



DIGITÁLIS KULTÚRA

13. A HTML/CSS PROGRAMOZÁS ALAPJAI

Összeállította: Kolman Krisztián

Tartalomjegyzék:

1. Bevezetés	3
2. A HTML dokumentum elvi felépítése.....	3
3. Karakterformátumok a HTML dokumentumban	5
3.1. Alap karakterformázási lehetőségek.....	5
3.2. További betűformázási lehetőségek a betűtípus, szín, és a méret beállítása	6
4. Bekezdések a HTML dokumentumban	7
5. A HTML dokumentum címszintjeinek felépítése	8
6. Különböző listafomátumok a HTML dokumentumban.....	9
7. Tabulátorok a HTML dokumentumban.....	10
8. Választóvonal alkalmazása HTML dokumentumban	11
9. Kereszthivatkozások HTML dokumentumok között	12
10. A HTML dokumentum szövegtestének felépítése	14
11. Képek elhelyezése a HTML dokumentumban.....	15
12. A HTML formátum táblázatai.....	18
13. A HTML dokumentumablak felosztása (FRAME-ek)	22
14. Kérdőívek a HTML dokumentumban	24
15. Összefoglaló példa	26
16. Mi az a CSS?	32
17. Példa egy egyszerű css kódra	33
18. Még egy példa a css kódolásra.....	34
19. Utolsó példa a css kódolásra.....	37

1. BEVEZETÉS:

- A HTML formátumú dokumentumok megtekintése egy ún. "World Wide Web **böngészőprogrammal**" lehetséges.
- A HTML dokumentum-formátumot tekinthetjük az ún. hyper-text egyik megvalósítási formájának is.
- A **HTML dokumentum** egy olyan **szövegfájl**, amely a szövegen kívül tartalmaz ún. "HTML-tag"-eket - formázóutasításokat -, valamint megjelenítendő objektumokra történő hivatkozásokat is. Ezek a HTML formázóutasítások (más szóval: parancsok, elemek) befolyásolják a dokumentum megjelenését, kapcsolatait. Ezeket az utasításokat a böngészőprogram értelmezi és végrehajtja. Ezen okból a **formázóutasítás mindig megelőzi azt a részét a dokumentumnak, amelyre vonatkozik.**
- A dokumentumkészítéshez használható HTML utasítások köre állandóan bővül, a nyelv fejlődik. A szabványosítás csak lassan követi a fejlődést. Ezért nem minden böngészőprogram tudja a HTML utasítások mindegyikét értelmezni. Egy böngésző, ha számára értelmetlen utasítással találkozik, akkor kihagyja, így nem okoznak problémát az újabb keletű - még szabványosítatlan - utasítások a régebbi kiadású WWW-böngészőknek sem.
- Sajnos a fentiek miatt ugyanazt a dokumentumot két különböző program nem biztos, hogy azonos formában fogja megjeleníteni. Más oka is van ennek. A WWW-n kalandozónál kicsi a valószínűsége annak, hogy rendelkezésére áll ugyanaz a betűtípus, mint a WWW-oldalt fejlesztőnek. Vagy képek esetén semmi garancia nincs arra, hogy minden böngészőprogram ugyanazon felbontásban és színszámmal tudja megjeleníteni a képet. És így tovább ...
- A HTML-ben mégis az a nagyszerű, hogy nagymértékben megközelíti a platformfüggetlenséget. Egy HTML dokumentum - ha nem is azonos módon - mindenki számára megtekinthető.



Néhány szóban a HTML alapvető szabályairól:

- A HTML dokumentum normál szövegfájl. Bármely szövegszerkesztővel létrehozható, ill. módosítható, amely nem használ különleges fájlformátumot vagy ha létezik TEXT formátumú mentési lehetőség benne.
- **A HTML utasításokat a szövegben < és > jelek közé kell zárni. Ezeket az utasításokat TAG-eknek hívjuk. Egy-egy utasítás - HTML parancs, HTML elem - hatását általában a záró utasításpárja szünteti meg, amely megegyezik a nyitó utasítással, csak a / jel vezet be (természetesen a < és a > jelek között). Az utasítások nagy része opcionális elemeket is tartalmazhat, melyek csak a nyitóutasításban szerepelhetnek, a záróban nem. Az opciók értékadásánál az idézőjel nem mindig kötelező, csak ajánlott. A HTML utasítás kulcsszavaiban nem különböztetjük meg kisbetűket és nagybetűket.**



2. A HTML DOKUMENTUM ELVI FELÉPÍTÉSE:

- **Minden HTML formátumú szövegfájl a <HTML> utasítással kezdődik és a </HTML> záróutasítással végződik. Ezen elemek közé kell zárni a teljes dokumentumot - formázóutasításokkal és hivatkozásokkal együtt.**
- **A HTML dokumentumot két részre lehet bontani a fejlécre és dokumentumtörzsre.** (Egy harmadik rész lehet a keretek definíciója.)
- A dokumentumot a **fejlécelemek** vezetik be, melyek kezdetét a **<HEAD>** utasítás jelzi. A fejlécelemek között szokás a dokumentumcím megadni, mely címet a **<TITLE>** és a **</TITLE>** utasítások közé kell zárni. A fejlécet a **</HEAD>** utasítás zárja. Ezt a részét a dokumentumnak általában az ablak címsorában jelenítik meg a böngészőprogramok.
- A **dokumentumtörzs** - amit voltaképpen a WEB-böngésző meg fog jeleníteni - a fájl **<BODY>** és **</BODY>** utasítások közötti része. Ezen elemek között kell elhelyezni mindent: a szöveget, hivatkozásokat, képeket, stb. (A keretek és a JavaScript kódok kivételével!)
- Az utasításokat egyszerűen a Windows operációs rendszer, **Jegyzetömb** nevű programjába írjuk be. Arra kell vigyázni, hogy ha végeztünk az utasítások begépelésével, akkor **valami.html** néven mentjük el! A mentésnél figyelni kell arra, hogyha egy tárhelyre szeretnénk feltenni a későbbiekben, akkor a megfelelő működés érdekében ékezet nélküli nevet adjunk a fájloknak!



Témakör: A HTML/CSS programozás alapjai

Az alábbi egyszerű példadokumentum tartalmaz fejléct, törzsében pedig kiírja a képernyőre a következő szöveget: „A tulajdonképpeni dokumentumtörzs következik ... Az első honlapom!”

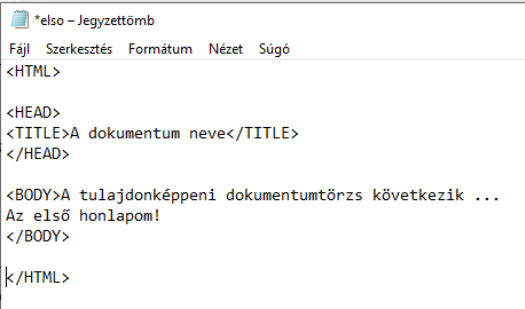
Gépelj be a következő utasításokat egy szövegfájlba, majd mentsd el **elso.html** néven!

```
<HTML>

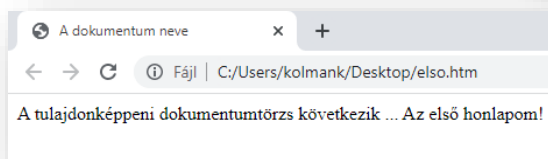
<HEAD>
<TITLE>A dokumentum neve</TITLE>
</HEAD>

<BODY>A tulajdonképpeni dokumentumtörzs következik ...
Az első honlapom!
</BODY>

</HTML>
```



Képernyőkép:



Az előző feladatban érdekesség az, hogy hiába írtam külön, attól még nem törte meg a szöveget. Azért, mert nem adtam rá utasítást. Létezik egy úgynevezett **sortörő tag** a **
**, ami ellentétben a többi tag-gel önmagában áll (nincs záróutasítás) a sor végén, ahol törni szeretnénk.

Feladat01:

Készíts weblapot melynek Feladat01 szöveg legyen a címsorában és a képernyőre a következő szöveget írja ki több sorban:

Első sor
Második sor
Harmadik sor
Negyedik sor
Ötödik sor

Az oldal neve **01szoveg.html** legyen

Az elkészített weblapot úgy tudod megnézni, hogy a mentett helyen kétszer rákattintasz az ikonra! Az ikon kinézete függ az elsődleges böngészőtől!

Amely lehet:



Ha változtatni szeretnél a kódon, vissza akarsz menni, szerkeszteni, akkor az ikonon jobb egér ->társítás -> Jegyzetömb!

Célszerű az asztalon létrehozni például egy HTML_prog nevű mappát , és minden feladatot abba menteni.

3.1 Alap karakterformázási lehetőségek:

A HTML formátumú szövegfájlban is használhatjuk a szövegszerkesztőkben megszokott karakterformátumokat. Az alábbi táblázat a formázás kezdő és záróutasítása között a mintát is tartalmazza. (Nem mindegyik működik a különböző böngészőkben.)

Kezdő elem	Ilyen betűformátumot eredményez	Záró elem
	Félkövér betűformátumot eredményez	
<I>	<i>Dőltbetűs formátumot eredményez</i>	</I>
<U>	<u>Aláhúzott formátumot eredményez</u>	</U>
<S>	Áthúzott formátumot eredményez	</S>
<TT>	Fixpontos betűket eredményez	</TT>
	<i>Kiemeli a szöveget</i>	
<CITE>	<i>Idézetekre használható</i>	</CITE>
<VAR>	<i>Változónevet jelöl</i>	</VAR>
	Ez is egy kiemelési lehetőség	
<CODE>	Kódknál használjuk	</CODE>
<SAMPLE>	Minták jelzésére használjuk	</SAMPLE>
<KBD>	Billentyűfelirat jelzése	</KBD>
<BQ>	Idézet megjelenítése	</BQ>
<BIG>	Nagyméretű betűformátumot eredményez	</BIG>
<SMALL>	Kisméretű betűformátumot eredményez	</SMALL>
_{	Alsóindexet eredményez	}
^{	Felsőindexet eredményez	}
<BLINK>	Villogó szöveget eredményez	</BLINK>
	A részleteket lásd lentebb	

Gépelj be a következő utasításokat, majd mentsd **masodik.html** néven!

Az alábbi példadokumentumban elvileg meg kellene jelennie a fenti karakterformátumoknak (azért elvileg, mert a megjelenő betűk formája a böngészőprogram beállításaitól függően nagymértékben különbözhet):

```

<HTML>
  <B>Vastag</B>
  <I>Dőlt</I>
  <U>Aláhúzott</U>
  <S>Áthúzott</S>
  <TT>Fixpontos</TT>
  <EM>Kiemelt</EM>
  <CITE>Idézet</CITE>
  <VAR>Változónév</VAR>
  <STRONG>Kiemelt</STRONG>
  <CODE>Kód</CODE>
  <SAMPLE>Minta</SAMPLE>
  <KBD>Billentyűfelirat</KBD>
  <BQ>Idézet</BQ>
  <BIG>Nagyméretű</BIG>
  <SMALL>Kisméretű</SMALL>
  <SUB>Alsóindex</SUB>
  <SUP>Felsőindex</SUP>
  <BLINK>Villogó</BLINK>
</HTML>

```

Képernyőkép:



3.2 További betűformázási lehetőségek a betűtípus, szín, és a méret beállítása:

A ``, `` utasításpárral direkt módon előírhatók a megjelenő szöveg betűinek a jellemzői. A **FACE** opciót nem szokás használni, mert nem valószínű, hogy minden rendszerben rendelkezésre áll pl. az *ARIAL CE FÉLKÖVÉR* betűtípus. A **COLOR** opció pontosan meghatározza a megjelenítendő szöveg színét. A **SIZE** opcióban egy számot megadva a betűméretet határozza meg direkt módon. (A **SIZE** opcióban előjeles szám is szerepelhet, ami az alapbetűtípushoz viszonyított méretet jelöl.)

Gépeld be a következő példát, és mentsd **harmadik.html** néven! Nézd meg, hogy hogyan néz ki a weblap!

```
<HTML>
  <FONT COLOR="#FF3333" SIZE=1>Kicsi piros</FONT>
  <FONT SIZE=2>N</FONT>
  <FONT SIZE=3>ö</FONT>
  <FONT SIZE=4>v</FONT>
  <FONT SIZE=5>e</FONT>
  <FONT SIZE=6>k</FONT>
  <FONT SIZE=7>v</FONT>
  <FONT SIZE=8>ő</FONT>
  <FONT FACE="Arial CE" SIZE=9>ARIAL CE</FONT>
  <FONT FACE="Symbol" SIZE=10>SYMBOL</FONT>
</HTML>
```

A képernyőkép:



Feladat02:

Gépelés előtt gondold végig, hogy hogyan fog kinézni a weblap. Majd gépeld be az utasításokat és mentsd **02alapform.html** néven!

```
<HTML>
  <b>első</b> <br>
  <u>második</u> <br>
  <i>harmadik</i> <br>
  <br>
  <br>
  <b><u>negyedik</u></b> <br>
  <br>
  <br>
  <br>
  <u><b><i>ötödik</i></b></u> <br>
</HTML>
```

Feladat03:

Az előző weblapot bővítsd ki úgy, hogy minden sornak legyen más a mérete és legyen más a színe! Aztán mentsd másként **03szinalapform.html** néven!

Feladat04:

Készíts olyan weblapot, melynek a címsorában a vezetékneved jelenik meg. Az oldalon pedig a harmadik sorban a SZIVÁRVÁNY szöveg 9-es betűmérettel, Verdana betűtípussal úgy, hogy a sötét színekből betűnként átmegy a világosokba! (pl.: fekete; sötétkék; ...; sárga). Az oldal neve legyen **04szivarvany.html**!

Feladat05:

Bővítsd ki az előző feladatot úgy, hogy az első 5 betűig növekedjen a betű mérete, majd ugyanabban a sorrendben csökkenjen! Az újabb weboldalt mentsd másként **05bov_szivarvany.html** néven!

Feladat06:

Készíts olyan weblapot melynek a fejlécében a „név” felirat szerepel, az oldalon pedig a vezeték- és keresztnéved jelenjen meg külön sorokban, más-más betűtípusokkal a következő sorrendben: Times New Roman, Arial, Arial Narrow, Arial Black, Courier New, Symbol. Mentse **06betutipus.html** néven!

4. BEKEZDÉSEK A HTML DOKUMENTUMBAN

Minden dokumentum, így a HTML formátumú dokumentum is, alapvetően bekezdésekre tagolódik.

A HTML fájlban a **bekezdések kezdetét a <P> utasítás jelzi** a böngészőprogram számára. (Tehát ezen elem nem az előző bekezdés végét, hanem a következő kezdetét jelzi, vagyis mindig megelőzi azt a bekezdést, amelyre vonatkozik!)

A bekezdés a következő bekezdés kezdetét jelző utasításig tart, tehát nincs záróutasításpárja a bekezdés elemnek. A legtöbb böngészőprogram két bekezdés között egy üres sort szűr be megjelenítéskor!

A bekezdés elem hordozhat magában a bekezdés stílusát meghatározó opciókat. A bekezdés igazítását a **<P ALIGN="hely">** formájú utasítással szabályozhatjuk. Az automatikus tördelést a **NOWRAP** opció tiltja meg a böngészőprogram számára. Amennyiben tördelhetetlen szóközt igényel a szöveg, akkor az egyszerű szóköz helyett alkalmazzuk a ** ** különleges karaktert.

Amennyiben egy bekezdésen belül mindenképpen új sort szeretnénk kezdeni, akkor a **
** utasítást kell használni. (Nincs zárópárja, ahogy az elején tárgyaltuk!)

Az alábbi példa tartalmazza az említett formázási lehetőségeket:

```

<P>Ez alapértelmezett (balra igazított) bekezdés.
<P ALIGN=left>Ez balra igazított bekezdés.
<P ALIGN=center>Ez középre igazított bekezdés.
<P ALIGN=right>Ez jobbra igazított bekezdés.
<P ALIGN=justify>Ez sorkizárt bekezdés lenne.
<P>
    <NOBR>Ez itt egy tördeletlen - jó hosszú - bekezdés,
    amelynek eltűnik a vége a semmiben
    .....</NOBR>
<P>
Itt egy sortörés elem található,
<BR>
melynek hatására új sorban folytatódik a szöveg,
és nem maradt&nbsp;ki&nbsp;üres&nbsp;sor.

```

Gépeld be, majd mentsd **negyedik.html** néven!

Feladat07:

Készítsd el a következő weblapot a minta alapján! Majd mentsd **07rendezes.html** néven!

Piros, középre, aláhúzott, Arial, 3-as betűméret;

Zöld, jobbra, dőlt, Arial Narrow, 2-es méret;

Ballra; zöld, 4-es méret, Courier New;

Sorkizárt, Times New Roman, 1-es méret sárga, félkövér;

Feladat08:

Készítsd el a következő weblapot a minta alapján! Majd mentsd **08rendezes.html** néven! Az első sor legyen piros és középre rendezett, Arial, a második félkövér, és zöld, és jobbra rendezett, szintén Arial!

11³⁰-16¹⁵

H₂O+NH₃

5. A HTML DOKUMENTUM CÍMSZINTJEINEK FELÉPÍTÉSE

A HTML formátumú szövegfájlban definiálhatunk címeket, ill. alcímeket hat szint mélységig. A legfelső szintű címet a **<H1 ALIGN="hely">** és a **</H1>** utasításpárral kell közrezárni. A második szintet a **<H2 ALIGN="hely">** és a **</H2>** utasítások határolják, és így tovább a hatodik szintig ...

Minden szint más-más betűformátumban jelenik meg a dokumentumban, a böngészőprogram beállításától függően. A címek igazítását szabályozza az **ALIGN** opció, melynek lehetséges értékei: *left, center, right*. Amennyiben túl hosszú a cím, de egy sorosnak kell maradnia, akkor a **NOWRAP** opció megakadályozza a cím betördelését több sorba.

Különleges címnek tekinthető a címszalag, melyet a **<BANNER>**, **</BANNER>** utasításpárral kell kijelölni. Tulajdonsága, hogy nem szkrrollozódik a dokumentum többi részével ellentétben.

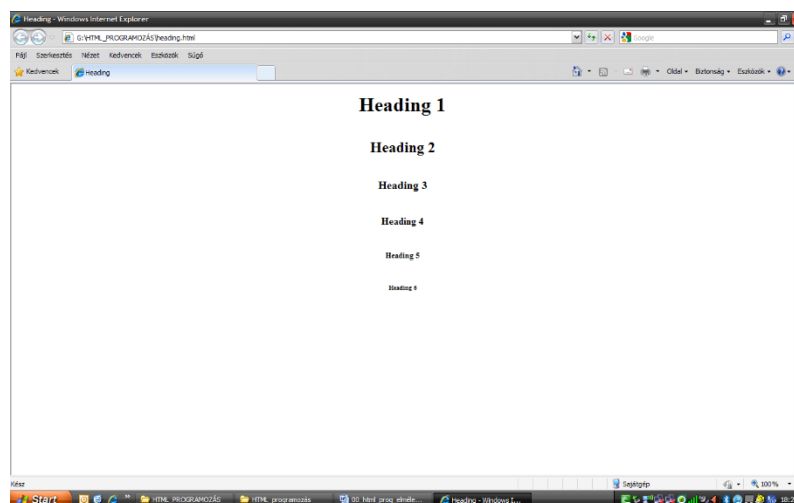
A címek csak a szemlélő számára keltik a tagoltság érzetét, a valóságban nem tagolják fizikailag szakaszokra a dokumentumot. Ezt a tagolást a **<DIV CLASS="osztály" ALIGN="hely">**, **</DIV>** utasításokkal lehet meghatározni, ahol a **CLASS** opció sorolja a megfelelő SGML osztályba a szakaszt, az **ALIGN** pedig a szakasz igazítási formátumát írja elő. Az automatikus tördelést itt is megakadályozza a **NOWRAP** opció, ez esetben a szakasz tördelését a **<P>** vagy a **
** utasításokkal lehet szabályozni.

Gépelj be a mintapéldát és mentsd **cim.html** néven!

```
<BANNER>Címszalag</BANNER>
<H1 ALIGN="left">Legfelső szintű címsor</H1>
<H2 ALIGN="center">Második szintű alcímsor</H2>
<H3 ALIGN="right">Harmadik szintű alcímsor</H3>
<H4 NOWRAP>Negyedik szintű alcímsor</H4>
<H5>Ötödik szintű alcímsor</H5>
<H6>Hatodik szintű alcímsor</H6>
<DIV ALIGN="center">
Ez egy szakasz, melyben a szöveg elvileg azonos módon
- középre igazítva - kezelendő.
</DIV>
```

Feladat09:

Készítsd el a következő egyszerű weblapot, majd mentsd **09heading.html** néven!



Feladat10:

Készítsd el az iskola csengetési rendjét a következő módon: középre rendezett növekvő méretű címsorokban, más-más színnel, egy más alá! Legyen az összes betű félkövér, a percek felső indexben, és aláhúzottak legyenek! Mentse a kész weblapot **10orarend.html** néven!

6. KÜLÖNBÖZŐ LISTAFOMÁTUMOK A HTML DOKUMENTUMBAN

- A HTML formátumú szövegfájlban használhatunk listákat is, amelyek szövegszerkesztőbeli megfelelői a felsorolások és bajuszos bekezdések.
 - A számozott bekezdések (felsorolások) megfelelője a számozott lista, az ún. "bajuszos" bekezdések megfelelője pedig a számozatlan lista.
 - A harmadik lista típus pedig a leíró lista, ahol az egyes lista elemekhez tartozhat egy hosszabb leírás is.
 - A számozott listát az **** és az **** utasítások közé kell zárni. A számozatlan listát pedig az ****, **** utasításpár közé. Mindkét típusú listában használhatjuk a listafejléct, melyet az **<LH>** utasítás vezet be - az **</LH>** pedig zár! Mindkét listatípusban a listák sorai az **** utasítással kezdődnek - és nem használatos a lezáró párjuk!
 - A fentiekből közvetkezik, hogy a listások egy **** utasítástól a következő **** utasításig, ill. a lista záróeleméig tartanak.
-
- Számozott lista esetében a kezdő sorszám közvetlenül megadható az **<OL SEQNUM="szám">** formájú kezdőutasítással. Másik lehetőség, hogy egy előzőleg definált lista számozása folytatható az **<OL CONTINUE>** kezdőutasítás használatával. Egyébként az **** utasítás 1-től kezdi a lista tagok számozását.
 - De legegyszerűbben a típust, és kezdősorszámot, így adhatjuk meg: **<ol type="1" start="34">**. Kisbetűs felsorolás: **<li type=a>**, nagybetűs felsorolás: **<li type=A>**
-
- A számozatlan listák kezdőeleme is hordozhat formázóinformációkat. Az **<UL SRC="fájlnev.kit">** formájú utasítás például a listasort megelőző bajuszként a megadott képfájlt használja. Az **<UL DINGBAT="karakter">** a megadott bajuszkaraktert alkalmazza. Az **<UL WRAP="irány">** pedig többoszlopos listák esetén az igazítás formáját határozza meg. (A **WRAP** opció a *horiz* és a *vert* értékeket veheti fel.
 - A számozatlan listák speciális - külön HTML utasításokkal létrehozható - fajtái a könyvtárlista és a menülista. A könyvtárlista típus a **<DIR>** utasítással kezdődik és a **</DIR>** utasításra végződik. A menülista pedig **<MENU>** és a **</MENU>** utasításokkal határolt. Ezek a listaformák a normál számozatlan listáktól mindössze annyiban különböznek, hogy a könyvtárlista tagjai 20 karakteresnél, a menülista tagjai pedig egy sorosnál nem lehetnek hosszabbak és nincs "bajuszuk".
-
- A leíró listát a **<DL>** és a **</DL>** utasítások közé kell zárni. A lista fejléc megadása azonos az többi listatípussal. A listák egyes alkotóelemeinek kezdetét a **<DT>** utasítás jelzi, az ehhez tartozó leírás kezdetét pedig a **<DD>** utasítás határozza meg. Nincs egyik utasításnak sem záró párja, ezért a lista tag a **<DT>** elemtől a **<DD>**-ig, a hozzá tartozó leírás pedig a **<DD>** elemtől a következő **<DT>**-ig tart.
-

A következő példa egy egyszerű felsorolást tartalmaz! Gépeled be és mentsd **felsorolás.html** néven!

```
<html>
<head>
<title>felsorolás</title>
</head>
<body>
<br>
<ul>
<li>Első elem
<li>Második elem
<li>Harmadik elem
<li>Negyedik elem
</ul>
</body>
</html>
```

Feladat11:

Az előző példaprogramot alakítsd át számozásra, majd mentsd el másként **11szamozas.html** néven!

Feladat12:

Készítsd el a következő weblapot, majd mentsd **12fels_szam.html** néven!

- Első elem
- Második elem
- Harmadik elem
- Negyedik elem

- A. Első elem
- B. Második elem
- C. Harmadik elem
- D. Negyedik elem

1. Első elem
 - a. Első alelem
 - b. Második alelem
 - c. Harmadik alelem
 - d. Negyedik alelem
 2. Második elem
 - Első elem
 - Második elem
 - Harmadik elem
 - Negyedik elem
 3. Harmadik elem
 4. Negyedik elem
-

7. TABULÁTOROK A HTML DOKUMENTUMBAN

A HTML formátumú fájlban a tabulátor pozíciókat a `<TABSTOP ID="azonosító">` utasítás jelzi.

A szöveget a tabulátorpozícióhoz igazítani a `<TAB TO="azonosító">` utasítással lehet. Ha még az igazítás formáját is meg kívánjuk adni, akkor a `<TAB TO="azonosító" ALIGN="hely">` alakú utasítást kell használni, ahol *left*, *center*, *right*, *decimal* típusú tabulátorokat jelölhet a "hely" paraméter. (Tizedes tabulátor alkalmazásánál megadható a **DP** opcióval az a karakter, amelyhez az igazítás történjen.)

Az alapértelmezett tabulátortávolságot `<TAB INDENT="szám">` elem határozza meg.

A tabulátorokat - a tapasztalatok szerint - kevés böngészőprogram tudja értelmezni, ezért a táblázat vagy előre formázott szöveg alkalmazása javallott a tabulátorok helyett ...

Az alábbi példa tartalmazza az említett tabulálási lehetőségek egy részét:

```
<P>Szöveg, <TABSTOP ID="elso">szöveg folytatás<TABSTOP ID="masodik"> ...  
<BR>  
<TAB TO="elso" ALIGN="center">Igazítás.  
<TAB TO="masodik" ALIGN="decimal" DP=",">48,125  
<P><TAB INDENT=32> Az alapértelmezett távolság: 32.  
<BR><TAB>Ez a 32. karakternél kezdődik.  
<BR><TAB>Ez pedig a 64. karakternél.
```

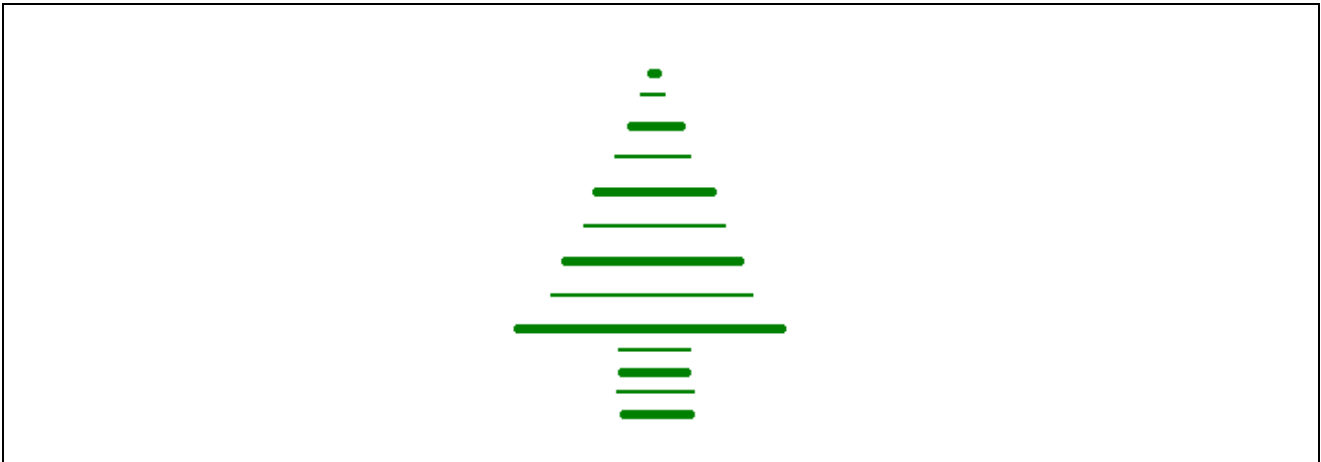
8. VÁLASZTÓVONAL ALKALMAZÁSA HTML DOKUMENTUMBAN

Sok weboldalon láthatunk elválasztó vonalakat, melyekkel tagolhatjuk szövegeinket, és nem utolsó sorban feldobhatjuk kinézetre oldalainkat.

A <HR> tag-et használjuk ebben az esetben. Mely vonaloknak a színét, szélességét, vastagságát, és rendezését is megadhatjuk a következő módon:

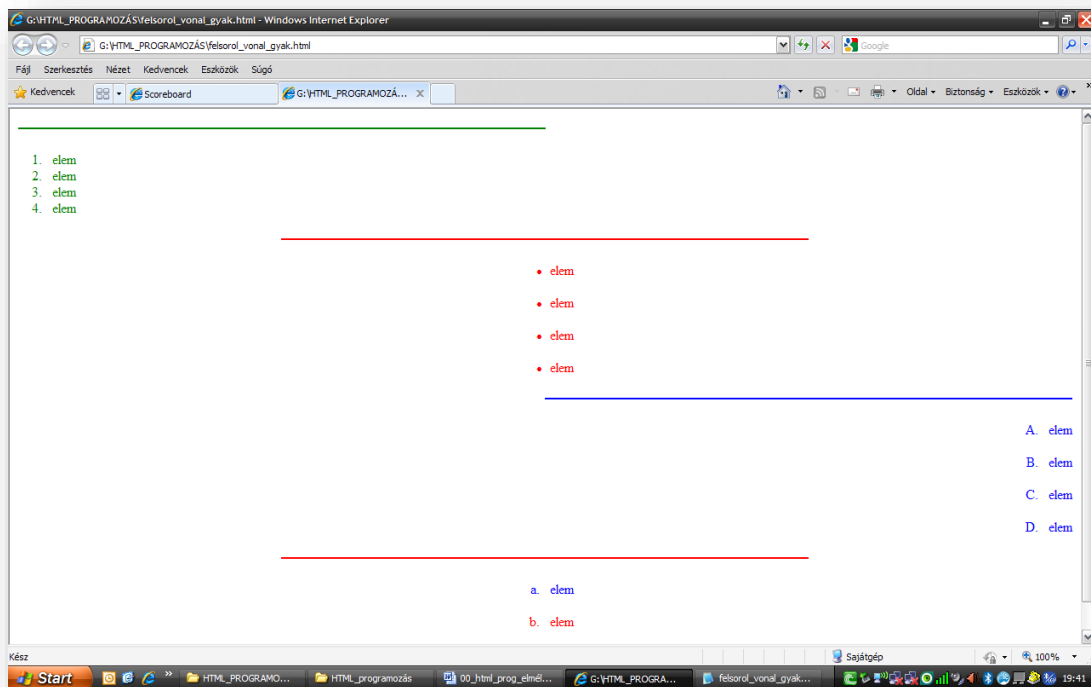
```
<hr color="green" width="50%" size=4 align="left">
```

Készítsd el a következő mintafeladatot zöld vonalakból! Majd mentsd **zoldvonal.html** néven!



Feladat13:

Készítsd el a képen látható weblapot. Az első vonal és felsorolás zöld; a második piros; a harmadik kék; a negyedik középre rendezett vonal piros és az alatta lévő kisbetűs felsorolásnál mindegyik sor más színű! A kész weblapot mentsd **13fels_szam.html** néven!



9. KERESZTHIVATKOZÁSOK HTML DOKUMENTUMOK KÖZÖTT

A HTML formátum lényegét az egymásra és egymás tartalmára való hivatkozások jelentik (vagyis a hypertext lehetőség). A dokumentum bármely részéhez hivatkozást (linket) helyezhetünk el, amelyet aktivizálva, a hivatkozottal összefüggésben lévő szöveghez jutunk el. A hivatkozó utasítások megjelenési formája sokféle lehet, a célobjektumtól függően:

A legegyszerűbb esetben a hivatkozás az adott fájl egy távolabbi részére mozdítja a böngészőablakot. A hivatkozás kezdetét a **** utasításnak a dokumentumban való elhelyezése jelzi. A hivatkozást a **** utasítás zárja le. Ez az elem pár közrezárhat szövegrészt, képet, stb. A közrezárt részt a böngészőprogram a dokumentum többi részétől eltérően jeleníti meg (pl. aláhúzással, kerettel, ...), az egérkurzorral fölé érve a mutató alakja megváltozik. Azt a részt (praktikusan: könyvjelzőt), ahová a hivatkozás mutat a **** és a **** utasítások kell, hogy határolják.

A legtöbb esetben a egy hivatkozás egy másik fájlra/dokumentumra mutat. A hivatkozás kezdetét ekkor a **** utasítás jelzi, a hivatkozást ekkor is a **** utasításelem zárja le. Mind a protokoll, mind az elérési út elhagyható, amennyiben azonos URL-en van a kiindulási dokumentum és a hivatkozott. A hivatkozott fájlnak e példában nincs külön névvel (könyvjelzővel) jelölt része. Működés szempontjából a fentebb leírtak vonatkoznak erre a hivatkozási formára is.

A legbonyolultabb esetben a hivatkozás egy másik fájl valamely pontosan meghatározott részére mutat. A hivatkozás kezdetét a **** utasítás jelzi, és a hivatkozást szintén a **** elem zárja le. Ebben az esetben a hivatkozott fájl kell, hogy tartalmazzon egy olyan részt (könyvjelzőt), ahová a hivatkozás mutat. Ezt a részt a **** és a **** utasítások határolják.

Megjegyzés: Ha az ****, **** utasításpár képet fog közre, akkor a kép szegéllyel jelenik meg, amely szegély letiltható az **** utasításban elhelyezett **BORDER=0** opció alkalmazásával. A képekkel kapcsolatos egyéb hivatkozási lehetőségeket lásd a képeknél.

Az alábbi példadokumentum többféle hivatkozást tartalmaz:

```
<HTML>
  Ennek a fájlnak a <A HREF="#vege">végére</A> visz ez a hivatkozás.
  <P>
  A Premonteri Gimnázium honlapjára vonatkozik e
  <A HREF="http://www.szent-norbert.sulinet.hu">hivatkozás</A>.
  <P>
  A fenti iskola újságjának egy adott
  <A HREF=" http://www.szent-norbert.sulinet.hu/kolmank/index.htm">
  helyére</A>
  repít e hivatkozás.
  Ezen leírással kapcsolatos véleményed
  <P>
  <br>
  <br>
  ///((((10 szer beszúrva –ezt nem kell beírni))))
  <A NAME="vege">Itt az oldal vége!</A>
  <P>
</HTML>
```

Gépelj be az utasításokat és mentsd **hivatkozások.html** néven! Próbáld ki a hivatkozásokat, működnek-e!

A több összetartozó weblapot egy mappában tároljuk. Ezt webhelynek nevezzük. A főoldalt ahonnan elindulunk, mindig *index.html*-nek kell nevezni. (Esetlegesen feltöltjük egy webtárhelyre, mindig az *index.html*-t indítja el) (vagy *index.php*)

Feladat14:

Hozzál létre, egy **14hivatkozások** nevű mappát a mappában készíts 5 db weblapot *index.html*, *1.html*; *2.html*; *3.html*; *4.html* néven! A minta alapján készítsd el az oldalakat!

Főoldal

1 | 2 | 3 | 4

Index.html

1. oldal

vissza a főoldalra | 2 | 3 | 4

1.html

2. oldal

1 | vissza a főoldalra | 3 | 4

2.html

3. oldal

1 | 2 | vissza a főoldalra | 4

3.html

4. oldal

1 | 2 | 3 | vissza a főoldalra

4.html

Készítsd el a hivatkozásokat úgy, hogy minden oldalról minden oldalra el lehessen jutni! Ha készen vagy mentse el minden oldalt, és ellenőrizd a hivatkozásokat!

Feladat15:

Készíts egy weboldalt, melyben könyvjelzőket, belső hivatkozásokat helyezel el, egymástól olyan távolságban, hogy látványosan ugráljon az oldalon. (nem baj, ha üres sorokkal növeled meg a távolságot) A weblap neve **15konyvjezo.html** legyen!

A lap teteje | [A lap ¼-de](#) | [A lap ½ -e](#) | [A lap ¾-e](#) | [A lap alja](#)

...

[A lap ¼-de](#) | [ugrás a lap tetejére](#)

[A lap ½ -e](#) | [ugrás a lap tetejére](#)

[A lap ¾-e](#) | [ugrás a lap tetejére](#)

[A lap alja –](#) | [ugrás a lap tetejére](#)

10. A HTML DOKUMENTUM SZÖVEGTESTÉNEK FELÉPÍTÉSE

Minden HTML formátumú szövegfájl a **<BODY>** és a **</BODY>** utasításokkal közrezárt részében tartalmazza a megjelenítendő részét. (A dokumentum-kereteket kivéve!)

Ezen elemek között kell elhelyezni mindent: a szöveget, hivatkozásokat, képeket, stb.

A **<BODY BACKGROUND="fájlnev.kit" BGCOLOR="színkód" TEXT="színkód" LINK="színkód" VLINK="színkód" ALINK="színkód">** utasításban a dokumentumtörzsre vonatkozó fenti előírások is szerepelhetnek opcióként.

A **BACKGROUND="elérési_út/fájlnev.kit"** opcióval a dokumentum háttéréül szolgáló fájlt jelölhetjük ki.

Háttérszínt a **BGCOLOR="színkód"** opcióval kiegészített utasítással definiálhatunk. (Amennyiben háttérmintául szolgáló fájlt - is megadunk, akkor a háttérszín csak nagyon ritkán fog előtűnni a dokumentumban, pl. a keretek szegélyében.)

A dokumentumban a szöveg színét a **TEXT="színkód"** opcióval jelölhetjük ki.

A **LINK="színkód"** opció a hivatkozások megjelenési színét határozza meg. A **VLINK="színkód"** pedig, a már bejárt hivatkozásokat jelölő színt határozza meg.

Természetesen egyszerre több opció is szerepelhet - tehát nem kötelező egyik sem - a **<BODY>** utasításban, mint azt az alábbi példa mutatja:

```
<HTML>
  <BODY BACKGROUND="zold.gif"
    BGCOLOR="#FF3333"
    TEXT="#000099"
    LINK="#993399"
    VLINK="#009999">
  <P>
  A háttérkép mozaikszerűen van elrendezve, és nem látszik a piros háttér.
  </BODY>
</HTML>
```

Arra kell figyelni, hogy a kép elérési útja helyes legyen. Célszerű egy mappába pakolni a felhasznált képeket és az oldalt. (ennél a feladatnál a nyersanyag a „hatter” nevű mappában van)

Feladat16:

Készíts a **16_feladat** mappában két weblapot **ho.html** és **szin.html** néven!

- A ho.html oldalon állítsd be háttérképnek a hopehely.jpg képet, és a lap tetejére, középre készíts linket UGRÁS felirattal!
- A link színe legyen világoszöld, a felkeresett link legyen sötétzöld A linkkel a szin.html oldalra ugorjon!
- A szin.html oldal háttérszíne legyen világoskék, jobb oldalon legyen egy link UGRÁS felirattal!
- A link színe legyen narancssárga, a felkeresett link színe legyen piros!
- Ezzel a linkkel a ho.html oldalra ugorjon vissza!

11. KÉPEK ELHELYEZÉSE A HTML DOKUMENTUMBAN

A HTML formátumú dokumentumban képeket - grafikákat - is is elhelyezhetünk. Az

`` utasítás a szöveg aktuális pozíciójába helyezi a megadott képet.

Ennél azért a legegyszerűbb szövegszerkesztő program is többet nyújt. A HTML dokumentum csinosítására is vannak a képek elhelyezésének finomabb lehetőségei is. Ha ezeket mind kihasználjuk, akkor az utasítás a következőképpen fog kinézni:

`<IMG ALIGN="hely" SRC="elérési_út/fájlnev.kit" HSPACE="vszám" VSPACE="fszám"
WIDTH="szméret" HEIGHT="mmeret" UNITS="egység" USEMAP="#jelző" ISMAP ALT="szöveg">`.

Az **ALIGN** opció meghatározza a kép igazításának módját, lehetséges értékei: *top, middle, bottom, left, right*.

A **HSPACE** a kép melletti vízszintes térközt, a **VSPACE** pedig a függőleges térközt (ha úgy tetszik: margókat) határozza meg.

A **WIDTH** a szélességét, a **HEIGHT** pedig a magasságát adja a képnek, az **UNITS** által meghatározott egységben (*pixel vagy en*).

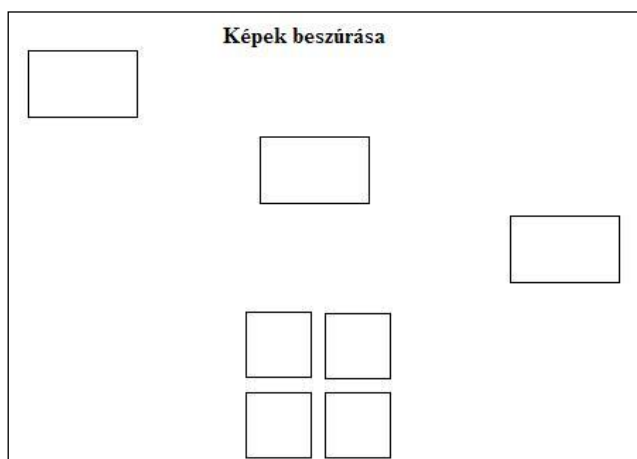
Az **ALT** azt a szöveget adja meg, amelyet nem grafikus böngészők használata esetén meg fog jelenni a kép helyett.

Hozzál létre egy mappát „KÉP” néven! A számítógépen található mintaképek közül másold be a mappába a kekhegyek.jpg-t! Majd hozzál létre egy **mintakep.html** nevű weblapot! Gépeled be a következő példát! Ments és nézd meg az eredményt!

```
<html>
<head>
<title>Kék hegyek</title>
</head>
<body bgcolor="gray">
<h1 align="center"><font color="Blue">Kékhegyek</font></h1>
<p align="center"></align>
</body>
</html>
```

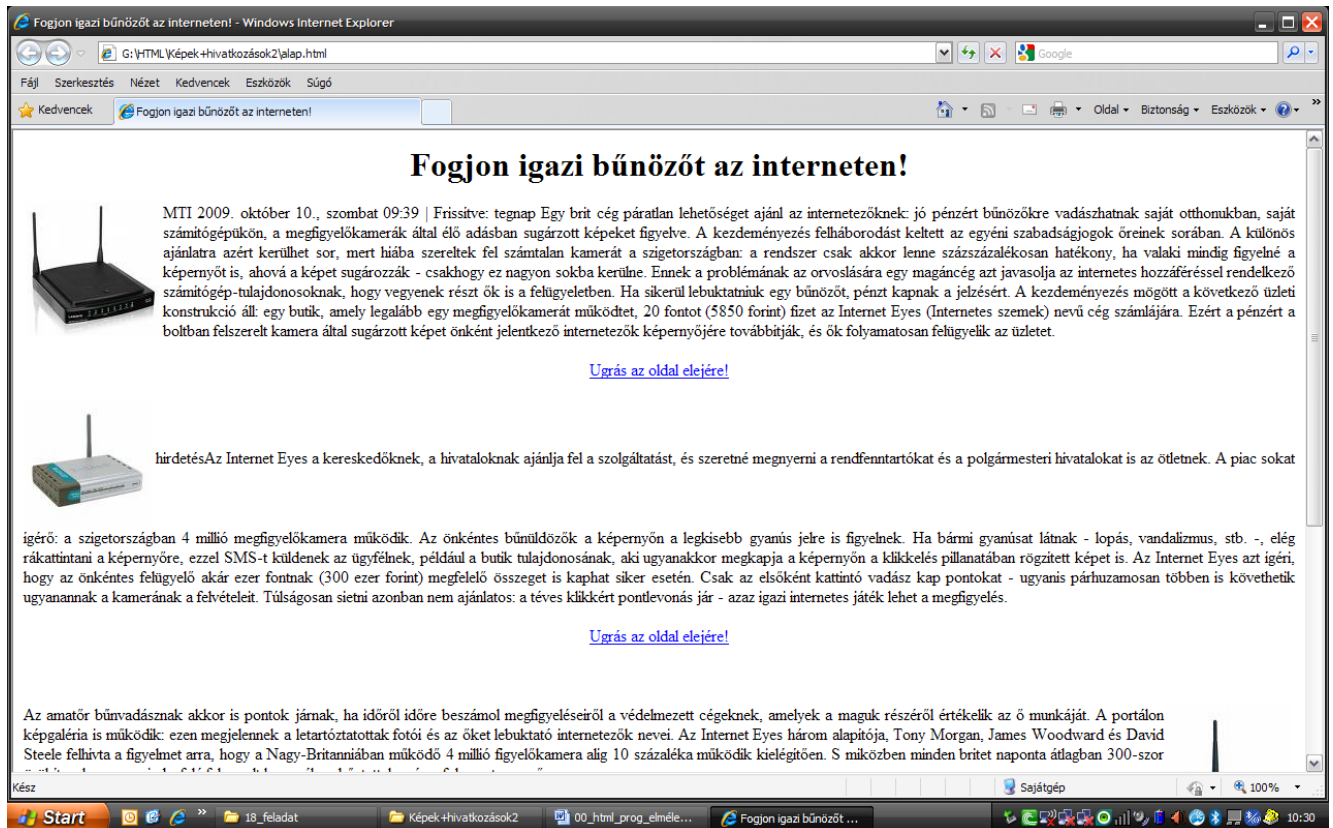
Feladat17:

Hozzál létre egy **17_feladat** nevű mappát! Másold bele mind a négy mintaképet! A képek felhasználásával készítsd el a következő weblapot **kepek.html** néven! Készíts egyes szintű címsort! Majd ballra, középre, és jobbra rendezve szúrjál be 200 képpontosra méretezett képeket (a négyből bármelyik hármat)! Aztán alá a négy kép felhasználásával egyenlő távolságra, 200*200 képpont nagyságban készíts montázst!



Feladat18:

Készíts weblapot a mintának megfelelően, a **18_feladat** mappában található nyersanyagok felhasználásával! Használj oldalon belüli hivatkozásokat! A weblap neve legyen **kepek_rend.html**!



Feladat19:

Készíts képgalériát a 19_feladat mappában található képekből! Minden oldalon szerepeljen mind a hat kép kicsiben ugyan abban a pozícióban, alatta az aktuális kép kinagyítva! Működjön a képre rákattintva, és az előző, következőre kattintva is! Ügyelj arra, hogy a kis képek arányosan legyenek elhelyezve, azonos méretben. A nagy kép ugyanolyan széles legyen, mint a felső hat kép szélessége! Összesen 6 weboldalt kell létrehoznod, az utolsó oldal után ugorjon az elsőre, tehát egy körforgást hozzá! Az oldalak elnevezését rád bízom!



12. A HTML FORMÁTUM TÁBLÁZATAI

A HTML formátumnak ez az utasításcsoportja képes a legváltozatosabb szöveg-, és képmegjelenítési formák előállítására. A **<TABLE>** és a **</TABLE>** utasítások közé zárt részt tekintjük egy táblázatnak.

A táblázatnak a címét a **<CAPTION>** és a **</CAPTION>** utasítások között kell megadni. (Figyelem! Az így megadott cím nem a táblázatban, hanem előtte fog megjelenni!) A cím **<CAPTION ALIGN="hely">** formájú megadással igazítható.

A táblázat minden sora a **<TR>** utasítással kezdődik és a következő **<TR>**-ig, ill. a táblázat végéig tart. Egy sor tartalmazhat oszlopfejléceket és adatokat. Az oszlopfejléceket a **<TH>** utasítás vezeti be és választja el egymástól. A táblázat adatcellái pedig a **<TR>**-rel megkezdett sorban egy **<TD>** utasítással kezdődnek és minden cella a következő **<TD>**-ig - ill. a következő sor elejét jelző elemig - tart, ahol értelemszerűen új cella kezdődik. Az oszlopfejléceknek és az adatcelláknak csak a kezdőutasítása használatos - habár van lezáró utasításuk is (**</TH>**, **</TD>**) -, mert a záróutasításuk elhagyható!

A táblázat nyitóutasítása tartalmazhat a teljes táblázatra vonatkozó beállításokat:

<TABLE BORDER="szám" ALIGN="hely" COLSPEC="oszlopjellemezők" UNITS="egység" NOWRAP CELLPADDING="pszám" CELSPACING="kszám" BGCOLOR="színkód">

Ahol a **BORDER** opció a rácsozat szélességét határozza meg. (0 esetén nincs rácsozat.) Az **ALIGN** a teljes tábla elhelyezkedését határozza meg (*left, right, center* lehet). A **COLSPEC** egy oszlop igazítását és szélességét adja meg. Egy oszlopra vonatkozóan egy betű és szám egybeírva (pl.: *L12 C24 R10*), melytől a következő oszlop értékeit egy köz választja el. Az **UNITS** a számokhoz tartozó mértékegységet jelöli ki (*en, relative* - oszlopszélességhez -, *pixel*). A **NOWRAP** opció a cellák szövegének tördelését tiltja le. Végül a **BGCOLOR** a táblázat háttérszínét határozza meg.

A következő példában egy 3 soros 5 oszlopos 500 képpont széles táblázatot helyezünk az oldal közepére!

```
<html>
<head>
<title>tablazat1</title>
</head>
<body>
<div align="center">
<table border="1" width="500" id="table1">
<tr>
<td align="center">11</td>
<td align="center">12</td>
<td align="center">13</td>
<td align="center">14</td>
<td align="center">15</td>
</tr>
<tr>
<td align="center">21</td>
<td align="center">22</td>
<td align="center">23</td>
<td align="center">24</td>
<td align="center">25</td>
</tr>
<tr>
<td align="center">31</td>
<td align="center">32</td>
<td align="center">33</td>
<td align="center">34</td>
<td align="center">34</td>
</tr>
</table>
</div>
</body>
</html>
```

A táblázat oszlopfejlécei nem csak a legfelső oszlopban szerepelhetnek, hanem a táblázatban bárhol (pl. sorok címeként is).

Mind az oszlopfejlécekben, mind az adatcellákban használhatók a következő formázásra való opciók:

- **COLSPAN="szám"**: Egyesít több egymással szomszédos cellát - vízszintesen.
- **ROWSPAN="szám"**: Egyesít több egymás alatti cellát - függőlegesen.
- **ALIGN="hely"**: Igazítja a cellák tartalmát - vízszintesen. Lehetséges értékei: *left, center, right, justify, decimal*
- **VALIGN="hely"**: Igazítja a cellák tartalmát - függőlegesen. Lehetséges értékei: *top, middle, bottom, baseline*

A következő példában látunk egyesített sorokat és oszlopokat!

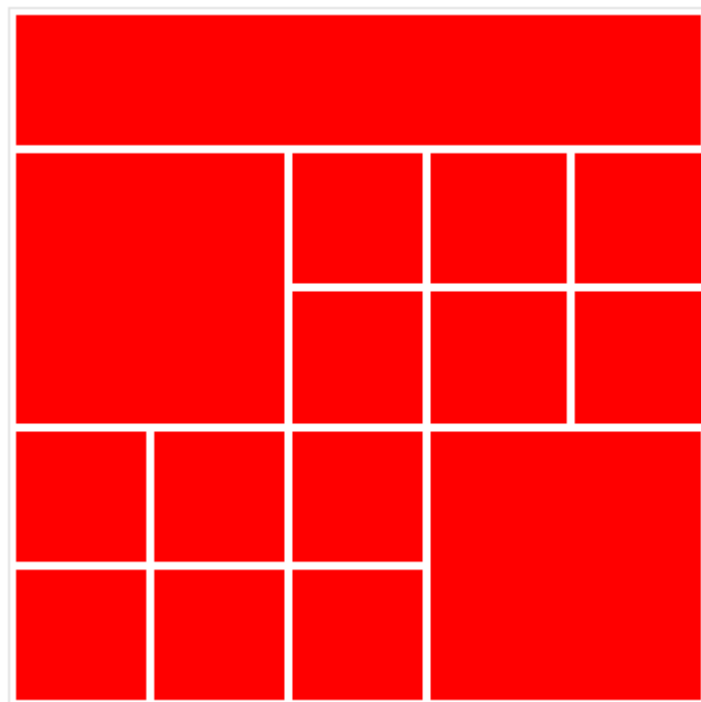
```
<html>
<head>
<title>tablazat2</title>
</head>
<body>
<div align="center">
  <table border="1" width="600" id="table1">
    <tr>
      <td colspan="6" bgcolor="#FF0000">
        <p align="center">11-16</p>
      </td>
    </tr>
    <tr>
      <td align="center">21</td>
      <td align="center">22</td>
      <td align="center">23</td>
      <td align="center">24</td>
      <td align="center">25</td>
      <td align="center">26</td>
    </tr>
    <tr>
      <td align="center">31</td>
      <td align="center">32</td>
      <td colspan="2" rowspan="2" bgcolor="#000080"
        align="center">33-44</td>
      <td align="center">35</td>
      <td align="center">36</td>
    </tr>
    <tr>
      <td align="center">41</td>
      <td align="center">42</td>
      <td align="center">45</td>
      <td align="center">46</td>
    </tr>
    <tr>
      <td align="center">51</td>
      <td align="center">52</td>
      <td align="center">53</td>
      <td align="center">54</td>
      <td align="center">55</td>
      <td align="center">56</td>
    </tr>
  </table>
</div>
</body>
</html>
```

Az előző példa begépelése után a következőt látjuk.

11-16					
21	22	23	24	25	26
31	32	33-44		35	36
41	42			45	46
51	52	53	54	55	56

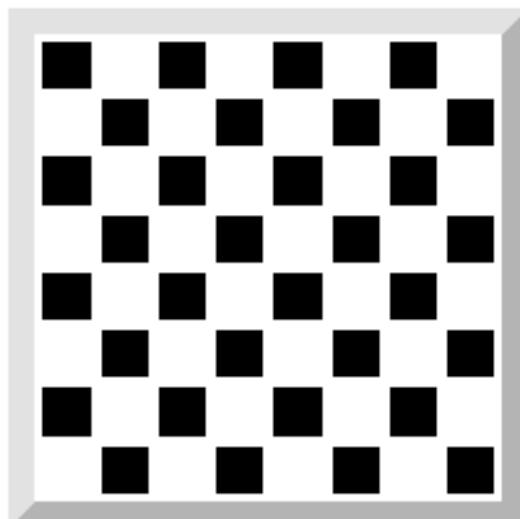
Feladat20:

Készítsd el a következő táblázatot, piros kitöltéssel! Az oldalt **20tablazat.html** néven mentsd!



Feladat21:

Készíts vastag szegélyű sakktáblát a minta alapján, **21sakk.html** néven!



Feladat22:

Készítsd el a következő táblázatot, **22tablazat.html** néven! A felső sorban a színek rendre, piros, zöld, sárga, kék. A méretek nincsenek megadva az a lényeg, hogy a lehető legjobban hasonlítson a mintához.

<ul style="list-style-type: none">• első• második• harmadik• negyedik		<u>1.</u>		<u>2.</u>		<u>3.</u>			

13. A HTML DOKUMENTUMABLAK FELOSZTÁSA (FRAME-EK)

Egyetlen böngészőablakban több HTML dokumentum is megjeleníthető a **<FRAMESET>** és a **</FRAMESET>** utasításpár, valamint a szorosan kapcsolódó **<FRAME>** utasítás együttes használatával.

Manapság már nem használják a frame-eket a weblapkészítők. Inkább táblázatokban dolgoznak.

A **<FRAMESET ROWS="oszlophatárok">** kezdőutasítással osztható fel a képernyő függőlegesen, a **<FRAMESET COLS="sorhatárok">** utasítással pedig vízszintesen. Ahol az oszlop- és sorhatárok megadhatók képernyőpontban ill. százalékosan - vesszővel elválasztva -, a maradék képernyőterületre pedig a * dőzserkarakter használatával lehet hivatkozni. Mivel vagy csak vízszintesen, vagy csak függőlegesen osztható fel a képernyő, ezért ha mindkét irányban osztott böngészőablak létrehozásához a **<FRAMESET>** elemeket egymásba kell ágyazni! Tehát például egy függőleges felosztáson belül kell vízszintesen elválasztott részekre tagolni egy oszlopot.

A fenti módon definiált területekre a **<FRAME SRC="objektum">** utasítás tölti be a megadott objektumot, mely objektum lehet egy teljes HTML fájl, annak egy meghatározott része, ill. egy kép. Az így kitöltendő keretek viselkedését szabályozza az utasítás **<FRAME NAME="név" SRC="objektum" SCROLLING="érték" MARGINWIDTH="szám" MARGINHEIGHT="szám">** alakú megadása.

Az adott keretnek nevet ad a **NAME** opció, a szkrollozást letilthatja **SCROLLING="no"** kiegészítés (ezenkívül a *yes* és az *auto* értékeket veheti fel a **SCROLLING** opció), a **MARGINWIDTH** és a **MARGINHEIGHT** pedig a kereten belüli margók szélességét szabályozza.

Például a fejlécben megadott **<BASE TARGET="név">** utasítás a **NAME="név"** opcióval elnevezett keretbe irányítja a hivatkozásokat. Egyébként az **** utasítás is ismeri a **TARGET="név"** opciót. (A **TARGET="_top"** opcióval az egész böngészőablakot elfoglalja a hivatkozott dokumentum, tehát feloldja az ablak keretekre osztását!) Ha ezek egyike sem szerepel, akkor a hivatkozás a hivatkozó objektum keretében jelenik meg, felülírva azt!

A **<FRAMESET>**, **<FRAMESET>** utasításpárral határolt területnek meg kell előznie a **<BODY>** utasítással kijelölt dokumentumtörzset! Sőt a egy **<NOFRAMES>** utasítással kell jelezni a dokumentum azon részének kezdetét, amelyet akkor kell a böngészőnek megjelenítenie csak, ha nem ismeri a keretutasításokat. És csak ez a **<NOFRAMES>**-mel kezdődő rész tartalmazhatja a **<BODY>** és a **</BODY>** utasításpárt.

Az alábbi példa a kerethasználat több lehetőségét is bemutatja:

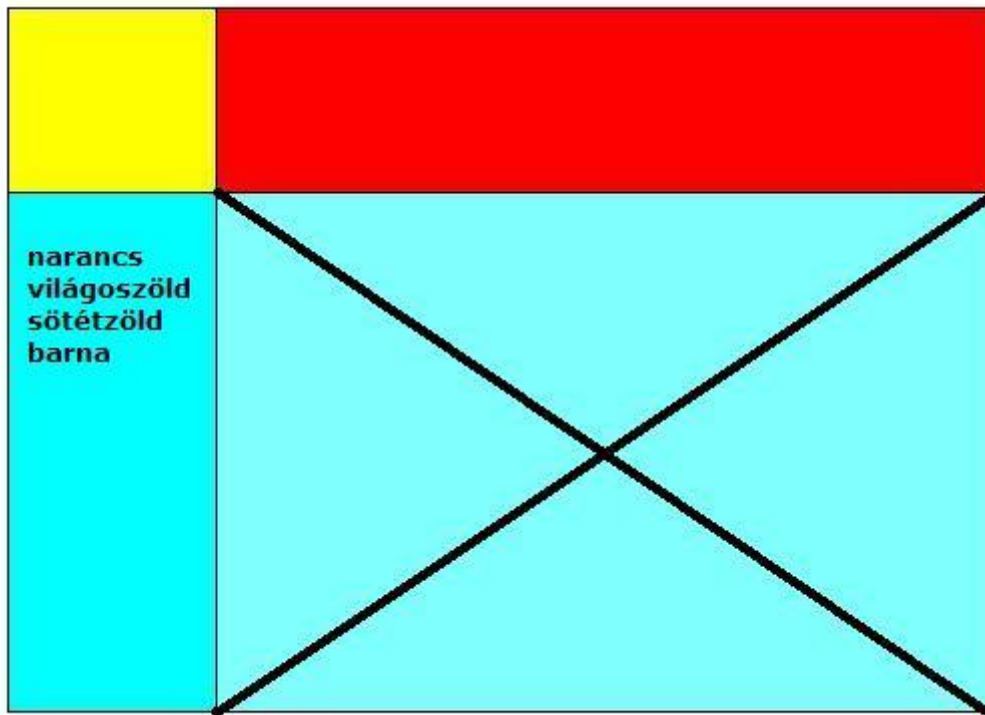
```
<HTML>
  <FRAMESET ROWS=185,*>
  <FRAMESET COLS=185,*>
  <FRAME SRC=k08.gif SCROLLING=NO NAME="cimer">
  <FRAME SRC=02.htm NAME="felepites">
  </FRAMESET>
  <FRAMESET COLS=25%,*>
  <FRAME SRC=index.htm NAME="tart">
  <FRAME SRC=13.htm NAME="keret" MARGINHEIGHT=10 MARGINWIDTH=10>
  </FRAMESET>
  </FRAMESET>

  <NOFRAMES>
  <CENTER>
  <BODY BGCOLOR="#FFFF00">
  <FONT COLOR="#FF3333">
  <H1>Sajnos ez a böngésző nem támogatja a keretek használatát!</H1>
  </FONT>
  </BODY>
</HTML>
```

Feladat23:

Hozzál létre egy mappát **23_feladat** néven! Aztán a mappában 8 weblapot hozzál létre, **sarga.html**; **vilzold.html**; **sotetzold.html**; **piros.html**; **vilkek.html**; **sotettek.html**; **narancs.html**; **barna.html** néven! Ezek az oldalak üresek legyenek!

Készíts frame-eket a minta alapján. A bal felsőben a citromsárga oldalra legyen hivatkozás. A jobb felsőben a piros oldalra legyen hivatkozás! A bal alsóban legyen a sötétkék oldal, viszont ezen az oldalon legyen négy hivatkozás, a maradék 4 oldalra, melyek a negyedik frame-ben (az X-el jelölt) jelenjenek meg, ha rákattintanak!



14. KÉRDŐÍVEK A HTML DOKUMENTUMBAN

A HTML formátumú dokumentumban kérdőíveket is közzétehetünk, melyek feldolgozásához külön programot kell írni. (Nem HTML-alapút! Általában a kiszolgáló */cgi-bin/* könyvtárában található programról van szó.)

A **<FORM METHOD="mód" ACTION="elérési_út/fájlnév.kit">** és a **</FORM>** utasítások zárják közre a kitöltendő kérdőívet/űrlapot.

Az opciókat ajánlott mindig megadni, már csak azért is, mert az **ACTION** határozza meg a feldolgozást végző programot, a **METHOD** pedig a kitöltött űrlap továbbítási módját a feldolgozó programnak. Lehetséges értékei: *GET* - az URL-ben, *POST* - adatcsomagban. Az alapértelmezés a *GET*, ami bizonyos veszélyeket rejt magában, mivel túl hosszúra nyúlhat az URL. A *POST* a biztonságosabb mód.

Az **<INPUT NAME="név" TYPE="típus" ALIGN="hely">** utasítással határozható meg egy kitöltendő űrlapmező.

A **NAME** természetesen a mezőnév, amely alapján a feldolgozóprogram azonosítja a bevitt adatot. A **TYPE** pedig az adattípus, melyet vár a beviteli mező. Lehetséges típusok: *TEXT* - szöveg, *PASSWORD* - jelszó (nem jelenik meg bevitelkor!), *HIDDEN* - rejtett (ez sem jelenik meg), *CHECKBOX* - kapcsoló (több is kiválasztható egyszerre), *RADIO* - kapcsoló (egyszerre csak egyet lehet kiválasztani), *RANGE* - numerikus adat, *FILE* - csatolandó fájl, *SUBMIT* - adattovábbító gomb, *RESET* - inicializáló gomb, *BUTTON* - egyéb nyomógomb. Az **<INPUT >** utasításban további opciók is szerepelhetnek, a fő opciók értékeitől függően:

- A **VALUE** kiegészítő opcióval megadott értéket veszi fel alapértelmezésként a szöveges vagy numerikus beviteli mező.

- *TEXT* típusú mező esetén egy további opció, a **SIZE="méret"** opció határozza meg a beviteli ablak szélességét, a **MAXLENGTH="érték"** pedig a bevihető maximális szöveghosszt.

- A *CHECKBOX* és a *RADIO* típusú mezők további paramétere lehet a **CHECKED** opció, mely bekapcsolja a kapcsolót - alapértelmezésként.

- *RANGE* típusú mező esetén megadható az a tartomány, melybe a bevitt értéknek bele kell esnie, a **MAX="maximum"** és a **MIN="minimum"** további opciókkal.

- A *FILE* típusú mezőben megadott fájl az **ACCEPT** kiegészítő opcióval megadott MIME módon csatolódik az elküldendő kérdőívhez. (Megjegyzés: Egy *Browse* nyomógommbal támogatott fájlkereső-ablakból lehet a fájlt kiválasztani.)

- A *SUBMIT* és a *RESET* gombokhoz tartozó kiegészítő opció a **VALUE="felirat"**, amely a gombok feliratát jelöli ki. Egyébként a *SUBMIT* gomb lenyomásának hatására küldi el az űrlapadatokat a kérdőív a feldolgozó programnak, a *RESET* gomb lenyomása pedig az alapértékekkel tölti fel a beviteli mezőket.

Hosszab szöveg bevitelére alkalmas a

<TEXTAREA NAME="név" ROWS="magaság" COLS="szélesség" VALUE="szöveg">, **</TEXTAREA>** utasításpár, amely egy beviteli ablakot nyit a **COLS**-ban megadott szélességben és a **ROWS**-ban megadott sorban. A **VALUE** az alapértelmezésként megjelenítendő szöveget adja meg.

Egy kérdésre adandó válasz egyszerű - menüből történő - kiválasztását teszi lehetővé a kérdőíven a **<SELECT NAME="név" SIZE="sor">**, **</SELECT>** utasításokkal létrehozott kiválasztásos menü, melynek menüpontjait az **<OPTION>** utasítással adhatjuk meg.

A **SIZE** opció azt határozza meg, hogy hány sorban jelenjenek meg a választható menüpontok. Megadásával szkrollozható menüt kapunk. Elhagyása esetén, ún. legördülő menüből lehet választani. A **MULTIPLE** opció esetén több menüpont is kijelölhető egyszerre. Az **<OPTION SELECTED>** formájú utasítás adja meg az alapértelmezett választást!

```
<HTML>
  <CENTER>
  <H1>Adatfelvételi lap:</H1>
  <FORM METHOD="post" ACTION="program.bin">
  <INPUT TYPE="reset" VALUE="Alapértelmezés">
  <P>
  Vezetéckneve:
  <INPUT NAME="vezeteknev" TYPE="text" VALUE="Kovács" SIZE="25"
    MAXLENGTH="30">

  Keresztneve:
  <INPUT NAME="keresztnev" TYPE="text" MAXLENGTH="50">
  Férfi: <INPUT NAME="neme" TYPE="radio" CHECKED>
  Nő: <INPUT NAME="neme" TYPE="radio">
  Kora: <INPUT NAME="kor" TYPE="range" SIZE="2" MIN="10" MAX="60">
  <P>
  Érdeklődési köre:
  Windows: <INPUT NAME="erdek" TYPE="checkbox" CHECKED>
  Win95: <INPUT NAME="erdek" TYPE="checkbox">
  LINUX: <INPUT NAME="erdek" TYPE="checkbox">
  OS/2: <INPUT NAME="erdek" TYPE="checkbox">
  <P>
  <TEXTAREA NAME="egyeb" COLS="40" ROWS="4">Közlendők:</TEXTAREA>
  <P>
  Foglalkozása:
  <SELECT NAME="foglakozas">
  <OPTION>diák
  <OPTION>tanár
  <OPTION SELECTED>nyugdíjas
  <OPTION>egyik sem
  </SELECT>
  <P>
  Csatolandó fájl(ok):
  <BR>
  <INPUT NAME="fajl" TYPE="file">
  <P>
  <INPUT TYPE="submit" VALUE="Elküldés">
  </CENTER>
  </FORM>
</HTML>
```

Az előző példa a következő képen néz ki:

Adatfelvételi lap:

Alapértelmezés

Vezetéckneve: Kovács Keresztneve: Férfi: Nő: Kora:

Érdeklődési köre: Windows: Win95: LINUX: OS/2:

Közlendők:

Foglalkozása: nyugdíjas

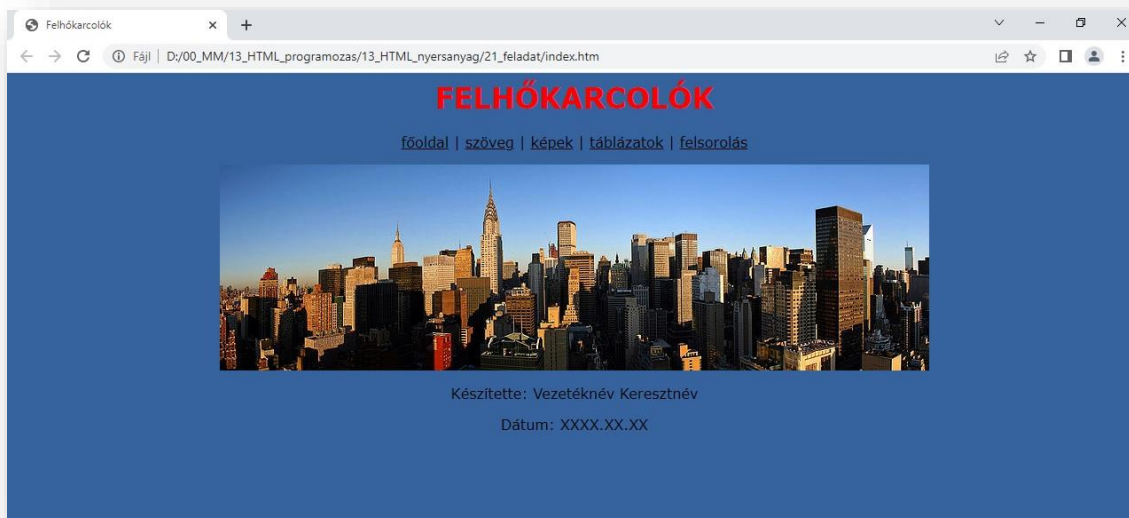
Csatolandó fájl(ok): Tallózás...

Elküldés

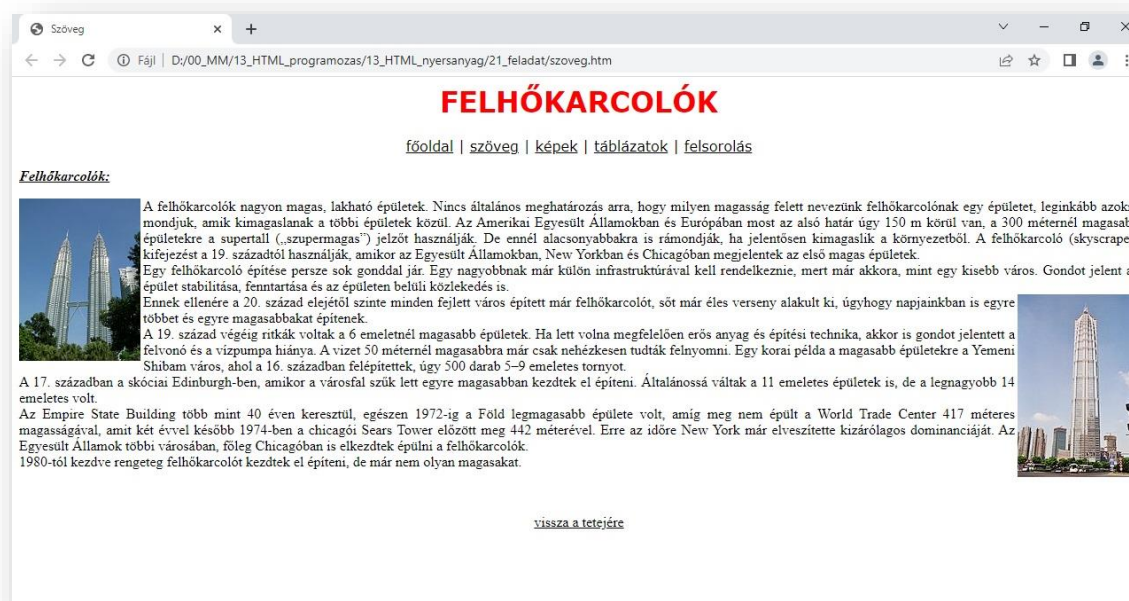
15. ÖSSZEFOGLALÓ PÉLDA

Készítsd el a következő 5 lapból álló webhelyet! Minden szükséges nyersanyagot a **24_feladat** nevű mappában találsz! A képeken látható **minta alapján készítsd el a feladatot!** A részfeladatok elkészítése közben mindig mentsél, és ellenőrizd a munkádat!

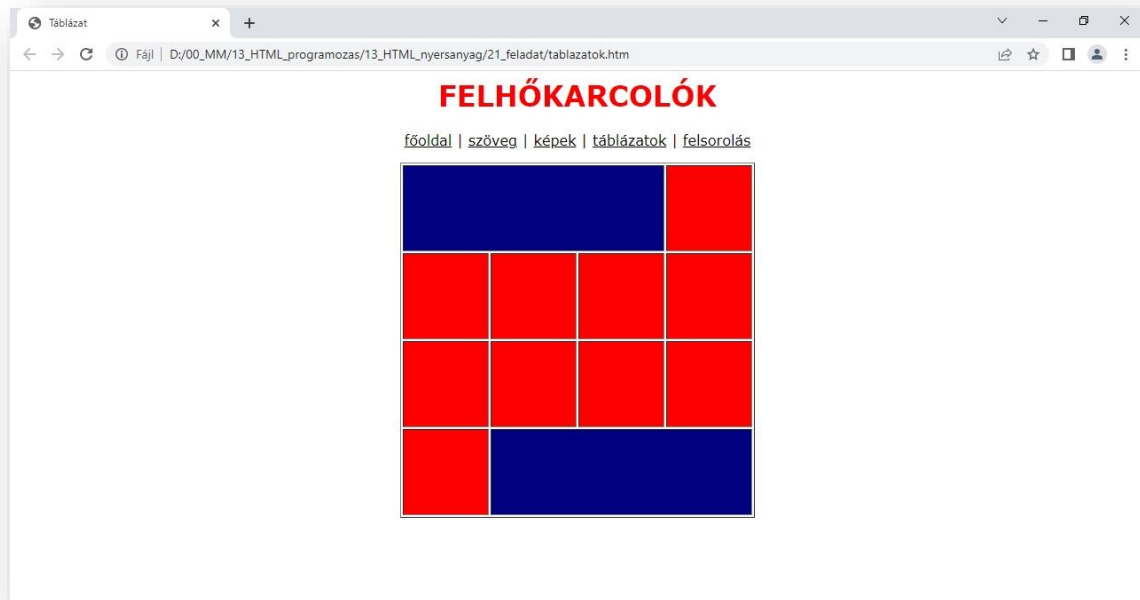
- A főoldalt index.html-nek nevezd el!
- Minden többi oldal nevét a mintáról olvasd le!
- A főoldal háttérképe legyen kék, a többi fehér
- Minden oldalon az egyes szintű címsor színe legyen piros és legyen középre rendezve!
- A fejléc részén a betűtípus legyen Arial Black! Minden többi betűtípust a képről azonosítsd! (Arial, Times New Roman)
- A címsor alatt készíts menürendszert, amit használj fel a többi oldalon is!
- A linkek minden állása legyen fekete!
- A böngésző címsorában a minta szerinti szöveg szerepeljen!



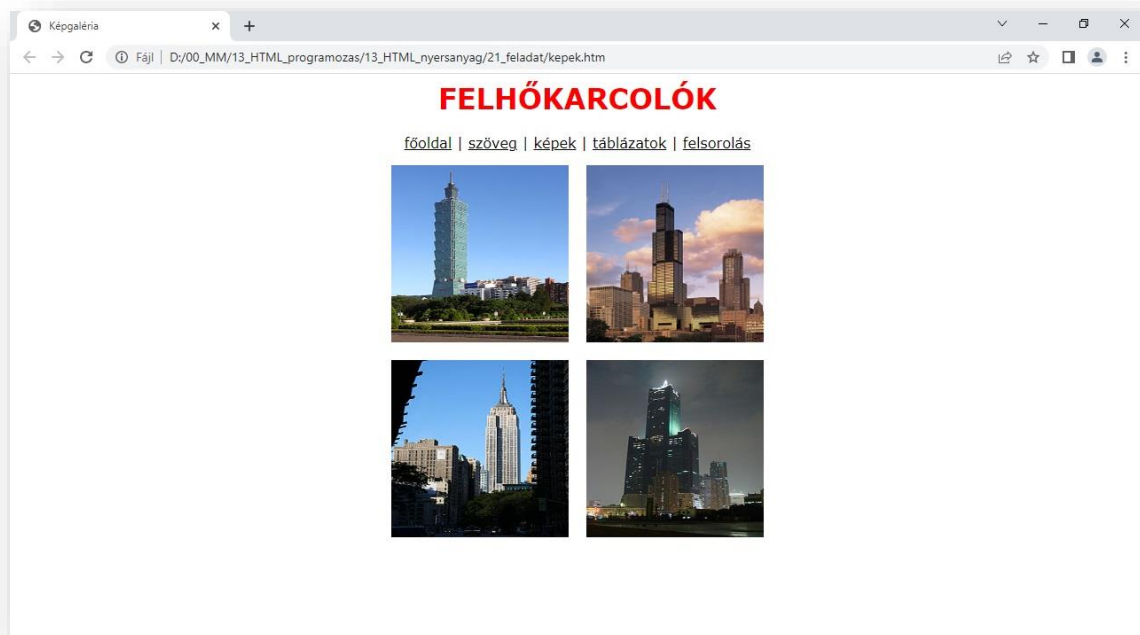
- Az alcím legyen félkövér és dőlt!
- A z összefüggő szöveg legyen sorkizárt!
- A képek legyenek körbefuttatva szöveggel! A méretük legyen tetszőleges, de körülbelül a mintának megfelelőek!
- A lap aljára készíts oldalon belüli hivatkozást, amellyel a lap tetejére ugorhatsz!



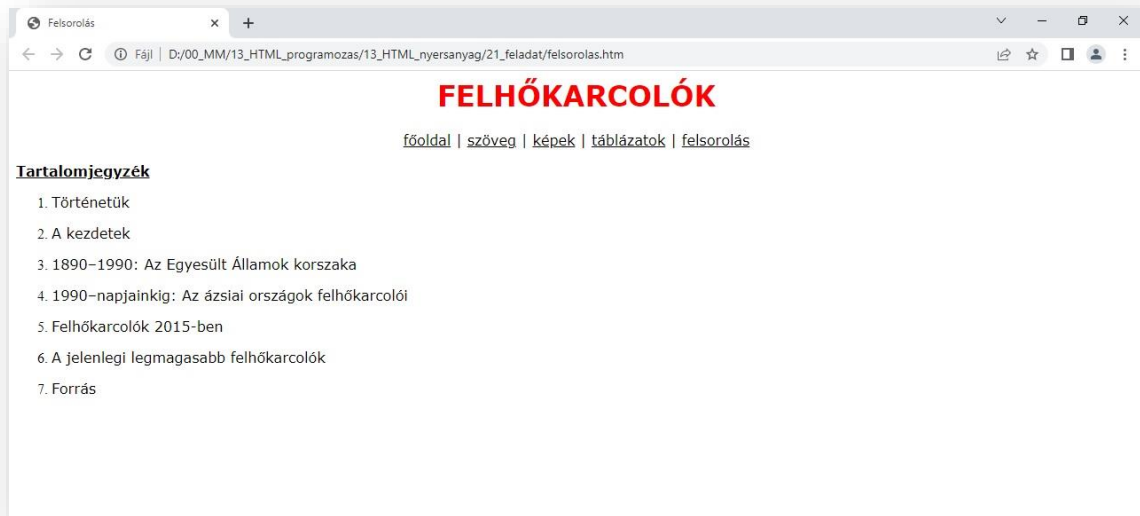
- A táblázat legyen középre rendezve!
- A táblázat mérete felejen meg a mintának!
- A 4*4-es táblázat felső és alsó sorában legyen a 3-3 cella összevonva!
- A kisebb cellák kitöltőszíne legyen piros, az egyesítetteké pedig legyen kék színű!



- A következő oldalon egy kis képgalériát készíts!
- A négy kép a minta szerint helyezkedjen el, és a képek a minta szerinti helyen legyenek!
- Az oldalak magassága és szélessége ugyanakkora legyen!
- Próbáld meg elérni azt is, hogy a képek között egyforma távolság legyen vízszintesen és függőlegesen is!



- Az alcím legyen félkövér és aláhúzott!
- A többi bekezdésre tegyél számozást a minta alapján!



- Ha elkészültél a feladattal, akkor ellenőrizd a munkádat és javítsd az esetleges hibákat!

1. Mi is az a CSS?

A CSS az angol „cascading style sheets” kifejezés rövidítése, ami magyarul „egymásba ágyazott stíluslapokat” jelent. A hangsúly a „stíluson” van – míg a HTML a weblap szerkezetét határozza meg (főcímek, bekezdések, stb.), és lehetővé teszi, hogy különböző elemeket (képek, videók) ágyazz webes dokumentumodba, addig a CSS a weblap vizuális stílusáért felel – az oldal elrendezéséért, a színekért, a betűkészletekért, azok méretéért, és így tovább. Tehát CSS-el formázhatjuk meg a weboldalunk megjelenését.



A webes böngészők megvizsgálják a dokumentum (weblap) CSS kódját, és ez alapján jelenítik meg a HTML elemeket.

Mire lehet használni a CSS-t?

- háttérkép beállítása
- háttérszín beállítása
- szövegméret beállítása
- szövegszín beállítása
- betűtípus meghatározása
- képek elhelyezkedésének meghatározása
- szövegigazítás változtatása
- tartalmak keretezése
- margó beállítása
- stb

Tehát, összefoglalva: A CSS (Cascading Style Sheets) egy stílusleíró nyelv, amit arra használunk, hogy formázzuk és dizájnoljuk a weboldalakat. Segítségével elkülöníthetjük az oldal tartalmát és a megjelenést.

2. CSS Szintaxis, alapstruktúra

```
választó {  
    tulajdonság: érték;  
}
```

Választó: A HTML elemet jelöli ki, amelyre a stílust alkalmazni szeretnénk.

Tulajdonság: Az a jellemző, amelyet módosítani szeretnénk.

Érték: A tulajdonság értéke.

Példa:

```
body {  
    font-family: 'Arial', sans-serif;  
    color: #333;  
}
```

3. Választók és Tulajdonságok

3.1. alapvető választók

- Elemválasztó: elem vagy #id
- Osztályválasztó: .osztaly
- Elemek kombinálása: elem1 elem2

3.2. Néhány Alapvető Tulajdonság

- Szöveg formázás: font-family, font-size, color

- Margók és padding: margin, padding
- Háttér: background-color, background-image

4. Box Modell

A dobozmodell segít megérteni, hogyan vannak elrendezve az elemek a weboldalon. Minden HTML elem egy doboznak tekinthető, és négy fő részből áll: margin, border, padding, és maga a tartalom.

5. CSS Időrend, alapértelmezett Stílusok Felülírása

CSS a kaskád elve alapján működik, ami azt jelenti, hogy a később beírt szabályok felülírják a korábbiakat.

Példa:

```
p {
  color: blue; /* Alapértelmezett szín */
}

.special {
  color: red; /* Felülírja az előző színt csak azokra az elemekre, amiknek van 'special' osztályuk */
}
```

17. PÉLDA EGY EGYSZERŰ CSS KÓDRA

Feladat 25:

- A feladat elején hozzunk létre egy mappát F25 néven!
- Ebben a mappában a végén négy fájlt fogunk tárolni!
- Először hozzunk létre egy pelda25.txt-t, melybe begépeljük a lenti kódot! A gépelés végeztével mentjük a változásokat és mentjük el másként a txt állományunkat pelda25.html néven!
- Aztán hozzunk létre egy style.txt fájlt melybe begépeljük a css kódját! Majd mentjük a változásokat és mentjük el másként style.css néven is a kódunkat!
- Majd a mappában nyissuk meg a böngészőnkkel a pelda25.html fájlnak!

HTML kód:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Példa</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <h1>Üdvözöllek a weboldalamon!</h1>
  </header>
  <section class="content">
    <p>Ez egy egyszerű példa arra, hogyan használhatod a CSS-t.</p>
    <p class="highlight">Ez a szöveg kiemelt stílusban van formázva.</p>
  </section>
</body>
</html>
```

CSS kód (style.css):

```
/* Alapértelmezett stílusok */
body {
  font-family: 'Arial', sans-serif;
  background-color: #f4f4f4;
  color: #333;
  margin: 0;
  padding: 0;
}

header {
  background-color: #007bff;
  color: #fff;
  text-align: center;
  padding: 20px;
}

/* Tartalom terület formázása */
.content {
  margin: 20px;
}

/* Kiemelt szöveg stílusa */
.highlight {
  font-weight: bold;
  color: #e44d26;
}
```

Ebben a példában a CSS kód formázza a weboldal háttérszínét, a szövegszíneket, a fejléc háttérszínét, és hangsúlyozza a kiemelt szöveget. Az osztályokat és az elemeket a CSS választókkal céloztuk meg, és különböző tulajdonságokat alkalmaztunk rájuk. A stílusok a style.css fájlban találhatóak, amit a HTML fájlban a <link> elem segítségével hivatkoztunk.

18. MÉG EGY PÉLDA A CSS KÓDOLÁSRA

Most itt egy hosszabb CSS kód példa, amely bemutatja különböző CSS tulajdonságokat és lehetőségeket. A kód egy egyszerű weboldalra vonatkozik, és tartalmaz példákat a dobozmodellre, elrendezésekre, animációkra és árnyékokra.

Feladat 26:

- A feladat elején hozzunk létre egy mappát F26 néven!
- Ebben a mappában a végén négy fájlt fogunk tárolni!
- Először hozzunk létre egy pelda25.txt-t, melybe begépeljük a lenti kódot! A gépelés végétével mentjük a változásokat és mentjük el másként a txt állományunkat pelda26.html néven!
- Aztán hozzunk létre egy style.txt fájlt melybe begépeljük a css kódját! Majd mentjük a változásokat és mentjük el másként style.css néven is a kódunkat!
- Majd a mappában nyissuk meg a böngészőnkkel a pelda25.html fájlunkat!

HTML kód:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Példa</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <h1>Weboldal Példa</h1>
  </header>
  <section class="content">
    <article>
      <h2>Cikk 1</h2>
      <p>Ez egy rövid cikk a CSS lehetőségeiről.</p>
    </article>
    <article>
      <h2>Cikk 2</h2>
      <p>Ez egy másik cikk, amely további példákat mutat be.</p>
    </article>
  </section>
</body>
</html>
```

CSS kód (style.css):

```
/* Alapértelmezett stílusok */
body {
  font-family: 'Arial', sans-serif;
  background-color: #f8f8f8;
  color: #333;
  margin: 0;
  padding: 0;
}

header {
  background-color: #333;
  color: #fff;
  text-align: center;
  padding: 20px;
}

/* Tartalom terület formázása */
.content {
  display: flex;
  justify-content: space-around;
  margin: 20px;
}

article {
  background-color: #fff;
  border: 1px solid #ddd;
```

Témakör: A HTML/CSS programozás alapjai

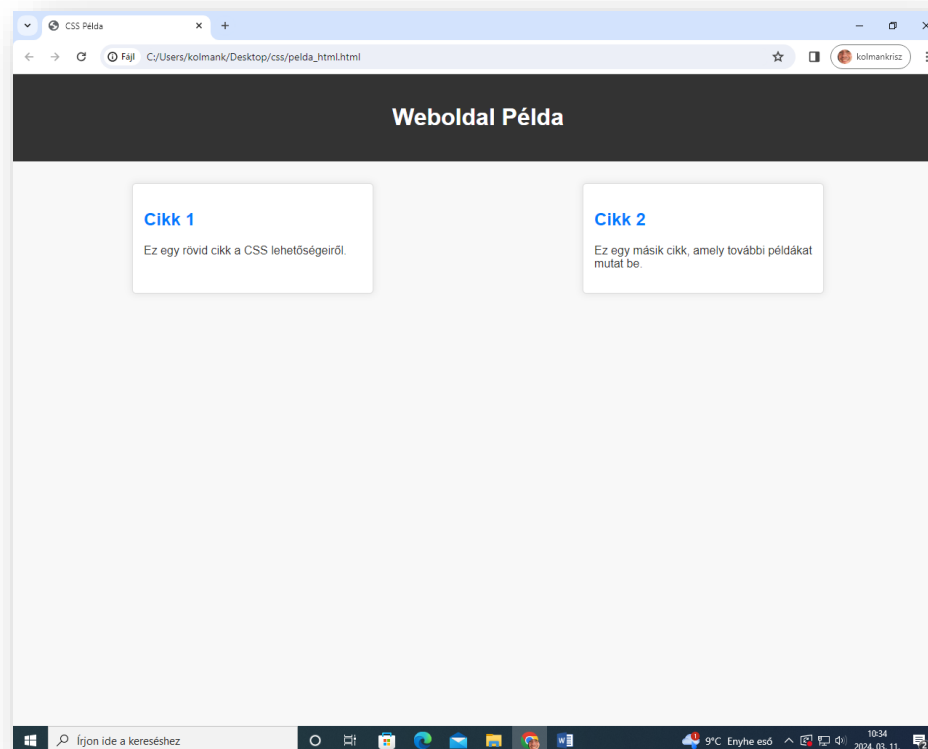
```
border-radius: 5px;
padding: 15px;
margin: 10px;
width: 300px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

/* Cikkfejlékek stílusa */
article h2 {
  color: #007bff;
}

/* Animáció */
article:hover {
  transform: scale(1.05);
  transition: transform 0.3s ease-in-out;
}
```

Magyarázat:

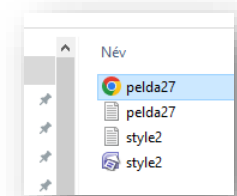
- Flexbox elrendezés: A .content osztályban a display: flex segítségével flexboxot alkalmaztunk a tartalom területre, és a justify-content: space-around elrendezéssel egyenletesen osztja el az elemeket.
- Dobozelemek formázása: A border, border-radius, padding, margin, és box-shadow tulajdonságokkal formázzuk az article elemeket, így szép keretet és árnyékot kapnak.
- Cikkfejlékek stílusa: Az article h2 választóval a cikkfejlékek külön színt kapnak.
- Animáció: Az article:hover segítségével egy egyszerű animációt alkalmazunk, amikor a felhasználó a cikkekre mutat a kurzorral. A transform: scale(1.05) növeli a méretét, és a transition tulajdonsággal szép lassított effektet kapunk.



Feladat 27:

A következő feladat az előzőre hajazó, de mégis máshogy kinéző oldalt hoz létre! Tehát nézzük akkor a további példát CSS kódra, amely különböző lehetőségeket és tulajdonságokat mutat be egy egyszerű weboldalra. A kód tartalmaz elrendezési beállításokat, szövegformázást, árnyékolást, és egy kis animációt is.

A fájl neveket a képről olvasd le!



HTML kód:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Példa 2</title>
  <link rel="stylesheet" href="style2.css">
</head>
<body>
  <header>
    <h1>Üdvözöllek a Weboldalamon!</h1>
    <p>Egy másik példa CSS kóddal.</p>
  </header>
  <section class="content">
    <div class="box">
      <h2>Box 1</h2>
      <p>Ez egy szép doboz a CSS tulajdonságokkal.</p>
    </div>
    <div class="box">
      <h2>Box 2</h2>
      <p>Még egy doboz a CSS különböző lehetőségeivel.</p>
    </div>
  </section>
</body>
</html>
```

CSS kód (style2.css):

```
/* Alapértelmezett stílusok */
body {
  font-family: 'Helvetica Neue', Arial, sans-serif;
  background-color: #f5f5f5;
  color: #333;
  margin: 0;
  padding: 0;
}

header {
  background-color: #3498db;
  color: #fff;
  text-align: center;
  padding: 20px;
```

```
}

/* Tartalom terület formázása */
.content {
  display: flex;
  justify-content: space-around;
  align-items: center;
  margin: 20px;
}

.box {
  background-color: #fff;
  border: 2px solid #ddd;
  border-radius: 8px;
  padding: 20px;
  margin: 10px;
  width: 250px;
  text-align: center;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

/* Cikkfejlékek stílusa */
.box h2 {
  color: #e74c3c;
}

/* Hover animáció */
.box:hover {
  transform: translateY(-5px);
  transition: transform 0.3s ease-in-out;
}

/* Szövegformázás */
p {
  line-height: 1.6;
  font-size: 16px;
  color: #555;
}
```

Magyarázat:

- Box Modell és Árnyékolás: A .box osztály formázása példázta a box modellt, a keretet (border), a borítást (padding), a margót (margin), és az árnyékolást (box-shadow).
- Flexbox és Alakítás: A .content osztályban a display: flex és a justify-content: space-around beállítások segítségével flexboxot használunk a tartalom elrendezésére. Az align-items: center középre igazítja a tartalmat mindkét tengelyen.
- Szövegformázás: A <p> elemekre alkalmazott szövegformázás példázta a betűméret (font-size), sorok közötti távolság (line-height), és a szövegszín (color) beállításokat.
- Hover Animáció: Az egyszerű hover animáció a :hover pszeudo-osztállyal és a transform tulajdonsággal valósul meg, amely elmozdítja a dobozt felfelé.
- Ez a példa bemutatja, hogy hogyan alkalmazhatók különböző CSS tulajdonságok és lehetőségek egy valóságos weboldal esetén.

