

HTML



L2: HTML5 és JavaScript laborgyakorlatok

Első rész: HTML5 űrlapok és JavaScript

- A feladatok lényege, hogy megismerjünk néhányat a **HTML5** új elemei és beviteli mezői közül, valamint gyakoroljuk a kliensoldali **JavaScript** használatát.
- **Fontos!:** mielőtt nekifognánk a feladatok megoldásának, olvassuk el a minőségi követelményeket tartalmazó megjegyzéseket

1. gyakorlat (közös)

- Kérjük be egy űrlap (**form**) segítségével az alábbi adatokat:
 - a felhasználó *neve* (család és keresztnév)
 - *születési dátuma* (*date picker*)
 - *e-mail címe*
 - kedvenc weboldalának a *címe*
- A születési dátum, e-mail cím, weboldal helyes megadásának érdekében **használjuk a dedikált input típusokat! Lásd a példát**
- Ellenőrizzük (a HTML **pattern** attribútumát használva), hogy a család- és keresztnév nagybetűvel kezdődjön, és csak betűket tartalmazzon. A reguláris kifejezéseke írjuk saját kezűleg, ne Internetről másoljuk (jó eszköz: <https://regex101.com/>)
- A dátum helyes formátumának ellenőrzését bizzuk a böngészőre.
- Ellenőrizzük JavaScript kód segítségével (**onblur** eseményre), hogy:
 - az e-mail cím érvényes gmail-es vagy yahoo-s e-mailcím
 - a weboldal címében pedig szerepel legalább egy doménium illetve egy aldoménium, csak kisbetűt, nagybetűt, számjegyet, alulvonást vagy kötőjelet tartalmaz, stb.
- Vonjuk össze a saját validátoraink a böngésző által generált automatikus validálás eredményével (tanulmányozzuk a `document.forms.formId.elementId.validity.valid` beállítást).

1. gyakorlat (közös) – folytatás

- Hiba esetén jelenítsünk meg egy megfelelő hibaüzenetet.
- A form leadását jelző `submit` gomb ne legyen elérhető, amíg hibák vannak a formban. Ez megoldható CSS validitás szerinti szelektorral, vagy JavaScripttel.
- JavaScriptet használva szűrjük be a HTML oldal aljára (a *footer*-be), középre igazítva az utolsó módosítás dátumát (lásd: `document.lastModified`).

Második rész: Interaktív oldalak

2.1. Feladat: Jelszóellenőrzés és szöveggörgetés

(a) Jelszó formátumának ellenőrzése:

Egészítsük ki az első részben létrehozott űrlapot (`form`) egy jelszó bekérésére alkalmas mezővel. Ellenőrizzük, hogy a begépelte jelszó teljesíti-e az alábbi feltételeket, és ha nem, írjuk ki, hogy mi az, ami nem felel meg:

- legalább egy, legtöbb két speciális karakter (ami nem betű és nem szám vagy alulvonás) kell szerepeljen benne, vagy legyen benne nagy- és kisbetű, illetve számjegy is
- legyen legalább 5 karakter hosszúságú
- 12 karakternél hosszabb nem lehet

Az ellenőrizhetőség kedvéért írjuk ki apró betűkkel a beírt jelszót.

(b) Szöveg görgetés:

- Helyezzünk el egy űrlapon egy szövegdobozt (`textarea`), ahova beírunk egy szöveget
- Ezt a szöveget fogjuk a szövegdoboz alatt elhelyezett `div` elem által kijelölt területen belül görgetni
- Helyezzünk továbbá egy legördülő listát (`select`), amelyből kiválaszthatjuk a görgetés irányát (balra vagy jobbra).
- Miközben beírjuk a szöveget, az elkezd a megadott irányba gördülni a következő szabályok alapján:
 - ha rövidebb a szöveg, mint a görgetés számára kijelölt `div` szélessége, akkor többször egymás után jelenik meg.
 - egy gomb megnyomásával újrakezdhető a görgetés.
 - a szöveg mindig a görgetésre szánt területnek a megadott iránnyal ellentétes oldalán indul és legyen olvasható.
 - úgy kell működjön, mint a (*nem standard!*) `marquee` tag, csak ebben az esetben egy `div`-ben kell megjelenjen a görgetett szöveg.

2.2. Feladat: Itt a piros, hol a piros

Egészítsük ki az első részben elkészített űrlapot (`form`) az alábbi adatok bekérésére alkalmas elemekkel:

- mennyi pénze van a felhasználónak – pénznem kiválasztása egy legördülő listából

Ellenőrizzük, hogy minden adat helyesen van megadva, illetve a játékot csak 18 éven felüli, egy bizonyos minimális pénzüsszeggel rendelkező egyénnek engedjük meg. Ha megkezdődött a játék, jelenjen meg a játékos neve az oldal egy jól látható helyén. A játék szabályai a következők:

- A játékos 3 kép közül választva tippelhet arra, hogy „hol a piros” (a képet, amire tippelt „fordítsuk fel”, hogy látható legyen az eredmény).
- Ha eltalálta a helyes (véletlenszerűen generált) pozíciót, (a pénznemtől függő) \times összeget nyer, különben kétszer annyit veszít.
- A játék addig ismétlődik, míg a játékos meg nem unja („meguntam” gombra kattintva) vagy a pénze a minimális alá csökken.
- Mindig legyen kiírva, hogy a játékosnak mennyi pénze maradt, illetve a nyerés vagy vesztes függvényében más-más kép jelenjen meg.

Tipp: véletlenszám generálása: `Math.random` segítségével.

2.3. Feladat: Kérdés-válasz párosítás

Egészítsük ki az első részben elkészített űrlapot (`form`) az alábbi adatok bekérésére alkalmas elemekkel: - alpművelet (összeadás/kivonás/szorzás/osztás) - 5 és 10 közötti érték (ennyi kérdés és válasz lesz generálva)

Generáljunk a kiválasztott művelet segítségével, megadott számú konkrét feladatot (az operandusok, és lehetőleg az eredmény is a 0-100 számkörben mozogjanak), pl. összeadás esetén (27+38, 44+21, stb.) Egy `canvas` elemen jelenítsük meg egy baloldali oszlopban a generált műveleteket úgy, hogy mindegyik egy színes téglalapban jelenjen meg, egy jobboldali oszlopban pedig az eredmények jelenjenek meg, véletlenszerű sorrendben, szintén színes téglalapokban.

A felhasználó feladata a műveletek párosítása a megfelelő eredményekkel: előbb a baloldali oszlopból választ egy téglalapot, rákattintva (váljon is láthatóvá, hogy kiválasztottuk, azaz keretezzük be egy másik színnel), majd a jobboldali oszlopból válasszunk egy eredményt (ekkor a baloldali kiválasztott téglalap köré rajzolt keret tűnjön el, és a két téglalapot kösse össze egy színes vonal).

Ha végeztünk minden párosítással, jelenjen meg egy „kiértékelés” gomb, melyre kattintva váljon láthatóvá, hogy melyik párosítások helyesek illetve melyek helytelenek.

A kértnél ergonomikusabban is megoldható a párosítás. Ha az aritmetikai műveletek párosítása nem mutat elég kihívást, lehet más jellegű generálható kérdés-válaszok párosítását implementálni.

2.4. Feladat: Tic-tac-toe

Írjunk tic-tac-toe (avagy x-o) játékot, amit egy 3x3-as táblán lehet játszani. Két játékos játszik egymás ellen, és mindig legyen jól láthatóan feltüntetve, hogy éppen melyik a soros. A megjelenítés történhet bármilyen módon (pl. képekkel egy táblázatban, `canvason`). Ellenőrizzük, és írjuk ki, ha nyert valaki, vagy írjuk ki a végén, hogy döntetlen, amennyiben egyik játékos sem nyert. A játékot egy gomb megnyomásával lehessen újratekdeni.

2.5. Feladat: Memóriajáték

Írjunk egy memóriajátékot, amely a rövid távú memória tesztelését hivatott megvalósítani. A játék a következő szabályok alapján működjön:

- Egy `canvason` véletlen pozíciókon rövid ideig egy kép jelenik meg.
- A felhasználónak a megjelent (és a rövid idő után eltűnt) kép helyére kell kattintania.
- Minél közelebb eltalálja a felhasználó a kép pozícióját, annál több pontot kap.
- A játékos aktuális pontszáma mindig legyen látható a `canvas` jobb felső sarkában.

- A játéknak legyen több szintje, attók függően, hogy mennyi ideig jelenik meg a kép (pl. 0.05s, 0.1s, 0.3s, stb.).
- A játéknak egy megadott számú forduló után van vége, ekkor az összpontszám és a „Játék vége” üzenet íródjon ki az oldalra.

2.6. Feladat: Távirányítás kocsi

Egy `canvas` elemre rajzolt „pálya” közepén jelenítsünk meg egy (alaphelyzetben balról–jobbra irányba néző) **autót** (lehet más kép is; nem kötelező, hogy autó legyen). Az első részben elkészített űrlapot (`form`) egészítsük ki az alábbi választási lehetőségekkel:

- egy legördülő lista, melyből a parancsot választhatjuk ki (előre/hátra mozdul, jobbra/balra fordul)
- egy szövegdoboz, melyben a parancs paraméterét adjuk meg:
 - „előre” vagy „hátra” esetén a lépésszámot adjuk meg
 - fordulás esetén pedig az elfordulás szögét adhatjuk meg

Ha az autó kifut a pályáról, jelenjen meg annak ellentétes oldalán. A megadott parancsokkal tetszőlegesen mozgathatjuk az autót, illetve egy gombra kattintva legyen visszaállítható az alaphelyzet.

2.7. Feladat: Hatból négyet játék

A játék kezdetén jelenjenek meg a játékos adatai az oldal egy jól látható helyén. Szabályok:

- A gép kiválaszt (véletlenszerűen generál) 6 megadott színből 4-et (egyik szín sem ismétlődhet), ezek sorrendjét is meghatározva. Ezt kell a játékos kitalálja.
- A játékos tippel (pl. négy legördülő listából választja ki a színeket, de tetszés szerint másképp is megoldható), majd a gép kiértékeli a tippet:
 - megmondja, hogy hány **pontatlan találat** (a szín talál, de a pozíciója nem) van
 - megmondja, hogy hány **pontos találat** (a szín és a pozíció is talál) van.
- A játékban hagyományosan fekete négyzet jelöli a pontatlan válaszokat, illetve fehér négyzet jelöli a pontos találatokat. Mi is hasonlóan oldjuk meg.
- A négyzeteket egy, az oldalon előre elhelyezett `canvas` elemen jelenítsük meg.
- A játék végén megjelenik az eredmény: kitalálta-e a játékos, s ha igen, akkor hány lépésből. A játékos életkorától függően más-más szöveget írjunk ki.

Lényeges, hogy a gép válaszából csak a helyes vagy helytelen találatok **száma** derül ki, az, hogy konkrétan melyik színre vonatkozik, az nem. Ha a játékos még nem találta ki a színeket a helyes sorrendben, akkor ismét tippelhet. A tippek és válaszok egymás alatt jelenjenek meg. Max. 8-at tippelhet a játékos, ha nem sikerül kitalálnia a 4 színt a helyes sorrendben, akkor veszített, különben nyer.

Tipp: véletlenszám generálása: `Math.random` segítségével.

2.8. Feladat: Jelszó és szövegszerkesztő

(a) Jelszó formátumának ellenőrzése

Egészítsük ki az első részben létrehozott űrlapot (`form`) egy jelszó bekérésére alkalmas mezővel. Ellenőrizzük, hogy a begépet jelszó teljesíti-e az alábbi feltételeket, és ha nem, írjuk ki, hogy mi az, ami nem felel meg:

- legalább egy számjegy kell szerepeljen benne (de ne csak számjegyekből álljon)
- legyen legalább 6 karakter hosszúságú
- 10 karakternél hosszabb nem lehet

Az ellenőrizhetőség kedvéért írjuk ki apró betűkkel a beírt jelszót.

(b) Szövegszerkesztő

Legyen két szövegdoboz (`textarea`), az egyikbe szóközzel elválasztott szavakat írunk, ezek lesznek a figyelt szavak. A másikba szöveget írunk. A második szövegdoboz alatt megjelenik a beírt szöveg megfelelőképpen kiszínezve, a gépeléssel egyidőben, a következő szabály szerint:

- Ha a második szövegdobozba írt szövegben valamely szó (akár részsóként is) szerepel az első szövegdobozban, akkor azt *pirossal* jelenítjük meg, a többi karakter színe *fekete*.
- Ha pl. az első szövegdobozban az **aaa** és **aab** szavak szerepelnek, akkor az **abaa** szó nincs kiszínezve, míg az az **aaab** teljesen piros.
- Az a) illetve b) pont egymástól független, de legyenek ugyanazon a oldalon (*vízszintes vonallal elválasztott külön részben*)

2.9. Feladat: Kő, papír, olló játék

Egészítsük ki az első részben elkészített űrlapot (`form`) az alábbi adatok bekérésére alkalmas elemekkel:

- játékos beceneve
- mennyi pénze van

Ha megkezdődött a játék, a játékos neve, beceneve, illetve pénze legyen feltüntetve jól látható módon. A játékos három megfelelő kép (*kő, papír vagy olló*) közül választva adja meg a saját tippjét, a gép pedig generálja a sajátját. Miután valamelyik képre kattintottunk, csak a kiválasztott kép maradjon látható, illetve a gép tippje is váljon láthatóvá. A nyertes fizet x összeget (*a gép kezdeti pénzösszegét generáljuk véletlenszerűen*). Az alábbi szabály adja meg, hogy ki a nyertes:

- Ha a két fél tippje megegyezik, nem nyert senki
- A kő elcsorbítja az ollót (*tehát a kő nyer*)
- Az olló elvágja a papírt (*az olló a nyertes*)
- A papír becsomagolja a követ (*a papír nyer*)

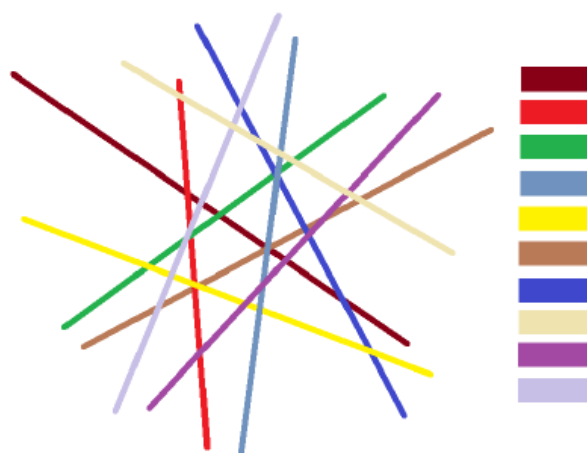
Tüntessük fel, hogy ki nyert, aktualizáljuk a játékos (és a gép) pénzösszegét, majd egy gombra kattintva folytathatjuk a játékot egy újabb fordulóval mindaddig, amíg óhajtjuk, illetve mindkét félnek van pénze.

2.10. Feladat: Pálcikák

Rajzoljunk 10 db. színes pálcikát egy `canvas`-ra (a szakaszok végpontjainak koordinátáit, illetve a tíz különböző színt megadhatjuk előre, a színek sorrendjét viszont generáljuk véletlenszerűen). A színeknek megfelelő téglalapokat helyezzünk el a kép jobb oldalán az ábrán szemléltetett módon.

A feladat az, hogy a pálcikákat a megfelelő sorrendben *távolítsuk el*, a színekre kattintva: mindig a legelső pálcikát kell elvenni. Ha a felhasználó a helyes színről kattintott, akkor a megfelelő pálcika tűnjön el a rajzról, különben a felhasználónak eggyel kevesebb „élete” marad (a kezdeti háromból, amit szintén jelenítsünk meg, tetszés szerint numerikus vagy grafikus formában). Egy „ismétlés” gombra kattintva ismétlődjön a játék.

Tipp: véletlenszám generálása: `Math.random` segítségével.



1. ábra. pálcikák

Megjegyzések

- A példaprogramokkal és a leadással kapcsolatos információk az [ubb-web-lab0-setup.pdf](#) állományban találhatóak.
- A létrehozott HTML oldalak teljes mértékben helyesek kell legyenek úgy szintaktikailag, mint szemantikailag. Ennek ellenőrzése céljából alábbi három eszközt használjuk:
 - A szintaktikai ellenőrzésre használjuk a W3C HTML validátorát: <http://validator.w3.org>
 - A szemantikai helyesség ellenőrzésére használhatjuk a következő eszközt: <https://gsnedders.html5.org/outliner/>
 - CSS állományok szintaktikai helyességének ellenőrzésére használtjuk a következő eszközt: <https://jigsaw.w3.org/css-validator/>
- További kódolási konvenciókat találunk az [ubb-web-javascript-konvencioik.pdf](#) állományban.
- HTML szerkesztők használatát *mellőzzük!*
- Próbáljunk meg az adott követelmények betartása mellett (esetleg annak ellenére) az oldal esztétikájára is figyelmet fordítani.

Segédanyagok és hasznos linkek

- A példaprogramok ezen témához tartozó része: [2-javascript/](#)
- *Beginning with DOM scripting and AJAX*: elektronikus könyv, mely elérhető a tárgy Canvas oldalán a [Files/Könyvészet/Beginning_JavaScript.pdf](#) útvonalon
- *Basic JavaScript and jQuery*: **Gitbook** elektronikus könyv, elérhető itt: <https://josefmonje.gitbooks.io/basic-JavaScript-and-jquery/content/>
- Form elemek helyességének ellenőrzése a developer.mozilla.org oldalon