

JEGYZŐKÖNYV

Számítógép Architektúrák

Féléves feladatok

HTML weblap készítése és shell script írása

Készítette: Erdélyi Péter

Neptunkód: JH3V7T

Dátum: 2023. november 28.

(Módosítva: 2023. december 05.)

TARTALOMJEGYZÉK

1. HTML WEBLAP KÉSZÍTÉSE	1
1.1. Feladatkiírás.....	1
1.2. A kezdőlap felépítése.....	2
1.2.1. Üdvözlő képernyő	2
1.2.2. A <head> elem.....	2
1.2.3. Main.css	3
1.2.4. A <body> elem	3
1.3. A tantárgyak weblapjainak felépítése.....	5
1.3.1. Automatika.html	5
1.3.2. Listák használata	5
1.3.3. Szöveg formázása	7
1.3.4. Videó lejátszása	7
1.3.5. Sebességváltó csúszka	8
2. SHELL SCRIPT ÍRÁSA.....	10
2.1. Feladatkiírás.....	10
2.2. Szükséges ellenőrzések.....	11
2.3. Utasítások végrehajtása	11

1. HTML WEBLAP KÉSZÍTÉSE

1.1. Feladatkiírás

Tervezzen meg egy HTML weblapot, használjon modern webfejlesztési elemeket (HTML4/HTML5, CSS, JavaScript). Készítsen egy statikus HTML4/HTML5 oldalt, amely az Ön által tanult/választott három tantárgy tematikáját mutatja be, majd illesszen be egy video fájlt egy oldalra.

Legyen egy kezdőlap, amely a három kurzus menüpontját tartalmazza, ill. a fejléc képet is tartalmazzon. A kezdőlapon egy-egy kurzus menüpontjára kattintunk, megjelenik az adott kurzus tematikának leírása és hozzá egy kép. Legyen lehetőség visszalépni a kezdőlapra.

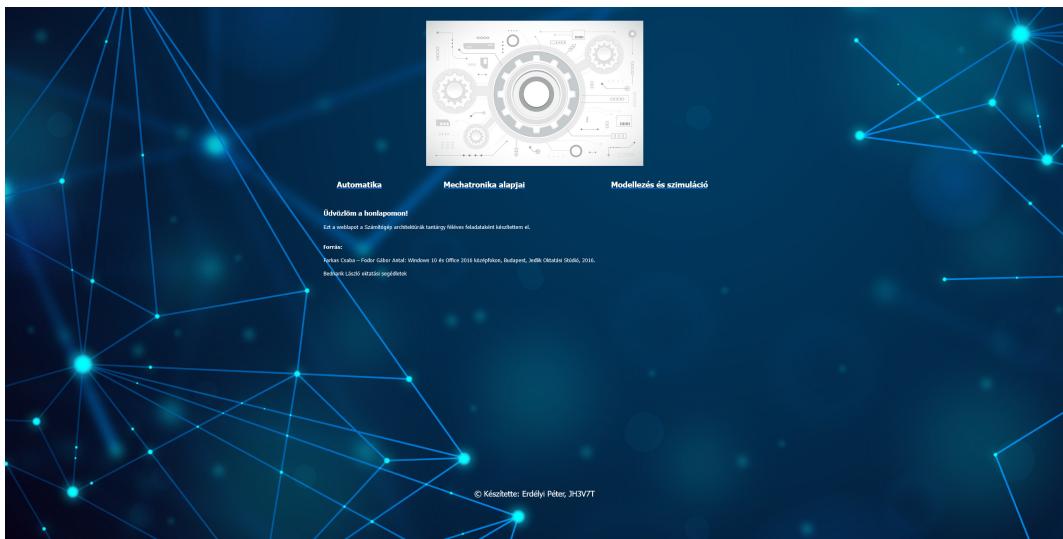
A lábléc tartalmazza a weblap készítőjének nevét és neptunkódját. Legyen egy mappa a képeknek.

Javaslat: a forrásfájl neve megegyezik az oldal nevével (pl.: kezdolap.html, index.html stb...).

1.2. A kezdőlap felépítése

1.2.1. Üdvözlő képernyő

A weblapra történő első látogatás alkalmával az alábbi képernyő (1. ábra) fogad:



1. ábra: Üdvözlő képernyő

Itt olvasható néhány, az oldallal kapcsolatos információ, többek között a weblap célja és az elkészítéshez használt források. Az üdvözlő képernyő felső részében elhelyeztem három menüpontot, melyek segítségével elnavigálhatunk az egyes tantárgyak oldalára.

1.2.2. A <head> elem

A HTML dokumentumok egyik legfontosabb alkotója a `<head>` elem, ami magyarul fejrész jelent. Az itt megadott elemek nem jelennek meg a weboldal részeként, viszont a működéshez szükséges metaadatokat tartalmaznak. A feladatban a 2. ábrán látható `<head>` rész látható.

```

<head>
  <title>Kezdőlap</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="css/main.css">
</head>

```

2. ábra: <head> tartalma

A `<title>` azonosítja az adott oldal címét, jelen esetben ez a kezdőlap. Karakterkódolásnak az UTF-8 formátumot jelöltetem ki. Továbbá konfiguráltam a `viewport` metarészt is, amely egy felhasználónak a weblap látható területét módosítja, és hozzájárul ahhoz, hogy az oldal eszköztől függően jelenítse meg a tartalmát. Ez a beállítás az oldal szélességét az eszköz szélességéhez igazítja, így a megtekintés nem okoz majd problémát.

A fejrész a `<link>` elemmel zárom. Célszerű a stílust külön fájlban kezelní, ezért a kezdőlapot (valamint a feladat többi HTML oldalának jelentős részét) hozzákapcsoltam a `main.css` stílusfájlhoz, melynek elérési útvonalát a `href` után adtam meg.

1.2.3. Main.css

A `main.css` fájl szolgál a stílus definiálására. Erre mutat példát a 3. ábra, ahol a `body` részben szerepel a háttérkép. Mivel a `body` minden HTML dokumentumban jelen van, ezért nem szükséges egyenként beállítani a háttérképet, hanem elegendő a stílusfájlt hozzájuk linkelni.

```
SemTaskJH3V7T > weblap > css > # main.css > ...
1   body {
2     |   background-image: url('../kepek/hatter.jpg');
3   }
```

3. ábra: Main.css (részlet)

1.2.4. A `<body>` elem

A `<body>` elem egy HTML oldal törzsét képezi, vagyis bekezdések, táblázatok, listák stb. tartoznak hozzá, ugyanakkor dokumentumonként csak egy törzsrész engedélyezett. A feladatban lévő `index.html` `<body>` részét vázolja a 4. ábra (következő oldal).

```

<body>

    <table align="center" style="table-layout:fixed; font-family:tahoma; font-size:14pt;
        color: #white" border = "0">

        <tr style="height:400px">
            <td colspan="3" style="width:1060px; text-align:center">
                <a href="kezdolap.html" target="keret"></a>
            </td>
        </tr>

        <tr style="height:30px; font-weight:bold">
            <td style="text-align:center">
                <a href="automatika.html" target="keret"><font style="color: #white;
                    text-decoration:none">Automatika</font></a>
            </td>
            <td style="text-align:center">
                <a href="mechatronika.html" target="keret"><font style="color: #white;
                    text-decoration:none">Mechatronika alapjai</font></a>
            </td>
            <td style="text-align:center">
                <a href="modszim.html" target="keret"><font style="color: #white;
                    text-decoration:none">Modellezés és szimuláció</font></a>
            </td>
        </tr>

        <tr style="height:700px">
            <td colspan="3" style="text-align:center">
                <iframe name="keret" src="kezdolap.html" width="100%" height="700" scrolling="auto"
                    frameborder="0" marginheight="10" marginwidth="20">
                    A böngészője nem támogatja a belső keretek használatát!
                </iframe>
            </td>
        </tr>

        <tr style="height:40px">
            <td colspan="3" style="width:1060px; text-align:center">
                © Készítette: Erdélyi Péter, JH3V7T
            </td>
        </tr>
    </table>
</body>

```

4. ábra: <body> tartalma (index.html)

A fenti ábrából jól látszik, hogy az index.html törzsrésze egy táblázat. A táblázat középre rendezett, és fix méretű sorokból (cellákból) tevődik össze. Legfelül található a kezdolap.jpg, ami egyúttal biztosítja a kezdőlaphoz való visszalépést. Ezt követi az a sor, melyet a három tantárgynak megfelelően három cellára osztottam fel. minden tantárgy rendelkezik egy saját weboldallal, ezért közülük bármelyikre kattintva átkerülünk az adott tantárgy oldalára. A táblázat utolsó sora pedig tartalmazza a nevemet és neptunkódomat.

Az <iframe> keretnek köszönhetően lehetőség van egy HTML weboldalt egy másikba ágyazni. Ezt a technikát alkalmazva, a táblázat 3. sorában jelenítem meg az egyes tantárgyak weblapjainak tartalmát.

1.3. A tantárgyak weblapjainak felépítése

A tantárgyak oldalai között csak tartalmi eltérés van, szerkezetileg viszont megegyeznek. Éppen ezért az alábbiakban ismertetem az `automatika.html` felépítését, a `mechatronika.html` és `modszim.html` ennek mintájára készültek.

1.3.1. Automatika.html

Az automatika menüpontra kattintva a következő, 5. ábrán vázolt oldalra érkezünk.



5. ábra: *Automatika.html* kinézete

Ezen az oldalon olvasható a tantárgy célja, az aláíráshoz szükséges feltételek, illetve a félévi tematika heti lebontásban. Jobb oldalt beillesztettem egy képet, ami egy PLC program részletét hivatott szemléltetni.

1.3.2. Listák használata

A tantárgyak weblapjainak egyik közös eleme a lista. A HTML három különböző lista alkotására ad módot: rendezett, rendezetlen és definíciós. A feladatban alkalmaztam egy rendezett és egy rendezetlen listát is, melyek közül az utóbbinak a felépítését illusztrálja a 6. ábra (következő oldal).

```
<ul>
  <li>Aktív részvétel a gyakorlatokon.</li>
  <li>Az évközi számonkérések sikeres (legalább 50%-os) teljesítése.</li>
</ul>
```

6. ábra: Rendezetlen lista felépítése

Egy rendezetlen lista az `` nyitóelemmel kezdődik, és az `` zárótaggal végződik. A közöttük lévő részben helyezkednek el az `` listaelemek.

Az ütemterv egy rendezett listából áll, melynek minden listaeleme tartalmazza a „hét: ” kifejezést. Ezt a hatást a `:before` szelektor felhasználásával értem el (7. ábra). A CSS szelektorjai mintákat keresnek a HTML elemek között, és a saját szabályaink szerint határozzák meg a stílust. Maga a `:before` az őt megelőző elemek elő illeszt be közvetlenül valamit.

```
ol {
  line-height: 150%
}

.ütemterv li::before {
  content: "hét: ";
  border-radius: 1px;
  padding-inline: 1px;
  margin-inline-end: 1px;
}
```

7. ábra: Ütemterv lista formázása

Mivel a `` listaelem nemcsak a rendezett, hanem a rendezetlen tagok előtt is előfordul, ezért őket meg kell különböztetni egymástól. A szétválasztáshoz létrehoztam az `ütemterv` osztályt, amit a lista fejlécébe beírva, csak az ütemterv listaelemeinek stílusa fog a fentiek szerint megváltozni. Az olvashatóság javítása érdekében pedig a vonalmagasságot 1,5-szeresre állítottam.

1.3.3. Szöveg formázása

A HTML szöveg formázása stílusfájlok mellett a dokumentumban elhelyezett `<style>` attribútummal is elvégezhető.

```
<br><br>

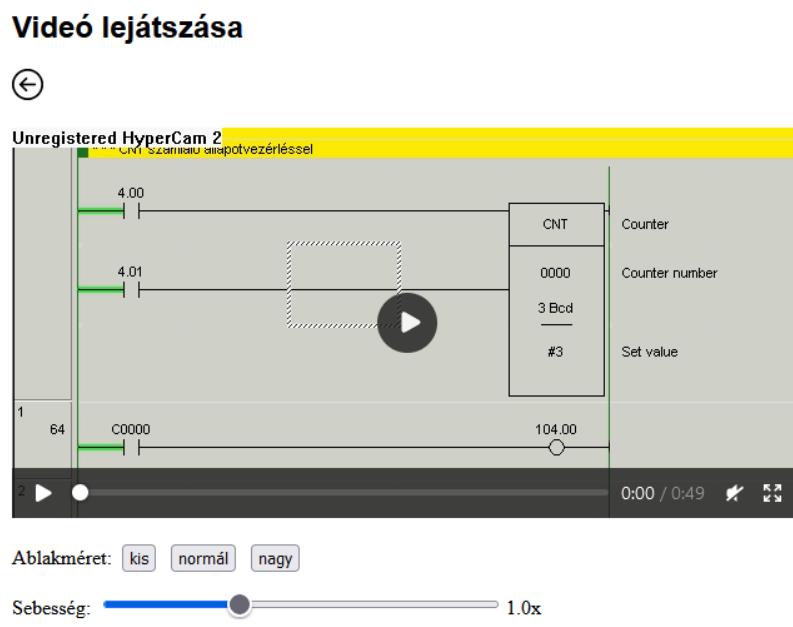
<font style="font-size:12pt"><u>Aláírás feltételei</u>:</font>
```

8. ábra: Szöveg formázása

Például az „Aláírás feltételei:” szövegrész (8. ábra) betűméretét 12-esre állítottam, az aláhúzást az `<u>` elemmel eszközöltetem. Új sorok közbeiktatására a `
` elemet használtam.

1.3.4. Videó lejátszása

Az automatika oldalon található a PLC mintavideo linkje. Ezt megnyitva átkerülünk a `video.html` weblapra, melyet a 9. ábra demonstrál. A nyílra lépve bármikor visszatérhetünk a kezdőlapra.



Forrás: Bólkény Ildi oktatási segédletek

9. ábra: Videó lejátszása (`video.html`)

Videók beillesztése a <video> elemmel történik. A <source> részben kell megadni a videó elérési útvonalát, aminek külön mappát hoztam létre. A <type> a videó formátumának leírására szolgál, ebben a feladatban a PLC mintavideo formátuma mp4. A fájl felbontásának méreteit külön számszerűsítettem, valamint írtam egy tartalék üzenetet arra az eshetőségre, ha a böngésző nem támogatja a videók lejátszását.

A videólejátszó a <controls> attribútum hozzáfűzésével alapvető funkciókat kap, mint pl. indítás, megállítás, hangerő-szabályozás. Az ablakméret kiválasztására a teljes képernyőre váltáson kívül három gombot helyeztem el, melyek kattintás hatására módosítják a video szélességét.

1.3.5. Sebességváltó csúszka

Sebességváltásra a video alatti csúszka vehető igénybe. A csúszka működésének értelmezéséhez tekintsük a 10. ábrát.

```
<div style="max-width: 632px;">
  <label for="csuszka" style="font-size:12pt;">Sebesség: </label>
  <input type="range" min="0.5" max="2" step="0.1" value="1" id="csuszka"
    oninput="modositErtek(value)" onchange="modositSebesseg()" style="height: 12px; width: 50%;"
    <output for="csuszka" id="playbackRate" style="font-size:12pt;">1.0</output>x
</div>
```

10. ábra: Sebességváltó csúszka

Az <input> type="range" egy tartományon belüli értékválasztást valósít meg. A feladatban az alsó határ 0,5 (lassú lejátszás), míg a felső határ 2 (gyors lejátszás), közöttük pedig 0,1-es léptékkal lehet mozogni. A csúszka alapértelmezetten 1-es értékkel jelenik meg, tehát a video normál lejátszási sebességgel fog indulni.

A sebesség dinamikus változtatásáról egy JavaScript program (11. ábra) gondoskodik, amely a <script> elem után foglal helyet. A csúszka mozgatásával meghívom a modositErtek(value) függvényt, amely a value értékétől függően változtatja a kijelzett sebességszorzót. Mivel ezt egy oninput eseményhez rendeltem hozzá, ezért az értékváltoztatás azonnal

bekövetkezik. Ezzel ellentétben a `modositSebesseg()` függvény csak akkor hajtja végre a sebességmódosítást, amint a felhasználó elengedte a csúszkát.

```
<script>

    var videom = document.getElementById('videom');

    function modositErtek(sebessegKijelzett) {
        document.querySelector('#playbackRate').value = sebessegKijelzett;
    }

    function modositSebesseg() {
        videom.playbackRate = document.querySelector('#playbackRate').value;
    }

</script>
```

11. ábra: Dinamikus sebességváltás

2. SHELL SCRIPT ÍRÁSA

2.1. Feladatkiírás

Az MVK Zrt. elérhetővé tesz egy szabványos GTFS adatbázist a fejlesztők számára, hogy a *menetrendi adatokat* a saját applikációba tudják integrálni. Írjon egy shell script fájlt, amely *letölți ezt az adatbázist, és kilistázza belőle azokat az utakat, amelyek a Lakóhelyről indulnak, ill. a Lakóhelyre érkeznek!*

Parancsok: *wget, unzip (kitömörítésre), cat, grep*

Szükséges ellenőrzések!

Ha a letöltendő fájl már egyszer le volt töltve, akkor az újbóli letöltés előtt törölje az előző verziót! Ha egy mappába már korábban ki lett tömörítve a letöltött állomány, akkor az újbóli kitömörítés törölje a mappa tartalmát! Ha nincs ilyen menetrendi adatok, akkor Miskolc várost használják a feladat elkészítéséhez!

2.2. Szükséges ellenőrzések

A shell script forráskódja a 12. ábrán olvasható.

```
jh3v7t@jh3v7t-VirtualBox:~/JH3V7T_szgyak/feladat_shell
jh3v7t@jh3v7t-VirtualBox:~/JH3V7T_szgyak/feladat_shell$ cat lakohely_gtfs.sh
#!/bin/bash

if [ -e volanbusz_gtfs.zip ]; then
    rm volanbusz_gtfs.zip
fi

if [ -d volanbusz_gtfs ]; then
    rm -r volanbusz_gtfs
fi

wget "https://gtfs.kti.hu/public-gtfs/volanbusz_gtfs.zip"
unzip volanbusz_gtfs.zip -d volanbusz_gtfs
cat volanbusz_gtfs/routes.txt | grep "Felsőtárkány"
jh3v7t@jh3v7t-VirtualBox:~/JH3V7T_szgyak/feladat_shell$
```

12. ábra: Shell script forráskód

Feltételek megfogalmazásához az `if-else` szerkezet használatos, melynek lezárását az `fi` kifejezéssel kell jelezni. Először is ellenőrizendő a fájl létezése, melyet az `-e` kapcsolóval oldottam meg, törléshez pedig az `rm` parancsot adtam ki. Következő lépésben a jegyzék létezését vizsgáltam, melyet a `-d` kapcsolóval valósítottam meg, itt viszont a törlési lépés kiegészül egy `-r` -el, hiszen a jegyzék teljes tartalmáról van szó.

2.3. Utasítások végrehajtása

A szükséges feltételek teljesülése után a script elvégzi az utasításokat. A `wget` parancssal letöltöm az adatbázist. Ezt követően a `.zip` állományt kitömörítem az `unzip` segítségével. Végezetül pedig egy összetett utasításban kilistázom a lakóhelyi útvonalakat. A kiolvasás bemenetét átirányítva kiszűröm a „Felsőtárkány” járatokat. A shell script lefutását szemlélteti a 13. ábra, melyen jól látszik, hogy a program három találatot eredményezett.

The screenshot shows a terminal window with the following command and its output:

```
jh3v7t@jh3v7t-VirtualBox:~/JH3V7T_szgyak/feladat_shell$ bash lakohely_gtfs.sh
--2023-12-05 14:38:24-- https://gtfs.kti.hu/public-gtfs/volanbusz_gtfs.zip
Resolving gtfs.kti.hu... 172.67.68.22, 104.26.13.249, 104.26.12.249, ...
Connecting to gtfs.kti.hu (gtfs.kti.hu)|172.67.68.22|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 105844694 (101M) [application/zip]
Saving to: 'volanbusz_gtfs.zip'

volanbusz_gtfs.zip          100%[=====] 100,94M  56,4MB/s   in 1,8s
2023-12-05 14:38:26 (56,4 MB/s) - 'volanbusz_gtfs.zip' saved [105844694/105844694]

Archive: volanbusz_gtfs.zip
  inflating: volanbusz_gtfs/feed_info.txt
  inflating: volanbusz_gtfs/agency.txt
  inflating: volanbusz_gtfs/shapes.txt
  inflating: volanbusz_gtfs/routes.txt
  inflating: volanbusz_gtfs/stops.txt
  inflating: volanbusz_gtfs/trips.txt
  inflating: volanbusz_gtfs/stop_times.txt
  inflating: volanbusz_gtfs/calendar.txt
  inflating: volanbusz_gtfs/calendar_dates.txt
21_21_3405_1,3405,"Füzesabony – Eger, Felsőváros – Felsőtárkány",200,https://www.volabusz.hu/hu/menetrendek/vonal-lista/vonal?menetrend=3405,F6AC00,152B32,regional
21_21_3414_1,3414,Maklár – Andornaktálya – Eger – Felsőtárkány,200,https://www.volabusz.hu/hu/menetrendek/vonal-lista/vonal?menetrend=3414,F6AC00,152B32,regional
21_21_3435_1,3435,Füzesabony – Eger – Felsőtárkány,200,https://www.volabusz.hu/hu/menetrendek/vonal-lista/vonal?menetrend=3435,F6AC00,152B32,regional
jh3v7t@jh3v7t-VirtualBox:~/JH3V7T_szgyak/feladat_shell$
```

13. ábra: Script lefutása