_table.html
- algorithms.html
- forms.html
- introduction.html
- landingpage.html



```
<> algorithms_table.html
<> algorithms.html
<> forms.html
<> introduction.html
<> landingpage.html
```

A „*landingpage.html*” a „*public*” mappában helyezkedik el, a többi html fájl azon belül a „/html” mappában.

Az oldalakhoz tartozó URL címek:

- <http://localhost:3000/> : landingpage.html
- <http://localhost:3000/home> : landingpage.html
- <http://localhost:3000/introduction> : introduction.html
- <http://localhost:3000/algorithms> : algorithms.html
- <http://localhost:3000/forms> : forms.html
- [http://localhost:3000/algorithms\\_table](http://localhost:3000/algorithms_table) : algorithms\_table.html

## 1.2 – HTML5 szerkezet megtervezése

Fej rész: (<head>):

```
<!DOCTYPE html>
<html lang="en">
```

A <!DOCTYPE html> tag ismerteti a böngészővel a dokumentum típusát (*html*), a dokumentum nyelve angol (*en*).

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="/css/abstract.css">
  <link rel="stylesheet" href="/css/landingpage.css">
  <link rel="stylesheet" href="/css/jquery-ui.css">
  <script src="/jquery-3.7.1.js"></script>
  <script src="/jquery-ui.js"></script>
  <script src="/script/abstract.js"></script>
  <script src="/script/landingpage.js"></script>
  <title>Kezdőlap</title>
</head>
```

Az oldalak címe a böngészőben minden html fájlban megadott.

A HTML oldalak karakter kódolása, karakter készlete **UTF-8**.

Az oldalakhoz tartozó stílusok az „*abstract.css*”, az aktuális oldalhoz tartozó stílus fájl és a „*jquery-ui.css*” fájlok, a script fájlok a „*jquery-3.7.1.js*”, a „*jquery-ui.js*”, az „*abstract.js*” és az aktuális oldalhoz tartozó script fájl.

Header struktúra az oldalak törzsében:

```
<header>
  <div id="img_container">
    <a href="/landingpage.html"></a>
  </div>
  <div class="header_container" id="landingpage_header_container"><h1 id="landingpage_h1"
hidden>Üdvözlöm</h1></div>
```

```
<div class="hidden_object"></div>
</header>
```

A fejléc tartalmazza a „*landingpage.html*” hivatkozást, melyet a „*logo\_landingpage.png*”- képre kattintva érhetünk el.

A fejlécbe szerepel egy címsor is (<h1>). A fejléc és egyben az oldal címe minden oldalon adott.

A weboldalak tartalma a „*main\_content*” osztály attribútummal ellátott <div> elemekben található, melyek azonosítója (id attribútum) a „*main\_content*” szöveg és az aktuális html fájl nevének összefűzése egy ’\_’ karakterrel.

```
<div class="main_content" id="main_content_landingpage">
.
.
.
</div>
```

A tartalom szeparálására egyéni <hr> (horizontal rule) tageket használtam.

Az osztály attribútumok: „*navhr*”, „*fancy\_hr*”, „*footer\_horizontal\_rule*”.

Minden oldalon szerepel menu elem, illetve a hozzá tartozó gomb, mely megnyomásával nyitható / csukható a balra igazított menu lista.

```
<div id="toggle_nav_btn_container">
  <button class="toggle_nav_btn" id="toggle_nav_btn_landingpage">
    <div class="toggle_nav_text" id="toggle_nav_text_landingpage">>></div>
  </button>
</div>
```

A „*toggle\_nav\_btn*” osztály attribútummal rendelkező gombhoz tartoznak JavaScript és jQuery funkciók, illetve stílusok.

A gomb szövege: „>”, illetve „<”.

Az oldalakhoz tartozó menu:

```
<nav class="menubar" id="menubar_landingpage">
  <ul>
    <li class="submenu">
```

```

        <div class="complex_list_element"><div class="complex_list_element_reference"><a
href="/html/algorithms.html">Algoritmusok</a></div><div
class="subelement_arrow_container">&#x25B6;</div></div>
        <ul class="submenu_items"></ul>
        </li><hr class="navhr">
        <li><a href="/html/introduction.html">Bemutakozás</a></li><hr class="navhr">
        <li><a href="/html/forms.html">Űrlap kitöltése</a></li><hr class="navhr">
        <li><a href="/html/algorithms_table.html">Algoritmus táblázat</a></li><hr class="navhr">
    </ul>
</nav>

```

A menu egy „menubar” osztály attribútummal ellátott `<nav>` elem.

Az almenu egy rendezetlen (unordered) lista, melynek lista elemei a különböző algoritmusok felsorolása. Az itt megnyitott algoritmus és tulajdonságai az oldal újra töltődése nélkül kerül bemutatásra. Ennek szülő lista eleme egy komplex lista elem típus, melynek osztály attribútuma „complex\_list\_element”.

A lista elemek hivatkozást tartalmaznak, a menu fő elemei rendre az Algoritmusok, Bemutakozás, Űrlap kitöltése és az Algoritmus táblázat menupontokra hivatkoznak.

A fő lista elemeket az egyénileg létrehozott „navhr” osztály attribútummal rendelkező `<hr>` tagek szeparálják el egymástól.

Lábléc az oldalakon:

```

<footer>
    <div class="link_container">
        <a href="https://unithe.hu/" target="_blank">Az egyetem honlapja</a>
        <a href="/html/forms.html" target="_blank">Űrlap kitöltése</a>
    </div><hr class="footer_horizontal_rule">

    <p style="font-size: small;">&copy; 2024 CUNPO1 Webkönyvtárak záró. Minden jog fenntartva.</p>
</footer>

```

A lábléc szintén rendelkezik hivatkozásokkal, melyek mindegyike rendelkezik a `target="_blank"` attribútummal, mely biztosítja, hogy a hivatkozás vagy link egy új oldalon kerüljön megnyitásra a böngészőben.

A láblécben szintén egyéni stílusú tartalom szeparátor van, melynek osztály attribútuma „footer\_horizontal\_rule”.



Végül a lábléc bekezdés elemében (`<p>` tag) szerepel az úgynevezett *copyright* szimbólum (`&copy;`), mely egy speciális HTML karakter szimbólum.

## 1.3 – HTML elemek (div, span, p, címsorok, képek, táblázat, linkek, html5 elemei)

### Div

```
<div class="text_content_container">
    Weboldalamat az informatika és matematika iránti szenvedélyem inspirálta, és célom,
    hogy segítek mindenkinek megérteni ezeket a bonyolult, ám izgalmas területeket.
</div>
```

### Span

```
<span class="back">
    <aside>
        <strong>Végesség</strong><br>
        A feladat megoldásában mind a lépések számának, továbbá minden egyes
        lépésnek végesnek kell lennie!<br><br>
        <strong>Meghatározottság</strong><br>
        Minden lépésnek egyértelműnek kell lennie, hogy ne lehessen félreérteni!
        Ehhez pontos definiálásra van szükség.<br><br>
        <strong>Elvégezhetőség</strong><br>
        Fontos, hogy minden lépést valóban el lehessen végezni!
    </aside>
</span>
```

### Bekezdés (p)

```
<p style="font-size: small;">&copy; 2024 CUNPO1 Webkönyvtárak záró. Minden jog fenntartva.</p>
```

### Címsorok

```
<h1 id="hello_text" hidden>Üdvözlöm az algoritmusok és az informatikai matematika világában!</h1>
<h3>Téma</h3>
.
.
.
<h3>Inspiráció</h3>
```

## Képek

```
</a>  

```

A képek az „/images” mappában vannak elhelyezve.

## Táblázat (table)

```
<table id="algorithm_table" hidden></table>
```

*Megjegyzés: a tábla tartalma dinamikus hozzáfűzéssel valósul meg.*

A táblázat tartalma az algoritmusok listája 4 paraméter szerint:

- Elnevezés
- Kategória
- Rövid leírás
- Futás komplexitás

```
let titles = ["Elnevezés", "Kategória", "Rövid leírás", "Futás komplexitás"];
```

## Linkek

```
<a href="/images/TAOCP_book.png" target="_blank" title="The Art of Computer Programming"  
könyv'></a>  
    <a href="/images/DOP_book.jpg" target="_blank" title="A Discipline  
of Programming" könyv'></a>  
    <a href="/images/SOP_book.png" target="_blank" title="The Science  
of Programming" könyv'></a>
```

## Számozott lista

```
<ol>  
  <li>Az adott probléma meghatározása</li>  
  <li>Tervezés</li>  
  <li>Kezdőpont</li>  
  <li>Végpont</li>  
</ol>
```

## 1.4 – Űrlap elemek

### Az űrlap felépítése

1. „Adja meg a teljes nevét”: Szöveges beviteli mező, a felhasználótól várja a teljes nevét az űrlap kitöltésénél az azonosíthatóság érdekében.
2. „Adja meg az email címét”: szöveges beviteli mező, „*email*” típusú, a felhasználótól az email címét várja szintén az azonosíthatóság érdekében, illetve az esetleges kapcsolatfelvétel megvalósításának okából.
3. „Mikor született?”: Szöveges beviteli mező, a felhasználó születési dátumát várja, a jQuery UI JavaScript könyvtár „*datepicker*” metódusával dátum formátumú beviteli mező.
4. „Mi a kedvenc színe?”: Alapvetően szöveges beviteli mező, viszont egy kiegészítő JavaScript könyvtár „*colorpicker*” metódusával hexadecimális formában megadható egy szín (forrás feltüntetve).
5. „Hasznosnak gondolja ezt az oldalt?”: Radio gomb típusú beviteli mező, a jQuery UI JavaScript könyvtárának „*checkboxradio*” metódusával egy nem alapértelmezett radio gomb lesz, főképpen stílusbeli különbség figyelhető meg.
6. „Az alábbiak közül melyik algoritmus tetszik önnek a legjobban?”: Adatlista, a lista elemei dinamikusan egy script fájlban létrehozott algoritmus elnevezések, melyek közül a felhasználó választhat.
7. „Ajánlaná másnak is ezt az oldalt?”: Radio gomb típusú beviteli mező, hasonlóan a „Hasznosnak gondolja ezt az oldalt?” mezőhöz, két választási lehetőség van: Igen / Nem
8. „Véleménye szerint milyen mértékben tanulna hatékonyan egy weboldalról?”: Range (tartomány) típusú beviteli mező, [0; 10] intervallumon egész számként kifejezve kérdezi a felhasználót a weboldalakon történő személyes tanulási hatékonyságról kapcsolatban.
9. „Mi a véleménye erről a weboldalról?”: Szövegmező típusú beviteli mező, a felhasználónak lehetősége van hosszabb visszajelzést küldeni az oldalról.

Az űrlap alsó részében található gomb típusú beviteli mező megnyomásával egy script kód ellenőrzi a megadott bemenetek helyességét. A gomb szövege: „Űrlap elküldése”.

Az esetleges hibás kitöltések vagy nem megfelelő formátumok esetén az érintett beviteli mezők stílusa megváltozik (piros keret, korall piros háttérszín), illetve egy üzenő panel jelenik meg, melyben piros szöveggel van feltüntetve a hiba /

hibák forrása és oka, ellenkező esetben a panel tartalma zöld betűszínnel tartalmazza az űrlap kitöltésének sikerességére utaló szöveget.

Az űrlaphoz tartozó teljes programkód:

```
<section class="form">
  <form>
    <div class="form_element" id="name_container_forms">
      <div class="form_element_label">Adja meg a teljes nevét:</div>
      <input class="form_element_content" id="name" name="name" type="text" checked="">
    </div>
    <div class="form_element" id="email_container">
      <div class="form_element_label">Adja meg az email címét:</div>
      <input class="form_element_content" id="email" name="email" type="email" checked="">
    </div>
    <div class="form_element" id="age_container">
      <div class="form_element_label">Mikor született:</div>
      <input class="form_element_content" id="age" name="age" type="text">
    </div>
    <div class="form_element" id="favcolor_container">
      <div class="form_element_label">Mi a kedvenc színe:</div>
      <input class="form_element_content" id="favcolor" name="favcolor" type="text">
    </div>
    <div class="form_element" id="is_useful_container_forms">
      <div class="form_element_label">Hasznosnak gondolja ezt az oldalt:</div>
      <fieldset>
        <legend>Válasszon egyet</legend>
        <label for="useful">Igen</label>
        <input class="form_element_content" id="useful" name="useful" type="radio">
        <label for="not_useful">Nem</label>
        <input class="form_element_content" id="not_useful" name="not_useful" type="radio">
      </fieldset>
    </div>
    <div class="form_element" id="favalgos_container_forms">
      <div class="form_element_label">Az alábbiak közül melyik algoritmus tetszik önnek a
legjobban?</div>
      <input list="datalist_forms" id="favalgos" name="favalgos">
      <datalist id="datalist_forms">
        <option value="-">Válasszon</option>
      </datalist>
    </div>
    <div class="form_element" id="recommend_container">
      <div class="form_element_label">Ajánlaná másnak is ezt az oldalt:</div>
      <label for="reommend">Igen</label>
      <input class="form_element_content" id="recommend" name="recommend" type="radio">
      <label for="not_recommend">Nem</label>
      <input class="form_element_content" id="not_recommend" name="not_recommend" type="radio">
    </div>
```

```

        <div class="form_element" id="learnlevel_container">
            <div class="form_element_label">Véleménye szerint milyen mértékben tanulna hatékonyan egy
weboldalról?</div>
            <div class="range_input_label_container">0</div><input class="range_input" id="learnlevel"
name="learnlevel" type="range" min="0" max="10" value="5"><div class="range_input_label_container">10</div>
            </div>
            <div class="form_element" id="opinion_container">
                <div class="form_element_label">Mi a véleménye erről a weboldalról?</div>
                <input class="form_element_content" id="opinion" name="opinion" type="textarea" checked="">
            </div><hr>
            <div id="btn_container_forms">
                <input id="send_btn_forms" type="button" value="Űrlap elküldése">
            </div>
        </form>
    </section>

```

## 1.5 - Video

Minden felsorolt algoritmushoz tartozik egy-egy rövid médiatartalom, melyek vezérlése egy absztrakt script fájl metódusain keresztül valósulnak meg (*multimedia\_controller.js*).

A script fájl a következő metódusokat tartalmazza:

- *initializeVideo()* → a globálisan létrehozott *current\_algorithm* változó *id* adatmezőjének értékével inicializálja a szintén globálisan létrehozott *video* változó referenciáját az adott médiatartalomra.
- *togglePlayPause()* → a *play()* és a *pause()* metódusok használatával a videó leállítható, illetve elindítható az aktuális pozíciótól. Ezt a váltakozást valósítja meg ez a funkció.
- *toggleFullScreen()* → teljes képernyő módra állítható a nézet a *requestFullscreen()* metódus használatával, ellenkező esetben az *exitFullscreen()* metódus kerül meghívásra.
- *restartVideo()* → újra indítja a videót az elejétől úgy, hogy a *video* változó *currentTime* adattagjának értékét 0-ra állítja, és elindítja a videót a *play()* funkcióval.
- *volumeUp()* → a videó hangerejét növeli meghatározott értékkel a '+' operátor használatával (*video.volume* adattag).
- *volumeDown()* → a videó hangerejét csökkenti meghatározott értékkel a '-' operátor használatával (szintén *video.volume* adattag).
- *speedUp()* → a videó gyorsaságát növeli meghatározott értékkel a '+' operátor használatával (*video.playbackRate*).

- *speedDown()* → a videó gyorsaságát csökkenti meghatározott értékkel a `'.'` operátor használatával (*video.playbackRate*).
- *muteUnmuteVideo()* → a videó hangerejét 0-ra állítja, vagy 0.5-re, ha némított állapotban kattint a gombra a felhasználó. Két speciális karakter váltakozik ekkor, melyek hexadecimális kódja: `&#128264` és `&#128263`

Mindegyik funkció tartalmaz egy *if* feltétel blokkot, melynek szerepe a null hivatkozás kiküszöbölése, tehát ha a *video* változó értéke null vagy undefined (üres, definiálatlan), akkor azzal nem végezhető el semmiféle függvény vagy adattag hívás.

A videóhoz tartozó kód:

```
$("#main_content_algorithms").append(`<h1 style="text-align: center;">Kapcsolódó
médiatartalom</h1><div class="multimedia_container"><video id="video_${current_algorithm.id}"
controls><source src="/multimedia/algorithm_${current_algorithm.id}_multimedia.mp4" type="video/mp4">A
videó nem támogatott.</video><div
class="multimedia_controller">${multimedia_controller_content}</div></div>`);
```

A *multimedia\_controller\_content* változó értéke:

```
<button class="start_pause" onclick="togglePlayPause();">Lejátszás/leállítás</button>
<button class="restart_video" onclick="restartVideo();">Lejátszás az elejétől</button>
<button class="full_screen" onclick="toggleFullScreen();">Teljes képernyő</button>
<div class="volume_settings">
  <div class="volume_label">Hangerő</div>
  <div class="volume_btn_container">
    <button class="volume_down" onclick="volumeDown();">-</button>
    <button class="volume_up" onclick="volumeUp();">+</button>
  </div>
  <button class="mute_btn" id="mute_btn_${current_algorithm.id}"
onclick="muteUnmuteVideo();">🔊</button>
</div>
<div class="speed_settings">
  <div class="speed_change_label">Lejátszási sebesség</div>
  <div class="speed_btn_container">
    <button class="speed_down_btn" onclick="speedDown();">«</button>
    <button class="speed_up_btn" onclick="speedUp();">»</button>
  </div>
  <button class="set_default_speed_btn" onclick="setSpeedDefault();">Alaphelyzetbe állítás</button>
</div>
```

## 2. Feladat – CSS

A public mappában található „/css” mappában helyeztem el a különböző oldalakhoz tartozó stílusfájlokat.

Az „*abstract.css*” stílusfájlt minden oldal tartalmaz, ez az a fájl, melyben az oldalak alapstílusait meghatároztam.

A fájl a „*:root*” (a dokumentum fa szerkezetének gyökér eleme) elemmel kezdődik, melyben definiáltam a sötét/világos (dark-mode/light-mode) mód változóit a különböző elemstílus attribútumokra tekintve.

```
:root
{
  --background-color-light: #e7d3d3;
  --background-color-dark: #2b2b2b;
  --text-color-light: black;
  --text-color-dark: white;
  --content-container-background-color-light: white;
  --content-container-background-color-dark: black;
  --text-content-container-border-color-light: black;
  --text-content-container-border-color-dark: #232D3F;
  --scroll-bar-track-background-color-light: #f1f1f1;
  --scroll-bar-track-background-color-dark: #3c0c0c;
  --scroll-bar-thumb-background-color-light: #4a534c;
  --scroll-bar-thumb-background-color-dark: #828683;
  --scroll-bar-thumb-hover-background-color-light: #642121;
  --scroll-bar-thumb-hover-background-color-dark: #f1f1f1;
  --multimedia-controller-background-color-light: lightgray;
  --multimedia-controller-background-color-dark: gray;
}
```

Minden oldalon egyéni görgetőt és görgősávot használtam, mely a stílus fájl folytatása:

```
::-webkit-scrollbar
{
  width: 0.8em;
  height: 1em;
}

::-webkit-scrollbar-track
{
  background: var(--scroll-bar-track-background-color-light);
}
```

```

}

::-webkit-scrollbar-thumb
{
    background: var(--scroll-bar-thumb-background-color-light);
}

::-webkit-scrollbar-thumb:hover
{
    background: var(--scroll-bar-thumb-hover-background-color-light);
}

```

A *webkit* egy böngészőmotor, ami objektumoka tartalmaz és böngészőfunkciókat valósít meg.

Példa a színtéma változtatására:

```

body.dark-mode ::-webkit-scrollbar-track
{
    background-color: var(--scroll-bar-track-background-color-dark);
}

```

A fenti stílus részletben látható, hogyha az adott elem a „*dark-mode*” osztály attribútummal rendelkezik, akkor megváltozik a háttérének színe. Ugyanez igaz a „*light-mode*” osztály attribútummal rendelkező elemekre is, például az oldal háttérének színe alapértelmezetten világos, de megváltozik a sötét mód aktiválásánál.

További elemek stílusai szintén szerepelnek a fájlban.

## 2.1 – style attribútumban formázás

```

<span class="back" style="display: flex; justify-content: center; align-items: center; font-size: 3vh;">
.
.
.
.
</span>

```

A fenti példakódban egy *span* elem elemeinek irányvonala kerül beállításra, illetve a szövegének igazítása horizontálisan és vertikálisan, valamint a szöveg betűinek mérete.



## 2.2 – azonosító alapján formázás, táblázat formázása

```
#algorithm_table
{
  margin-top: 2em;
  border: 0.7em solid black;
  border-collapse: collapse;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
```

A kódrészletben egy táblázat formázása látható a tábla azonosítójára való hivatkozással, ezt jelzi a '#' karakter.

A tábla felső „behúzása” (ismertebb nevén „*margin-top*”) értéke kerül beállításra, a tábla körvonala, illetve a táblacellák közötti hely megszüntetése a kereteinek eltüntetésével, majd végül a betűtípus.

## 2.3 – osztály alapján formázás

```
.algorithm_widget
{
  padding: 2em;
  border-radius: 1em;
  box-shadow: 0 0 0.5em rgba(0, 0, 0, 0.3);
  background: linear-gradient(-45deg, #ee7752, #e73c7e, #23a6d5, #23d5ab);
  background-clip: padding-box;
  animation: gradient 15s ease infinite;
}
```

A fenti stílus részletben osztály hivatkozással állítok be stílusokat, így ez a stílus alkalmazható minden olyan elemre, mely rendelkezik az „*algorithm\_widget*” osztály attribútummal.

Az elem belső behúzásának mértéke (*padding*), doboz árnyék effektus hatás (*box-shadow*), keretének lekerekítésének mértéke, háttérszíne, animációjának beállítása, háttér levágásának egysége kerül definiálásra.

## 2.4 – menü kialakítása

Navigációs sáv:

```
nav
{
  display: flex;
  flex-direction: column;
  position: fixed;
  top: 0;
  right: 0;
  margin-bottom: auto;
}
```

Az elemeinek irányítottsága oszlopszerű, az elem pozíciója fix, az elem és az azt tartalmazó szülőelem felső része közötti kihagyás 0, ugyanez az elem és a szülőelem jobb oldalrészére is igaz.

A menü sáv:

```
.menubar
{
  position: fixed;
  top: 0;
  left: -12.5%;
  width: 12%;
  height: 100%;
  z-index: 1001;
  background-color: #333;
  color: white;
  transition: left 0.3s ease;
  font-size: 2vh;
}
```

Pozíciója szintén fix, az elem és az azt tartalmazó szülőelem felső része közötti kihagyás 0, a *left* attribútum negatív, így a menü sáv „kilóg” az oldalról balra, magassága 100%, a láthatósági értéke 1001, tehát nagyrészt minden elemnél nagyobb, azaz nem „takarja” el semmi, háttérszíne sötétebb, szövegének színe fehér a jobb láthatóság érdekében, betűmérete 2%-a a *viewport* magasságának, illetőleg a csúszás animáció különböző paraméterei láthatók.

## 2.5 – Linkek formázása

A Kezdőlap oldalon található linkek formázása:

```
.history_of_algorithms_content a
{
  color: blue;
  text-decoration: none;
  transition: color 0.3s ease-out;
}

.history_of_algorithms_content a:hover
{
  color: purple;
  transition: color 0.3s ease;
}
```

A „*history\_of\_algorithms\_content*” osztály attribútummal rendelkező elemek (főleg személyek) hivatkozásainak stílusozása a következő:

A szöveg színe alapértelmezetten kék, a szöveg dekorációja semmi, tehát megszűnik az alapból aláhúzott szöveg stílus, az egér ráhúzásával megváltozik a szöveg színe (:*hover*), ami egyenletesen kerül átszínezésre (*transition*).

## 2.6 – Űrlap elemek formázása

```
.form_element
{
  display: flex;
  justify-content: left;
  align-items: center;
  margin-bottom: 2em;
}
```

Az űrlap elemek a „*form\_element*” osztály attribútummal rendelkeznek, melyben meghatároztam az elemeinek irányítottságát, az elemek vertikális, illetve horizontális elhelyezését, majd az alsó behúzás mértékét.

A beviteli mezőkhöz tartozó szövegcímek stílusai:

```
.form_element_label
{
  font-size: x-large;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
```

```
margin-right: 1em;  
min-width: 50%;  
max-width: 50%;  
}
```

A „/css” mappa tartalma:

```
# abstract.css  
# algorithms_table.css  
# algorithms.css  
# forms.css  
# introduction.css  
# jquery-ui.css  
# landingpage.css
```

### 3. Feladat - JavaScript, jQuery, JSON, AJAX

Minden HTML oldal tartalmazza az „*abstract.js*” fájlt, mely olyan változókat és funkciókat tartalmaz, mely mindegyik oldal megfelelő működéséhez elengedhetetlen. Ebben a script fájlban valósítottam meg a menu sáv működését, az inicializáláshoz szükséges eljárásokat, az oldalak témaváltásának JavaScript funkcióját, illetve az algoritmusok adatainak lekérését a *getJSON()* metódussal, mely a *fetchAlgorithms()* funkcióban kerül meghívásra.

Továbbá a fájl tartalmaz jQuery utasításokat, melyek animációkat valósítanak meg vagy CSS attribútumokat állítanak be.

Az „*algorithms\_data.json*” fájl adatainak lekérése:

```
async function fetchAlgorithms()
{
    const response = await fetch('/json/algorithm_data.json');
    const data = await response.json();
    return data.algorithms;
}
```

Mivel ez egy asszinkron (*async*) eljárás, és nagyobb hibák lehetnek abból, ha előbb szeretnénk elérni egy adattagját egy még nem létező vagy nem inicializált változónak, így fontos, hogy az ezzel kapcsolatos műveleteket csak azután kezdje el a program, miután az adatok lekérése és betöltése sikeres. Ezt a bönghészőnek az „*await*” kulcsszóval mondjuk meg.

Az alábbi kódrészletben látható, hogy a „*then()*” metódussal érhető el ez:

```
fetchAlgorithms().then((result) =>
{
    algorithms = result;

    for ( let i = 0; i < algorithms.length; i ++ )
    {
        $(".submenu_items").append(`<li><a
href="/html/algorithms.html?id=${algorithms[i].id}">${algorithms[i].name}</a></li>`);
    }
});
```

Tehát miután az adatok lekérése sikeres, írja ki a „*submenu\_items*” osztály attribútummal rendelkező almenü listába a különböző algoritmusok elnevezését.

Minden script fájl rendelkezik a `$(document).ready()` funkcióval, mely biztosítja, hogy a műveletek csakis az oldal betöltődésekor hajtódnak végre.

## 3.1 – form ellenőrzés

Az űrlap helyes kitöltésének ellenőrzése a „*forms.js*” script fájlban történik.

A `checkFormFields()` metódus akkor kerül meghívásra, amikor a felhasználó a felületen megnyomja az „Űrlap elküldése” gombot.

A funkció úgy működik, hogy a `document.querySelectorAll()` metódusával kigyűjti azokat az elemeket, melyek a függvény paramétereiben megadott típusoknak megfelelnek (a függvény paramétere tulajdonképpen egy típus lista)

```
let formElements = document.querySelectorAll('input[type="text"], input[type="email"],  
input[type="radio"], input[type="textarea"], select');
```

Ezután az algoritmus végig iterál a `formElements` változón, és ha az éppen aktuális elem tartalma üres vagy email esetén helytelen formájú, akkor ellátja egy osztály attribútummal, melynek neve „*invalid\_form\_fields*”.

Az ezen osztály attribútummal rendelkező elemek keretszíne, illetve háttérszíne megváltozik (piros háttér, korall színű keret), ezzel jelezve a felhasználónak a beviteli mező helytelen kitöltését.

Az email formájának ellenőrzését egy funkció keretein belül valósítottam meg, melynek neve `isValidEmail()`.

A funkció paraméterén keresztül kapott email cím helyességét reguláris kifejezésekkel ellenőrzi.

```
function isValidEmail(email)  
{  
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
    return emailRegex.test(email);  
}
```

## 3.2 – jQuery animáció

Az Algoritmus táblázat oldalon a tábla jQuery animációval jelenik meg.

Az ehhez tartozó programkód részlet:

```
$("#algorithm_table").show("drop", 2000);
```

Az animáció neve „*drop*”, időtartama 2000 ms, azaz 2 másodperc.

Ezek a *show()* metódus paraméterei.

## 3.2 – elemek kiválasztása html tag név alapján, osztály alapján, azonosító (id) alapján

Az „*abstract.js*” script fájlban a fejlécben szereplő kép (logo\_landingpage.png) címét (*title* attribútum) az alábbi programkód ellátja a „Vissza a kezdőlapra” szöveggel:

```
$(".header_img_container img").attr("title", "Vissza a kezdőlapra");
```

Szintén az „*abstract.js*” script fájlban a dokumentum betöltődését követően az alábbi programkód eltünteti az almenü elemeket osztály alapú kiválasztással:

```
$(".submenu").find(".submenu_items").hide();
```

A „*forms.js*” script fájlban az alábbi kód stílust alkalmaz a „*message\_container\_forms*” azonosítójú html elemre:

```
$("#message_container_forms").css("color", "green");
```

## 3.3 – új elem készítése

Az alábbi *for* ciklus az „*algorithms\_table.js*” script fájlban a *table\_content* változóhoz új cellákat fűz hozzá, majd a végén hozzáfűzi a táblához a tábla azonosítójára való hivatkozással:

```
for ( let i = 0; i < titles.length; i ++ )
{
    table_content += `<th style="background-
color:${title_colors[i]};">${titles[i]}</th>`;
}
.
.
.
$("#algorithm_table").append(table_content);
```

## 3.3 – beágyazott JSON fájl elkészítése

Elkészítettem egy beágyazott JSON fájlt a „*/json*” mappába, mely a különböző algoritmusok azonosítóját, kód azonosítóját, elnevezését, kategóriáját, leírását, programkódját, futási komplexitását, illetve részletesebb leírását tartalmazza.

*Megjegyzés: a fájl tartalmát méretének okából nem mellékeltem a dokumentációban.*

## 3.4 – JSON fájl megjelenítése

A JSON file megjelenítése több oldalon is megtörténik, részletesebben az Algoritmusok menüpont alatt az adott algoritmus megtekintésénél, illetve kevesebb paraméter szerint az Algoritmus táblázat oldalon.

Az „*algorithms\_table.js*” tartalmaz egy *getJSON()* metódust, mely **AJAX** hívással valósítja meg az adatok lekérését az „*algorithms\_data.json*” fájlból.

Az itt alkalmazott *getJSON()* metódus 3 *callback* funkciót tartalmaz, a *done()*, a *fail()* és az *always()* funkciókat.

Ha a művelet sikeresen befejeződött, a *done()* funkció kerül meghívásra, ha nem volt sikeres, akkor a *fail()* funkció, de minden esetben meghívásra kerül az *always()* funkció, mely jelen esetben üres, ugyanis a programnak nem kell különösebben semmit sem csinálnia az eljárás sikerességének függetlenül,



ugyanis a *fail()* metódusban a felhasználó értesülhet arról, ha a művelet sikertelen volt.

```
.fail(function()  
{  
    $("#algorithm_table").append("<tr><td>Nem sikerült betölteni az  
adatokat.....</td></tr>");  
})
```

## 4. Feladat – Server

A szerver létrehozását *node.js* használatával valósítottam meg.

A gyökérmappa tartalmazza a „*server.js*” fájlt, emellet a „*package-lock.json*” és a „*package.json*” fileok is itt helyezkednek el.

A szerver a 3000-es porton fut.

Az oldalak útvonalát a *path.join()* funkcióval valósítom meg.

# Struktúra szerkezet

<ul style="list-style-type: none"><li>▼ WEBPROG<ul style="list-style-type: none"><li>&gt; node_modules</li><li>&gt; public</li><li>{ } package-lock.json</li><li>{ } package.json</li><li>JS server.js</li></ul></li></ul>	<ul style="list-style-type: none"><li>▼ images<ul style="list-style-type: none"><li> dark_mode.png</li><li> DOP_book.jpg</li><li> Euclid's_algorithm.png</li><li> Euclidean_algorithm.gif</li><li> header.png</li><li> hide.png</li><li> light_mode.png</li><li> logo_landingpage.png</li><li> logo_original.png</li><li> logo.png</li><li> show.png</li><li> SOP_book.png</li><li> TAOCP_book.png</li></ul></li></ul>	<ul style="list-style-type: none"><li>▼ json<ul style="list-style-type: none"><li>{ } algorithm_data.json</li></ul></li></ul>
<ul style="list-style-type: none"><li>▼ css<ul style="list-style-type: none"><li># abstract.css</li><li># algorithms_table.css</li><li># algorithms.css</li><li># forms.css</li><li># introduction.css</li><li># jquery-ui.css</li><li># landingpage.css</li></ul></li></ul>		<ul style="list-style-type: none"><li>▼ public<ul style="list-style-type: none"><li>&gt; colorpicker-master</li><li>&gt; css</li><li>&gt; html</li><li>&gt; images</li><li>&gt; json</li><li>&gt; multimedia</li><li>&gt; script</li><li>JS jquery-3.7.1.js</li><li>JS jquery-ui.js</li><li>&lt;&gt; landingpage.html</li></ul></li></ul>
<ul style="list-style-type: none"><li>▼ script<ul style="list-style-type: none"><li>JS abstract.js</li><li>JS algorithms_table.js</li><li>JS algorithms.js</li><li>JS forms.js</li><li>JS introduction.js</li><li>JS landingpage.js</li><li>JS multimedia_controller.js</li></ul></li></ul>	<ul style="list-style-type: none"><li>▼ html<ul style="list-style-type: none"><li>&lt;&gt; algorithms_table.html</li><li>&lt;&gt; algorithms.html</li><li>&lt;&gt; forms.html</li><li>&lt;&gt; introduction.html</li></ul></li></ul>	<ul style="list-style-type: none"><li>▼ multimedia<ul style="list-style-type: none"><li> algorithm_0_multimedia.mp4</li><li> algorithm_1_multimedia.mp4</li><li> algorithm_2_multimedia.mp4</li><li> algorithm_3_multimedia.mp4</li><li> algorithm_4_multimedia.mp4</li><li> algorithm_5_multimedia.mp4</li><li> algorithm_6_multimedia.mp4</li><li> algorithm_7_multimedia.mp4</li><li> algorithm_8_multimedia.mp4</li><li> algorithm_9_multimedia.mp4</li><li> algorithm_10_multimedia.mp4</li><li> algorithm_11_multimedia.mp4</li></ul></li></ul>

## A futtatás eredményei

Elkészítettem egy rövid videót az oldalról működés közben, melyet feltöltöttem a Google Drive-ba.

[Ide kattintva](#) megtekinthető.

## Források

**Logo kép:**

<https://logo.com/editor/icons>

**Ikonok** (dark\_mode.png, light\_mode.png, show.png, hide.png):

<https://www.flaticon.com/search?word=show%20hide>

**„A Discipline of Programming” könyv kép:**

<https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcRiXER1T9HLMN-r8w84jWkEQGXM7amuPFJZY4ZqLYBnNDxffb2>

**„The Science of Programming” könyv kép:**

<https://media.springernature.com/w158/springer-static/cover-hires/book/978-1-4612-5983-1?as=webp>

**„The Art of Computer Programming” könyv kép:**

[https://m.media-amazon.com/images/I/81eZ5mt0TGL. AC\\_UF1000,1000\\_QL80 .jpg](https://m.media-amazon.com/images/I/81eZ5mt0TGL. AC_UF1000,1000_QL80 .jpg)

**Szín ötletek:**

<https://colorhunt.co/palette/0f0f0f232d3f005b41008170>

<https://paletton.com/#uid=50u0s0kvzPB6nVV8PW4OwiPZ8p9>

**jQuery UI keretrendszer karakter ikonok:** <https://api.jqueryui.com/theming/icons/>

**Szöveges forrás az algoritmusokról:**

<https://hwellkft.hu/marketing-szotar/algoritmus>

**Programozás szöveg forrás:**

<https://infotanseged.wordpress.com/category/programozas-tortenete/>

**jQuery UI color picker:**

<https://github.com/vanderlee/colorpicker>

**Program kódok formázásának segítése:**

<https://codebeautify.org/>

## Videók:

- **Lineáris keresés:** <https://www.youtube.com/watch?v=SByG6SZdPQw>
- **Bináris keresés:** <https://www.youtube.com/watch?v=E6IOrZUpvSE>
- **Buborék rendezés:** <https://www.youtube.com/watch?v=9I2oOAr2okY>
- **Kiválasztásos rendezés:**  
<https://www.youtube.com/watch?v=92BfuxHn2XE>
- **Kiválasztás:** [https://www.youtube.com/watch?v=yRErgiYyv\\_E](https://www.youtube.com/watch?v=yRErgiYyv_E)
- **K-középérték:** <https://www.youtube.com/watch?v=5I3Ei69I40s>
- **Bináris fabejárás:** <https://www.youtube.com/watch?v=L88rWs3xpyU>
- **N-ágú fabejárás:** <https://www.youtube.com/watch?v=LrP7hF9jDmk>
- **Elárasztásos kitöltés:**  
[https://www.youtube.com/watch?v=QlcwfGo8W\\_E](https://www.youtube.com/watch?v=QlcwfGo8W_E)
- **A\*:** <https://www.youtube.com/watch?v=19h1g22hby8>
- **Futáshossz-kódolás:** <https://www.youtube.com/watch?v=Yl50cJScObI>
- **Euklideszi algoritmus:**  
<https://www.youtube.com/watch?v=fwuj4yzoX1o>